**Методические материалы**

**Бакалавриат 01.03.02 Прикладная математика и информатика**

**ОПОП Системное программирование и компьютерные науки**

**Алгебра и геометрия**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

1. Может ли определитель матрицы

-го порядка () быть равен 69 и, если да, то при каком значении ?

1. Исследовать и найти общее решение системы

в зависимости от значения .

1. Найти первый столбец матрицы, обратной к матрице

-го порядка.

1. Известно, что векторы линейного пространства линейно независимы. Выяснить, при каких значениях линейно независимы векторы .
2. Пусть — квадратные матрицы одинакового порядка и . Доказать, что присоединённые матрицы удовлетворяют соотношению .
3. Доказать, что если ранг квадратной матрицы равен единице, то существует число такое что .

**Контрольная работа № 2**

1. Известно, что объём параллелепипеда, построенного на векторах равен 2. Найдите объём параллелепипеда, построенного на векторах и .
2. Найти все векторы , удовлетворяющие равенству , где .
3. В треугольнике известны его вершина и уравнения двух высот и . Составить уравнение стороны .
4. Составить уравнение биссекторной плоскости двугранного угла между плоскостями , в котором лежит точка .
5. Составить уравнение общего перпендикуляра к прямым
6. Центр окружности, описанной около правильного треугольника , расположен в точке . Найти координаты вершин и , если известно, что .
7. Плоский выпуклый четырёхугольник задан своими вершинами в пространстве: . Найти необходимые и достаточные условия того, что заданная точка является его внутренней точкой.

**Контрольная работа № 3**

1. Решить уравнение .
2. Найти геометрическое место точек, изображающих на комплексной плоскости числа , удовлетворяющие условию

.

1. Пользуясь методом Лагранжа, определить вид линии второго порядка .
2. Составить уравнения касательных к эллипсу , перпендикулярных прямой .
3. Найти смежные классы
   1. мультипликативной группы комплексных чисел, отличных от нуля, по подгруппе положительных действительных чисел;
   2. мультипликативной группы комплексных чисел, отличных от нуля, по подгруппе чисел, равных по модулю единице.

**Контрольная работа № 4**

1. Найти базисы суммы и пересечения подпространств и , где , , а .
2. Доказать, что множество образует линейное подпространство пространства . Найти два различных дополнительных подпространства к .
3. Построить какой-либо ортонормированный базис линейной оболочки матриц , , .
4. Найти ортогональную проекцию вектора на подпространство

.

1. Определить расстояние от многочлена до многообразия .
2. Доказать, что если две гиперплоскости не пересекаются, то они параллельны.

**Контрольная работа № 5**

1. Оператор действует в пространстве по правилу . Построить матрицу этого оператора в базисе и указать какие-либо базисы его ядра и образа .
2. Найти все собственные значения и собственные векторы матрицы

.

1. Показать, что матрица диагонализуема, и привести её к диагональной подходящим преобразованием подобия.
2. Найти жорданову форму следующей матрицы и построить соответствующий канонический базис:

.

1. Оператор задан матрицей в базисе пространства со стандартным скалярным произведением. Найти матрицу сопряжённого оператора в этом же базисе .
2. Найти квадратный корень из матрицы .
3. Известно, что операторы удовлетворяют условию: произведение является тождественным оператором в пространстве . Доказать, что если пространства и имеют разную размерность, то произведение не может быть тождественным оператором в пространстве .

**Контрольная работа № 6**

1. Линейный оператор задан в некотором ортонормированном базисе матрицей

.

Построить ортонормированный базис из собственных векторов этого оператора и найти его матрицу в построенном базисе.

1. Найти канонический вид матрицы

и указать ортогональную матрицу , такую что .

1. Пусть – положительно определённый линейный оператор, действующий в евклидовом пространстве . Доказать, что существует положительное число такое, что для любого вектора справедливо неравенство .
2. Найти ортогональное преобразование, приводящее квадратичную форму

к каноническому виду и написать этот канонический вид.

1. Найти нормальное псевдорешение системы линейных алгебраических уравнений

**Вопросы к коллоквиуму (первый семестр)**

Коллоквиум проводится в форме устного собеседования. Для не сдавших коллоквиум зачёт (и зачётная комиссия) начинается с вопросов по теоретическому материалу коллоквиума. Билет коллоквиума содержит один вопрос из следующего списка:

1. Перестановки.
2. Определитель, свойства определителя.
3. Миноры и их алгебраические дополнения. Теорема Лапласа.
4. Разложение определителя по строке (столбцу). Определитель произведения матриц.
5. Обратная матрица. Критерий обратимости.
6. Ранг матрицы. Теорема о базисном миноре.
7. Ранг матрицы и линейная зависимость её строк (и столбцов).
8. Ранг произведения матриц. Инвариантность ранга относительно элементарных преобразований.
9. Системы линейных алгебраических уравнений с квадратной невырожденной матрицей. Правило Крамера.
10. Критерий совместности и определённости системы линейных алгебраических уравнений.
11. Исследование и решение системы линейных алгебраических уравнений общего вида. Общее решение.
12. Эквивалентность систем линейных алгебраических уравнений. Элементарные преобразования систем.
13. Метод Гаусса исследования и решения систем линейных алгебраических уравнений.
14. Линейное пространство. Арифметическое пространство.
15. Линейная зависимость в линейном пространстве.
16. Базис и размерность линейного пространства.
17. Линейное подпространство и линейное аффинное многообразие в линейном пространстве. Определение и простейшие свойства.
18. Геометрические свойства решений однородной системы линейных алгебраических уравнений. Фундаментальная система решений.
19. Геометрические свойства решений неоднородной системы линейных алгебраических уравнений. Общее решение.

**Вопросы к коллоквиуму (второй семестр)**

Коллоквиум проводится в форме устного собеседования. Для не сдавших коллоквиум зачёт (и зачётная комиссия) начинается с вопросов по теоретическому материалу коллоквиума. Билет коллоквиума содержит один вопрос из следующего списка:

1. Линейные операторы. Определение, основные свойства, примеры. Теорема о существовании и единственности оператора по заданным образам базисных векторов.
2. Матрицы линейных операторов. Взаимно-однозначное соответствие между линейными операторами и матрицами.
3. Линейное пространство линейных операторов и его связь с пространством матриц.
4. Матрица линейного оператора. Связь между координатами вектора и его образа.
5. Матрицы линейного оператора в различных базисах.
6. Эквивалентные матрицы. Критерий эквивалентности.
7. Образ и ядро линейного оператора.
8. Произведение линейных операторов. Матрица произведения.
9. Обратный оператор. Критерий обратимости.
10. Инвариантные подпространства. Индуцированный оператор.
11. Инвариантные подпространства минимальной размерности (в комплексном и вещественном случаях).
12. Собственные значения и собственные векторы линейного оператора. Определение и простейшие свойства. Примеры.
13. Характеристический многочлен линейного оператора. Определение и простейшие свойства.
14. Условие существования собственных векторов линейного оператора. Собственные векторы линейного оператора в комплексном пространстве.
15. Собственное подпространство. Алгебраическая и геометрическая кратности собственного значения.
16. Операторы простой структуры. Критерий простой структуры.
17. Треугольная форма матрицы линейного оператора в комплексном пространстве.
18. Нильпотентный оператор. Определение, простейшие свойства, примеры.
19. Расщепление линейного оператора.
20. Корневые векторы. Канонический базис корневого подпространства.
21. Жорданова нормальная форма матрицы линейного оператора. Канонический базис.
22. Теорема Гамильтона-Кэли.
23. Подобные матрицы. Критерий подобия.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Зачетная работа первого семестра**

**Вариант 1 (для проведения в группах)**

1. Вычислить определитель n-го порядка

,

где .

1. Исследовать систему и найти общее решение в зависимости от значений
2. В аффинной системе координат написать уравнение прямой, проходящей через точку и равноудалённой от точек и .
3. Составить параметрическое уравнение прямой, параллельной прямой

и пересекающей прямые и .

1. Построить однородную систему уравнений по заданной фундаментальной системе решений: .
2. Вычислить объём параллелепипеда , зная его вершину и координаты концов выходящий из неё рёбер: .
3. На плоскости заданы две системы координат: и . Вторая система получена из первой поворотом вокруг точки на угол в направлении кратчайшего поворота от к . Найти координаты точки в первой системе координат, если известны её координаты во второй системе координат.
4. Составить уравнение биссектрисы острого угла между прямыми и .
5. Написать уравнение плоскости, проходящей через точку и удаленной от точки на расстояние 1, а от точки на расстояние 3.
6. Решить уравнение в комплексных числах: .
7. Найти все образующие элементы циклической группы 11-го порядка.
8. Определить тип кривой, заданной уравнением

и найти уравнения осей ее канонической системы координат.

**Вариант 2 (для проведения зачетной комиссии)**

1. Вычислить определитель n-го порядка

где .

1. Исследовать систему и найти общее решение в зависимости от значений
2. В аффинной системе координат написать уравнение прямой, проходящей через точку и равноудалённой от точек и .
3. Вычислить ранг матрицы

.

1. Составить параметрическое уравнение прямой, параллельной прямой

и пересекающей прямые и

1. В треугольнике заданы уравнение стороны и медиан , . Составить уравнение высоты треугольника, проведенной из вершины .
2. Написать уравнение плоскости , проходящей через начало координат перпендикулярно прямой

,

найти расстояние от точки до этой плоскости и координаты проекции этой точки на плоскость .

1. Определить тип поверхности, заданной уравнением .

Найти геометрическое место точек комплексной плоскости, удовлетворяющих условию .

**Зачетная работа второго семестра**

**Вариант 1 (для проведения в группах)**

1. Найти базисы суммы и пересечения линейных подпространств и где , , ; , , .
2. Применяя процесс ортогонализации, построить ортогональный базис линейной оболочки векторов , , , .
3. Найти угол между вектором и линейной оболочкой векторов , , .
4. Найти канонический базис и жорданову форму матрицы

.

1. Доказать, что неоднородная система линейных уравнений совместна тогда и только тогда, когда вектор-столбец ортогонален всем решениям сопряженной однородной системы .
2. В пространстве многочленов со стандартным скалярным произведением задан ортогональный оператор с определителем, равным , который переводит многочлен в , а многочлен в . Найти матрицу оператора в базисе .
3. Найти нормальный вид квадратичной формы

и приводящее к нему треугольное преобразование координат.

1. Найти нормальное псевдорешение системы уравнений
2. В пространстве введено скалярное произведение

Найти матрицу оператора, сопряженного к оператору дифференцирования, в базисе .

1. Доказать, что пространство является прямой суммой подпространств и , и найти проекцию многочлена на параллельно , если , .

**Вариант 2 (для проведения зачетной комиссии)**

1. Найти базисы , если задано однородной системой

а является ортогональным дополнением к множеству решений системы

1. Найти базисы образа и ядра линейного оператора, отображающего матрицы , , , соответственно в матрицы , , , .
2. Построить жорданову форму и канонический базис для матрицы

.

1. Найти расстояние от точки, заданной вектором до линейного аффинного многообразия , заданного системой уравнений
2. Выписать канонический вид и приводящее к этому виду ортогональное преобразование координат для квадратичной формы

.

1. Найти двумерное инвариантное подпространство для линейного оператора, действующего в пространстве и заданного в некотором его базисе матрицей

.

**Вопросы к экзамену**

Экзамен сдается в устной форме. В экзаменационном билете – два вопроса из приведенных ниже списков по семестрам.

**Первый семестр**

**Линейная алгебра**

1. Операции над матрицами и их свойства.
2. Приведение матрицы к ступенчатому виду. Приведение к диагональному виду.
3. Перестановки, транспозиции, чётность.
4. Определитель и его свойства как функции столбцов (строк).
5. Определитель транспонированной матрицы.
6. Определитель произведения матриц.
7. Миноры и их алгебраические дополнения. Теорема Лапласа.
8. Невырожденные матрицы. Обратные матрицы. Критерий обратимости матрицы.
9. Линейное пространство. Определение и примеры. Арифметическое пространство.
10. Линейная зависимость в линейном пространстве.
11. Базис и размерность линейного пространства.
12. Переход к другому базису, матрица перехода.
13. Ранг матрицы. Теорема о базисном миноре.
14. Ранг матрицы и линейная зависимость строк и столбцов.
15. Ранг произведения матриц. Ранг матрицы и элементарные преобразования.
16. Эквивалентные матрицы. Критерий эквивалентности.
17. Системы линейных алгебраических уравнений. Эквивалентность систем. Элементарные преобразования систем.
18. Системы с невырожденной матрицей. Правило Крамера.
19. Критерий совместности системы линейных алгебраических уравнений. Критерий единственности решения.
20. Исследование системы линейных алгебраических уравнений общего вида. Главные и свободные неизвестные. Общее решение системы.
21. Метод Гаусса исследования и решения систем линейных алгебраических уравнений. Число арифметических операций в методе Гаусса.
22. Линейное подпространство. Геометрические свойства множества решений однородной системы линейных алгебраических уравнений. Фундаментальная система решений. Общее решение.
23. Линейное многообразие. Геометрические свойства множества решений неоднородной системы линейных алгебраических уравнений. Общее решение.

**Аналитическая геометрия**

1. Направленные отрезки. Свободный вектор.
2. Линейные операции над векторами. Координаты вектора.
3. Проекции вектора. Свойства линейности проекций.
4. Линейная зависимость векторов. Коллинеарные и компланарные векторы.
5. Аффинная система координат. Преобразование координат.
6. Преобразования прямоугольных декартовых координат. Ортогональные матрицы.
7. Скалярное произведение геометрических векторов. Скалярное произведение в прямоугольных декартовых координатах.
8. Векторное произведение векторов.
9. Смешанное произведение векторов.
10. Векторное и смешанное произведения в прямоугольных декартовых координатах.
11. Алгебраические линии и поверхности. Инвариантность порядка линии (поверхности).
12. Параметрические уравнения прямой на плоскости и плоскости в пространстве.
13. Общее уравнение прямой на плоскости в аффинной системе координат. Критерий параллельности вектора прямой.
14. Общее уравнение плоскости в пространстве в аффинной системе координат. Критерий параллельности вектора плоскости.
15. Взаимное расположение двух прямых на плоскости и плоскостей в пространстве.
16. Пучок прямых на плоскости и плоскостей в пространстве.
17. Полуплоскости и полупространства.
18. Уравнения прямой в пространстве.
19. Взаимное расположение прямых в пространстве.
20. Метрические задачи на прямую и плоскость в прямоугольных координатах.
21. Общее уравнение линии второго порядка на плоскости. Матричная запись общего уравнения и его квадратичной части.
22. Приведённые уравнения линии второго порядка на плоскости. Метод вращений.
23. Классификация линий второго порядка на плоскости.
24. Эллипс. Фокусы и директрисы.
25. Гипербола. Фокусы и директрисы.
26. Парабола. Фокус и директриса.
27. Общее уравнение поверхности второго порядка в пространстве. Матричная запись общего уравнения и его квадратичной части.
28. Приведённые уравнения поверхности второго порядка. Метод вращений.
29. Классификация поверхностей второго порядка. Эллипсоиды, гиперболоиды, параболоиды, конусы и цилиндрические поверхности.
30. Прямолинейные образующие алгебраических поверхностей второго порядка.

**Общая алгебра**

1. Декартово произведение множеств и бинарное отношение. Отношение эквивалентности. Фактор-множество.
2. Отображения. Обратное отображение.
3. Алгебраические операции. Обобщённый закон ассоциативности.
4. Группы. Основные свойства.
5. Подгруппы. Симметрическая и знакопеременная группы.
6. Группа невырожденных матриц. Группа невырожденных треугольных матриц. Группа ортогональных матриц.
7. Конечные группы. Теорема Лагранжа.
8. Степени элемента. Циклические группы. Подгруппы циклической группы.
9. Подгруппы, смежные классы, нормальные делители.
10. Изоморфизм групп.
11. Гомоморфизм групп.
12. Кольцо.
13. Поле. Характеристика поля. Алгебраическое расширение поля.
14. Кольцо вычетов. Поле вычетов по простому модулю.
15. Линейное пространство над полем. Число элементов в конечном поле.
16. Поле комплексных чисел. Комплексная плоскость.
17. Тригонометрическая форма комплексного числа. Модуль и аргумент произведения комплексных чисел.
18. Возведение в степень комплексного числа. Формула Муавра.
19. Извлечение корня из комплексного числа.
20. Группа корней из единицы. Первообразные корни.
21. Кольцо многочленов. Деление с остатком.
22. Наибольший общий делитель, его свойства. Алгоритм Евклида.
23. Значения многочлена и корни. Теорема Безу.
24. Многочлены как формальные выражения и как функции. Эквивалентность двух определений равенства многочленов.
25. Основная теорема алгебры. Разложение многочлена на линейные множители.
26. Каноническое разложение многочлена над полем комплексных чисел. Кратность корня.
27. Каноническое разложение многочленов над полем вещественных чисел.
28. Формулы Виета. Симметрические многочлены.

**Второй семестр**

1. Линейное пространство над произвольным полем. Ранг и база системы векторов.
2. Изоморфизм линейных пространств.
3. Сумма и пересечение линейных пространств.
4. Прямая сумма линейных пространств.
5. Евклидово и унитарное пространство. Неравенство Коши-Буняковского-Шварца.
6. Скалярное произведение в ортонормированном базисе. Существование ортонормированного базиса.
7. Изометрия.
8. Матрица Грама. Критерий линейной зависимости.
9. Ортогональное дополнение. Ортогональная сумма подпространств. Расстояние от вектора до подпространства.
10. Ортонормированный базис и унитарные (ортогональные) матрицы.
11. Процесс ортогонализации Грама-Шмидта. QR-разложение матрицы.
12. Линейное аффинное многообразие в линейном пространстве. Гиперплоскость в евклидовом и унитарном пространстве.
13. Линейные операторы. Матрица линейного оператора.
14. Матрица линейного оператора при переходе к другому базису. Эквивалентность и подобие матриц.
15. Линейное пространство линейных операторов и матриц.
16. Произведение линейных операторов и его матрица.
17. Ядро и образ линейного оператора. Каноническая пара базисов.
18. Линейные функционалы. Сопряжённое пространство. Линейные функционалы и гиперплоскости.
19. Обратный оператор. Критерии обратимости.
20. Собственные значения и собственные векторы. Операторы простой структуры и диагонализуемые матрицы.
21. Характеристический многочлен линейного оператора. Условие существования собственных значений.
22. Собственное подпространство. Геометрическая и алгебраическая кратности собственных значений.
23. Инвариантные подпространства. Сужение оператора.
24. Треугольная форма матрицы линейного оператора. Теорема Шура.
25. Сдвиг оператора, нильпотентность и обратимость его сужений.
26. Корневые подпространства. Расщепление линейного пространства в прямую сумму корневых подпространств.
27. Жорданов базис и жорданова матрица линейного оператора в комплексном пространстве.
28. Критерий подобия матриц.
29. Теорема Гамильтона-Кэли. Минимальный многочлен.
30. Инвариантные подпространства минимальной размерности.
31. Вещественный аналог жордановой формы.
32. Сопряжённый оператор. Существование и единственность. Матрица сопряжённого оператора.
33. Нормальный оператор и нормальная матрица.
34. Блочно-диагональная форма вещественной нормальной матрицы.
35. Эрмитовы операторы и эрмитовы матрицы. Эрмитово разложение линейного оператора.
36. Симметрические операторы и симметрические матрицы.
37. Унитарные операторы и унитарные матрицы.
38. Блочно-диагональная форма ортогональной матрицы.
39. Знакоопределённые операторы и матрицы. Квадратный корень из оператора.
40. Сингулярные числа и сингулярные векторы. Полярное разложение оператора (матрицы).
41. Ортогональные дополнения ядра и образа линейного оператора. Теорема и альтернатива Фредгольма.
42. Билинейные и квадратичные формы. Приведение к каноническому виду. Конгруэнтность и эрмитова конгруэнтность.
43. Закон инерции квадратичных форм.
44. Приведение квадратичной формы к главным осям.
45. Одновременное приведение к каноническому виду пары квадратичных форм.
46. Положительно определённые квадратичные формы. Критерий Сильвестра.
47. Общий вид скалярного произведения в конечномерном евклидовом и унитарном пространствах.
48. Гиперповерхность второго порядка в евклидовом пространстве. Приведённые уравнения.
49. Нормированное пространство. Нормы Гёльдера.
50. Длина вектора. Тождество параллелограмма и критерий евклидовости нормы.
51. Эквивалентность норм в конечномерном пространстве.
52. Задача о наилучшем приближении в конечномерном нормированном пространстве.
53. Линейный оператор в нормированных пространствах. Непрерывность и ограниченность. Норма линейного оператора.
54. Матричные нормы. Унитарно инвариантные нормы.
55. Сингулярное разложение матрицы и обобщённое решение линейных систем.
56. Вариационные (экстремальные) свойства собственных значений самосопряжённого оператора (матрицы).
57. Вариационные (экстремальные) свойства сингулярных чисел.
58. Соотношения разделения собственных значений и сингулярных чисел матриц и подматриц.

**Примеры экзаменационных билетов:**

1. Ранг матрицы. Теорема о базисном миноре.
2. Метрические задачи на прямую и плоскость в прямоугольных координатах.
3. Ортогональное дополнение. Ортогональная сумма подпространств. Расстояние от вектора до подпространства..
4. Треугольная форма матрицы линейного оператора. Теорема Шура.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Математический анализ I**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

1.Найти inf,  ;

2.Доказать, что ;

3.Пользуясь критерием Коши, исследовать на сходимость :

;

4.Доказать, что .

**Контрольная работа № 2**

1.Вычислить предел: ;

2. Выделить у данной функции  главный член вида: .

3.Определить характер точек разрыва следующей функции .

4. Исследовать на непрерывность следующую функцию: 

4. Исследовать на непрерывность следующую функцию: 

**Контрольная работа № 3**

1. Найти , если ;
2. Найти , если ;
3. Найти , если .
4. Разложить данную функцию  по формуле Тейлора в окрестности указанной точки  до членов III порядка включительно: .
5. Найти предел, пользуясь формулой Тейлора: .

6. Раскрыть неопределённость: .

**Контрольная работа № 4**

**Вариант №1**

Вычислить следующие интегралы:

1. ;
2. ;
3. ;
4. 

**Вариант №2**

Вычислить следующие интегралы:

1. ;
2. ;
3. ;

4. ****

**Самостоятельная работа № 5**

1) В задачах №1, 2, 3 выполнить полное исследование функции и построить её график:

1)  ; 2) ; 3) 

2) Найти прямоугольник наибольшей площади, вписанный в эллипс: .

**Вопросы к коллоквиуму**

Коллоквиум проводится в форме устного собеседования. Темы коллоквиума:

1. Вещественные числа, правило их сравнения. Теорема о существовании точной верхней (нижней) грани у ограниченного сверху(снизу) числового множества.
2. Арифметические операции над вещественными числами. Свойства вещественных чисел.
3. Понятие об эквивалентных и неэквивалентных (равномощных и неравномощных) множествах. Счётные множества и множества мощности континуум. Доказательство их неэквивалентности. Полнота множества вещественных чисел. Аксиоматический метод задания вещественных чисел.
4. Числовая последовательность. Предел числовой последовательности. Теорема о единственности предела. Теорема об ограниченности сходящейся последовательности.
5. Бесконечно малые и бесконечно большие последовательности. Их взаимосвязь и свойства. Примеры.
6. Арифметические операции над сходящимися последовательностями.
7. Предельный переход в неравенствах для последовательностей.
8. Расширенная числовая ось. Бесконечно удалённые точки. Понятие -окрестности конечных и бесконечных точек. Понятие предела последовательности в терминах окрестностей.
9. Теорема о пределе монотонной ограниченной последовательности. Число e.
10. Понятие предельной точки множества и предельной точки последовательности. Теорема о существовании верхнего и нижнего пределов у бесконечного ограниченного множества. Теорема Больцано-Вейерштрасса об ограниченной последовательности.
11. Фундаментальная последовательность и её свойства. Критерий Коши сходимости последовательности.
12. Два определения предела (предельного значения) функции: по Гейне и по Коши, их эквивалентность. Единственность предела функции в данной точке. Односторонние пределы. Бесконечные пределы и пределы на бесконечности.
13. Критерий Коши существования предела функции.
14. Бесконечно малые функции в окрестности данной точки, сравнение порядков их малости. Бесконечно большие функции в окрестности данной точки, сравнение порядков их роста. Символы о-малое, O-большое, O-большое со звёздочкой. Понятие об эквивалентных бесконечно малых (бесконечно больших) функциях. Примеры.
15. Арифметические операции над функциями, имеющими пределы.
16. Предельный переход в функциональных неравенствах.
17. Непрерывность функции в точке. Определения непрерывности по Гейне и по Коши. Непрерывность функции в точке слева или справа. Локальные свойства непрерывных функций: ограниченность, сохранение знака.
18. Арифметические операции над непрерывными функциями. Суперпозиция функций. Непрерывность сложной функции.
19. Точки разрыва функции. Их классификация. Примеры.
20. Непрерывность функции на множестве. Свойства функций, непрерывных на отрезке: теоремы о прохождении функции через нуль и через промежуточное значение.
21. Теоремы об ограниченности функции, непрерывной на отрезке (I теорема Вейерштрасса) и о достижении такой функцией точных верхней и нижней граней её значений (II теорема Вейерштрасса).
22. Равномерная непрерывность функции на множестве. Теорема Кантора о равномерной непрерывности функции на отрезке.

**Типовой билет коллоквиума**

1. Дать определение того, что число M является точной верхней гранью множества значений функции f(x) на отрезке [0;2].
2. Сформулировать первую теорему Вейерштрасса.
3. Дать определение по Коши того факта, что соотношение  неверно.
4. Дать определение точки разрыва II рода для функции одной переменной.
5. Дать определения, формулировки всех утверждений и привести их доказательства по следующей теме:

Предельные точки множества и последовательности. Теорема о существовании верхнего и нижнего пределов у ограниченной последовательности.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Зачетная работа**

1. Исследовать на сходимость следующую числовую последовательность:

.

1. Вычислить предел функции:.
2. Исследовать функцию на непрерывность и дифференцируемость: .
3. Найти , где , а функция  задана так: .
4. Найти , если , и  заданы следующим образом:

.

1. Вычислить следующие неопределённые интегралы:

а);

б) ;

в).

1. Вычислить главный член функции  вида:  при .
2. Вычислить главный член функции  вида:  при .
3. Исследовать функцию на равномерную непрерывность: , .
4. Доказать функциональное неравенство: .
5. Разложить по формуле Тейлора в окрестности указанной точки до членов III порядка следующую функцию: .
6. Пользуясь формулой Тейлора, найти предел: .

**Вопросы к экзамену**

1. Вещественные числа и правила их сравнения. Теорема о существовании точной верхней (нижней) грани у ограниченного сверху (снизу) множества вещественных чисел.
2. Приближение вещественного числа рациональным. Арифметические операции над вещественными числами. Свойства вещественных чисел.
3. Счетные множества и множества мощности континуум. Неэквивалентность множества мощности континуум счетному множеству.
4. Ограниченные и неограниченные последовательности. Бесконечно большие и бесконечно малые последовательности. Их основные свойства.
5. Понятие сходящейся последовательности. Основные теоремы о сходящихся последовательностях (единственность предела, ограниченность сходящейся последовательности, арифметические операции над сходящимися последовательностями).
6. Предельный переход в неравенствах. Теорема о пределе монотонной ограниченной последовательности. Число е.
7. Понятие предельной точки последовательности. Теорема о существовании верхнего и нижнего пределов у ограниченной последовательности. Теорема Больцано-Вейерштрасса.
8. Необходимое и достаточное условие сходимости последовательности (критерий Коши).
9. Два определения предельного значения функции (по Гейне и по Коши) и доказательство их эквивалентности. Критерий Коши существования предельного значения функции.
10. Арифметические операции над функциями, имеющими предельное значение. Предельный переход в неравенствах. Бесконечно малые и бесконечно большие (в данной точке) функции и принципы их сравнения. Предел сложной функции.
11. Понятие непрерывности функции в точке и на множестве. Арифметические операции над непрерывными функциями. Классификация точек разрыва.
12. Локальные свойства непрерывных функций. Непрерывность сложной функции.
13. Обратная функция. Условия непрерывности монотонных функций и обратных функций.
14. Простейшие элементарные функции и их основные свойства.
15. Замечательные пределы.
16. Прохождение непрерывной функции через любое промежуточное значение.
17. Ограниченность функции, непрерывной на сегменте (первая теорема Вейерштрасса).
18. О достижении функцией, непрерывной на сегменте, своих точной верхней и нижней граней (вторая теорема Вейерштрасса).
19. Понятие равномерной непрерывности. Теорема Кантора.
20. Понятие производной и дифференцируемости функции в точке.
21. Правила дифференцирования суммы, произведения и частного двух функций, сложной функции и обратной функции. Формулы дифференцирования простейших элементарных функций.
22. Первый дифференциал функции. Инвариантность его формы. Использование дифференциала для приближенного вычисления приращения функции.
23. Производные и дифференциалы высших порядков, формула Лейбница. Дифференцирование функции, заданной параметрически.
24. Понятие возрастания (убывания) в точке и локального экстремума функции. Достаточное условие возрастания (убывания) и необходимое условие экстремума дифференцируемой в данной точке функции.
25. Теорема о нуле производной (теорема Ролля) и ее геометрический смысл.
26. Формула конечных приращений (формула Лагранжа). Следствия теоремы Лагранжа.
27. Обобщенная формула конечных приращений (формула Коши).
28. Раскрытие неопределенностей (правила Лопиталя).
29. Формула Тейлора с остаточным членом в общей форме (в форме Шлемильха-Роша).
30. Остаточный член в формуле Тейлора в форме Лагранжа, Коши и Пеано. Его оценка.
31. Разложение по формуле Тейлора-Маклорена элементарных функций. Примеры приложений формулы Тейлора для приближенных вычислений элементарных функций и вычисления пределов.
32. Понятие первообразной и неопределенного интеграла функции. Простейшие свойства неопределенного интеграла. Таблица неопределенных интегралов.
33. Простейшие методы интегрирования (замена переменной, интегрирование по частям).
34. Интегрируемость в элементарных функциях класса рациональных дробей (с вещественными коэффициентами).
35. Интегрируемость в элементарных функциях дробно-линейных иррациональностей и других классов функций.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Математический анализ II**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

1. Исследовать на сходимость интеграл: .
2. Исследовать на абсолютную и условную сходимость: .

Вариант №2.

1. Исследовать на сходимость интеграл: .

2. Исследовать на абсолютную и условную сходимость: .

**Контрольная работа № 2**

1. Исследовать на непрерывность по каждой переменной и по совокупности:



1. Исследовать на дифференцируемость:



3. Найти  функции  если 

**Самостоятельная работа № 3**

1. Найти , если , где



1. Разложить по формуле Маклорена до членов 4-го порядка малости функцию , если

.

1. Написать уравнения касательной прямой и нормальной плоскости в данной точке к следующей кривой:

.

**Контрольная работа № 4**

1. Произвести замену переменных в следующем дифференциальном выражении:

 при .

1. Найти условные экстремумы функции:  при условии:.
2. Найти наибольшее и наименьшее значения функции:  в области .

**Вопросы к коллоквиуму**

Коллоквиум проводится в форме устного собеседования. Темы коллоквиума:

*Графическое исследование функции*

1. Понятие локального экстремума функции. Необходимое условие локального экстремума дифференцируемой функции.

2. Понятие монотонности функции в точке и на множестве. Критерий монотонности дифференцируемой функции.

3. Первое достаточное условие локального экстремума.

4. Второе достаточное условие локального экстремума.

5. Направление выпуклости графика функции. Понятие о точках перегиба.

6. Достаточные условия локальной выпуклости графика и выпуклости его на интервале (a;b).

7. Необходимое условие перегиба графика в данной точке.

8. Первое достаточное условие перегиба в данной точке.

9. Второе достаточное условие перегиба в данной точке.

10. Отыскание асимптот к графику функции (вертикальных и наклонных).

11. Отыскание наибольшего и наименьшего значения функции на сегменте [a;b] (глобальный экстремум). Понятие о краевом экстремуме.

*Определенный интеграл.*

12. Понятие об определённом интеграле. Верхняя и нижняя интегральные суммы (суммы Дарбу), их свойства. Интегралы Дарбу.

13. Критерий интегрируемости функции.

14. Интегрируемость непрерывных, монотонных, кусочно-непрерывных функций.

15. Свойства определённого интеграла: аддитивность, линейность, интегрируемость произведения функций, сравнение интегралов от двух различных функций, интегрируемость модуля функции.

16. Свойства определённого интеграла: первая теорема о среднем, формулировка второй теоремы о среднем, интеграл с переменным верхним пределом, теорема о существовании первообразной у всякой непрерывной функции. Формула Ньютона-Лейбница – основная формула интегрального исчисления.

17. Замена переменной и интегрирование по частям в определённом интеграле.

*Приложения определенного интеграла.*

18. Квадрируемость и понятие площади плоской фигуры. Вычисление площади криволинейной трапеции и площади криволинейного сектора. Геометрический смысл определённого интеграла.

19. Кубируемость и понятие объёма тела в пространстве. Вычисление объёма тела, полученного вращением криволинейной трапеции вокруг оси OX. Формула (без вывода) для объёма тела , полученного вращением криволинейной трапеции вокруг оси OY,

20. Спрямляемость кривой и понятие длины кривой. Вычисление длины дуги кривых, заданных параметрически, а также в декартовых или в полярных координатах. Понятие о дифференциале длины дуги кривой.

21. Понятие о физических приложениях определённого интеграла.

22. Приближённые методы вычисления определённого интеграла. Метод прямоугольников. Его погрешность

23. Приближённые методы вычисления определённого интеграла. Метод трапеций. Его погрешность (без доказательства).

24. Приближённые методы вычисления определённого интеграла. Метод парабол (Симпсона). Его погрешность (без доказательства).

*Несобственные интегралы*

25. Несобственный интеграл первого рода, его сходимость. Критерий Коши сходимости несобственного интеграла. I-го рода. Вычисление с помощью формулы Ньютона-Лейбница.

26. Достаточные условия сходимости несобственного интеграла. I-го рода. Признаки сравнения: общие, специальные (с интегралом Дирихле), признаки сравнения в предельной формулировке.

27. Абсолютная и условная сходимость несобственного интеграла I рода. Признак Абеля-Дирихле.

28. Исследование на абсолютную и условную сходимость интеграла .

29. Замена переменных и интегрирование по частям в несобственном интеграле первого рода.

30. Несобственный интеграл второго рода. Понятие о его сходимости. Критерий Коши. Признаки сравнения для несобственного интеграла II рода: общие и специальные (с интегралом Дирихле II рода).

31. Понятие о главном значении по Коши несобственных интегралов I и II рода.

**Типовой билет коллоквиума**

**Дать определение или формулировку:**

1. Второе достаточное условие локального экстремума.
2. Определённый интеграл от функции f(x) на отрезке [a;b].
3. Первая теорема о среднем для определённого интеграла.
4. Формула для объёма тела, полученного вращением криволинейной трапеции вокруг оси OX.

**Основной вопрос (с доказательством):**

1. Интеграл с переменным верхним пределом, теорема о существовании первообразной у всякой непрерывной функции.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Зачетная работа**

1. Найти длину дуги кривой: 
2. Вычислить площадь 
3. Исследовать на сходимость: 
4. Исследовать на абсолютную и условную сходимость: 
5. Исследовать на непрерывность по каждому аргументу и по совокупности:



1. Исследовать на дифференцируемость:

при и 

1. Найти дифференциалы  для функции  если 
2. Найти  неявной функции  если  где 
3. Разложить по формуле Маклорена до членов 6-го порядка малости: 
4. Определить наибольшее и наименьшее значения функции  в области 

**Вопросы к экзамену**

1. Отыскание точек локального экстремума функции. Достаточные условия экстремума.
2. Направление выпуклости графика функции и точки перегиба. Достаточные условия перегиба.
3. Асимптоты графика функции. Общая схема исследования графика функции.
4. Понятие интегрируемости функции. Леммы Дарбу о верхних и нижних суммах.
5. Необходимое и достаточное условие интегрируемости.
6. Классы интегрируемых функций.
7. Основные свойства определенного интеграла. Оценки интегралов. Формулы среднего значения.
8. Основная формула интегрального исчисления. Формулы замены переменного и интег­рирования по частям.
9. Понятие длины плоской кривой. Формулы для вычисления длины дуги кривой.
10. Понятие квадрируемости (площади) плоской фигуры. Площадь криволинейной трапе­ции и криволинейного сектора.
11. Понятие кубируемости (объем тела). Кубируемость некоторых классов тел.
12. Абсолютная сходимость несобственных интегралов. Формулы замены переменного и интегрирования по частям для несобственных интегралов.
13. Признак Абеля-Дирихле. Главное значение несобственного интеграла.
14. Метод хорд и его обоснование.
15. Метод касательных и его обоснование.
16. Приближенные методы вычисления определенных интегралов (для одного из методов вывести оценку погрешности)
17. Различные множества точек и последовательности точек n-мерного пространства. Тео­рема Больцано-Вейерштрасса.
18. Понятие функции *п* переменных и ее предельного значения.
19. Непрерывность функции *п* переменных. Основные теоремы о непрерывных функциях.
20. Понятие дифференцируемости функции нескольких переменных. Достаточное условие дифференцируемости. Касательная плоскость к поверхности.
21. Дифференцирование сложной функции нескольких переменных. Инвариантность формы первого дифференциала.
22. Производная по направлению. Градиент.
23. Частные производные и дифференциалы высших порядков. Теоремы о равенстве сме­шанных производных.
24. Формула Тейлора с остаточным членом в форме Лагранжа.
25. Формула Тейлора с остаточным членом в форме Пеано.
26. Экстремум функции нескольких переменных и его отыскание.
27. Теорема о существовании и дифференцируемости неявно заданной функции.
28. Теорема о разрешимости системы функциональных уравнений.
29. Понятие зависимости функций. Функциональные матрицы и их роль при исследова­нии зависимости функций.
30. Условный экстремум и методы его отыскания.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Математический анализ III**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

**Вариант 1**

1. Исследовать на сходимость ряд .

2. Исследовать на абсолютную и условную сходимость ряд .

3. Исследовать на абсолютную и условную сходимость бесконечное произведение .

4. Исследовать ряд на равномерную сходимость: .

5. Определить область  существования функции  и исследовать ее на дифференцируемость во внутренних точках .

6. Найти множество сходимости степенного ряда .

**Вариант 2**

1. Исследовать ряд на сходимость: .

2. Исследовать ряд на абсолютную и условную сходимость: .

3. Найти область сходимости функционального ряда .

4. Исследовать на равномерную сходимость на области сходимости

а) ряд , б) последовательность .

5. Исследовать на непрерывность на области существования сумму ряда .

6. Определить радиус сходимости ряда .

7. Разложить в ряд Тейлора по степеням  функцию , указать область сходимости ряда.

**Контрольная работа № 2**

**Вариант 1**

1. Найти .

2. Найти , где  ограничено плоскостями .

3. Найти , где  – часть сферы .

4. Найти поток поля  через полную внешнюю поверхность

тела .

5. Найти циркуляцию поля  вдоль контура .

**Вариант 2**

1. Найти площадь фигуры, ограниченной кривыми:  .

2. Найти объем тела , ограниченного поверхностями:   точка .

3. Найти площадь части поверхности , заключенной внутри

цилиндра .

4. Вычислить криволинейный интеграл пересечение плоскости  и поверхности . Направление обхода – против часовой стрелки, если смотреть с положительной стороны оси .

5. Вычислить поверхностный интеграл   – направляющие косинусы внешней нормали.

6. Найти поток вектора  через поверхность .

**Вопросы к коллоквиуму**

Коллоквиум проводится в форме устного собеседования. В каждый билет входит один теоретический вопрос из списка, а также предлагается дополнительная задача. Темы коллоквиума:

1. Понятие числового ряда. Критерий Коши. Необходимое и достаточное условие сходимости рядов с неотрицательными членами.
2. Признаки сравнения.
3. Признаки Даламбера и Коши, их сравнение.
4. Признак Коши-Маклорена.
5. Теорема Римана о перестановке членов в числовых рядах.
6. Теорема Коши о перестановке членов в числовых рядах.
7. Последовательности с ограниченным изменением и их свойства.
8. Признаки сходимости произвольных числовых рядов (Абеля, Дирихле-Абеля, Лейбница).
9. Теорема Мертенса.
10. Взаимосвязь между сходимостью четырех рядов: повторных, двойного и "одинарного".
11. Метод Чезаро суммирования расходящихся рядов.
12. Метод Пуассона-Абеля суммирования расходящихся рядов.
13. Бесконечные произведения и их свойства.
14. Последовательности с равномерно ограниченным изменением и их свойства.
15. Признаки Абеля равномерной сходимости функциональных рядов.
16. Признак Дини равномерной сходимости функциональных последовательностей и рядов.
17. Непрерывность суммы функционального ряда.
18. Почленное интегрирование функциональных рядов.
19. Почленное дифференцирование функциональных последовательностей.
20. Сходимость в среднем, связь с равномерной сходимостью, теорема о почленном интегрировании.
21. Теорема Арцела.
22. Теорема Коши-Адамара.

**Задачи для коллоквиума**

1. Пусть  и ряд  сходится. Доказать, что ряд  расходится.

2. Привести пример сходящегося ряда , для которого .

3. Привести пример расходящегося ряда , для которого .

4. Пусть  и  сходится абсолютно. Доказать, что сходимость ряда  эквивалентна сходимости интеграла .

5. Пусть последовательность  монотонна, но не является бесконечно малой. Доказать, что ряды , расходятся при всех  .

6. Пусть последовательность  монотонна и является бесконечно малой, причем ряд  расходится. Доказать, что ряды  и  сходятся условно при всех  .

7. Пусть ряды ,  сходятся условно, а их произведение по Коши  сходится. Доказать, что .

8. Доказать, что любую последовательность с ограниченным изменением можно представить в виде разности двух монотонных ограниченных последовательностей.

9. Для любого множества , обозначим   , . Проверить, что для двойных сумм имеет место *преобразование Харди*: , где . В качестве применения исследовать на сходимость двойной ряд  .

10. Пусть  – произвольное множество и последовательность  непрерывных на  функций сходится равномерно на . Доказать, что она сходится равномерно на .

11. Может ли последовательность разрывных на  функций равномерно сходиться на  к непрерывной функции?

12. Может ли последовательность непрерывных на  функций равномерно сходиться на  к разрывной функции?

13. Привести пример двух последовательностей , равномерно сходящихся на  таких, что последовательность  сходится на  неравномерно.

14. Показать, что последовательность гладких функций   равномерно сходится на , а последовательность расходится в каждой точке .

15. Исследовать последовательность  на равностепенную непрерывность на множестве , где: а) ; б) .

16. Найти сумму функционального ряда .

17. Просуммировать ряд  методом Пуассона-Абеля.

18. Пусть ряд  суммируем методом Чезаро и  при . Доказать, что ряд  сходится.

19. Доказать, что если ряд  суммируем методом Пуассона-Абеля, то для любого  имеем  при .

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Работа для проведения зачетной комиссии**

1. Исследовать на абсолютную и условную сходимость относительно параметра :

.

2. Разложить в степенной ряд по степеням , определить область сходимости:

.

3. Исследовать функцию на непрерывность: .

4. Найти объём тела: .

5. Найти:  где .

6. Найти поток  через часть  поверхности , вырезаемую плоскостью :



(нормаль – внешняя к замкнутой поверхности, образуемой данными поверхностями,  – *незамкнутая* поверхность).

**Вопросы к экзамену**

Экзамен сдается в устной форме. В экзаменационном билете – один вопрос из приведенного ниже списка.

1. Понятие числового ряда. Критерий Коши. Необходимое и достаточное условие сходимости рядов с неотрицательными членами.
2. Признаки сходимости рядов с неотрицательными членами (признаки сравнения, Даламбера, Коши, Коши-Маклорена).
3. Теоремы Коши и Римана о перестановке членов в числовых рядах.
4. Признаки сходимости произвольных числовых рядов (два признака Абеля, признаки Дирихле-Абеля, Лейбница).
5. Арифметические операции над сходящимися числовыми рядами. Теорема Мертенса.
6. Бесконечные произведения, критерии их сходимости.
7. Необходимое условие сходимости двойного ряда. Связь между сходимостью двойного ряда и повторного ряда. Критерий сходимости двойного ряда с неотрицательными членами.
8. Абсолютная сходимость двойного ряда. Взаимосвязь между сходимостью четырех рядов: повторных, двойного и "одинарного".
9. Обобщенные методы суммирования расходящихся рядов (методы Чезаро и Пуассона-Абеля).
10. Функциональные последовательности и ряды. Равномерная сходимость. Критерий Коши.
11. Признаки равномерной сходимости функциональных рядов (два признака Абеля, признаки Дирихле-Абеля, Вейерштрасса).
12. Признак Дини равномерной сходимости функциональных рядов и последовательностей. Почленный переход к пределу, непрерывность предельной функции функциональных последовательностей и рядов.
13. Почленное дифференцирование, существование первообразных функций для функциональных последовательностей и рядов.
14. Почленное интегрирование функциональных последовательностей и рядов (две теоремы). Сходимость в среднем, связь с равномерной сходимостью.
15. Теорема Арцела. Признак равностепенной непрерывности функциональной последовательности.
16. Степенные ряды. Теорема Коши-Адамара. Непрерывность суммы, почленное интегрирование и дифференцирование степенного ряда. Разложение функций в степенные ряды.
17. Определение и доказательство существования двойного интеграла при помощи прямоугольных разбиений области. Классы интегрируемых функций. Основные свойства двойного интеграла.
18. Определение двойного интеграла при помощи произвольных разбиений области. Эквивалентность двух определений.
19. Сведение двойного интеграла к повторному однократному.
20. Кратные несобственные интегралы от неотрицательных функций. Признаки сходимости.
21. Кратные несобственные интегралы от знакопеременных функций. Эквивалентность понятий сходимости и абсолютной сходимости.
22. Криволинейные интегралы первого и второго рода.
23. Понятие поверхности. Нормаль и касательная плоскость к поверхности. Лемма о проекции окрестности точки на касательную плоскость.
24. Площадь поверхности. Квадрируемость поверхности.
25. Поверхностные интегралы первого и второго рода.
26. Преобразование базисов. Инварианты линейного оператора.
27. Дивергенция, ротор и производная по направлению векторного поля. Повторные операции теории поля.
28. Формула Грина. Формула Остроградского-Гаусса.
29. Формула Стокса.
30. Условия независимости криволинейного интеграла второго рода на плоскости от пути интегрирования.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Действительный и комплексный анализ**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

**Вариант 1**

1. Исследовать на равномерную сходимость на области существования интегралы:

а) ; б) .

2. Исследовать интеграл  на непрерывность на области существования.

3. Вычислить интеграл .

4. Определить область существования интеграла  и вычислить этот интеграл.

**Вариант 2**

1. Найти , если .

2. Исследовать на равномерную сходимость  в случаях: a) ; б) .

3. Исследовать на непрерывность .

.

4. Вычислить . Обосновать вычисление.

5. Вычислить .

**Контрольная работа № 2**

**Вариант 1**

1. Разложить в ряд Фурье на отрезке  функцию , нарисовать график суммы ряда и исследовать ряд на равномерную сходимость на .

2. Разложить в ряд Фурье по косинусам функцию ,

нарисовать график суммы ряда и исследовать ряд на равномерную сходимость на .

**Вариант 2**

1. Разложить функцию  по косинусам кратных дуг, нарисовать графики функции  и суммы ряда Фурье.

2. Разложить в ряд Фурье функцию , нарисовать графики функции  и суммы ряда Фурье.

**Контрольная работа № 3**

**Вариант 1**

1. Разложить функцию  в ряд Лорана по степеням  в кольце , содержащем точку . Указать границы кольца . .

2. Найти все особые точки функции  и определить их вид: .

3. Применяя теорию вычетов, вычислить интегралы

а) ; б) .

4. Отобразить конформно область  на верхнюю полуплоскость.

**Вариант 2**

1. Найти множество точек , в которых функция  является дифференцируемой.

2. Разложить функцию  в ряд Тейлора с центром в точке  и указать область, где справедливо разложение.

3. Разложить функцию  в ряд Лорана в кольце .

4. Определить все особые точки функции  и классифицировать их, включая точку .

5. Вычислить интеграл .

6. Вычислить интеграл .

7. Конформно отобразить на верхнюю полуплоскость внутренность угла  с выброшенным лучом .

**Вопросы к коллоквиуму**

Коллоквиум проводится в форме устного собеседования. В каждый билет входит один теоретический вопрос из списка, а также предлагается дополнительная задача. Темы коллоквиума:

**Вопросы к коллоквиуму по теме "Интегралы, зависящие от параметра"**

1. Собственные интегралы, зависящие от параметра (ИЗП). Случай постоянных пределов интегрирования.

2. Собственные ИЗП. Случай переменных пределов интегрирования.

3. Равномерная сходимость несобственных ИЗП. Примеры. Критерий Коши.

4. Признаки равномерной сходимости несобственных ИЗП (Вейерштрасса, Дирихле-Абеля, Дини).

5. Непрерывность и интегрируемость несобственных ИЗП на отрезке.

6. Дифференцируемость несобственных ИЗП.

7. Интегрируемость несобственных ИЗП на полупрямой.

8. Вычисление интеграла Дирихле.

9. Свойства Г-функции Эйлера.

10. Свойства В-функции Эйлера. Связь между эйлеровыми интегралами.

11. Вывод асимптотической формулы для интеграла .

12. Асимптотическая формула для функции . Формула Стирлинга.

**Задачи для коллоквиума**

1. Пусть . Доказать, что

.

2. Доказать, что  сходится неравномерно по , но замена  превращает его в равномерно сходящийся интеграл.

3. Задана функция 

Показать, что функция  определена и непрерывна на , функция  интегрируема на  для любого , но .

4. Пусть функция  непрерывна на  и интеграл  сходится равномерно на . Доказать, что этот интеграл сходится

равномерно на .

5. Задана функция  Показать, что интеграл  сходится равномерно на , а интеграл  сходится неравномерно на .

6. Задана функция  Показать, что при любом  функция  монотонна на  и ,

но интеграл  сходится неравномерно на . Какое условие признака Дирихле нарушено?

7. Пусть интеграл  сходится равномерно на множестве  и для любого  функция  монотонна на  и стремится к нулю при . Доказать, что  при  сходится к нулю равномерно на .

8. Доказать, что интеграл сходится неравномерно на , а ряд  сходится равномерно на .

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Работа для проведения зачетной комиссии**

1. Разложить функцию  в тригонометрический ряд Фурье в интервале . К чему сходится полученное выражение в точке ?

2. Обосновать возможность дифференцирования под знаком интеграла и вычислить интеграл: .

3. Исследовать на равномерную сходимость на множестве: .

4. Разложить в ряд Лорана на указанном множестве .

5. Применить методы ТФКП для вычисления интеграла, обосновать применимость метода: .

6. Отобразить конформно сектор  на .

**Вопросы к экзамену**

Экзамен сдается в устной форме. В экзаменационном билете – два вопроса из приведенного ниже списка (по одному из каждого раздела).

**Действительный анализ**

1. Собственные интегралы, зависящие от параметра (ИЗП).

2. Признаки равномерной сходимости несобственных ИЗП (Вейерштрасса, Дирихле-Абеля, Дини).

3. Непрерывность и интегрируемость несобственных ИЗП на отрезке.

4. Дифференцируемость несобственных ИЗП.

5. Интегрируемость несобственных ИЗП на полупрямой.

6. Вычисление интеграла Дирихле.

7. Свойства Г-функции Эйлера.

8. Свойства В-функции Эйлера. Связь между эйлеровыми интегралами.

9. Асимптотическая формула для функции . Формула Стирлинга.

10. Ортонормированные системы. Задача о наилучшем приближении элемента евклидова пространства.

11. Замкнутость и полнота ортонормированных систем.

12. Теорема Фейера.

13. Замкнутость тригонометрической системы. Следствия из замкнутости.

14. Теоремы Вейерштрасса о равномерном приближении непрерывной функции.

15. Локальная теорема Фейера.

16. Простейшие условия равномерной сходимости и почленной дифференцируемости рядов Фурье.

17. Уточнённые условия равномерной сходимости ряда Фурье.

18. Условие сходимости тригонометрического ряда Фурье в точке. Сходимость ряда Фурье кусочно-гельдеровой функции.

19. Принцип локализации Римана.

20. Свойства преобразования Фурье.

21. Условия разложимости функции в интеграл Фурье.

**Комплексный анализ**

1. Стереографическая проекция.

2. Функции комплексного переменного. Предел. Непрерывность.

3. Дифференцируемость функций комплексного переменного. Аналитичность.

4. Теорема Коши и её обобщение.

5. Интегральная формула Коши.

6. Принцип максимума модуля аналитической функции.

7. Гармонические функции и их свойства. Принцип максимума.

8. Разложение гармонических функций в ряды. Ряд Фурье для гармонической функции.

9. Бесконечная дифференцируемость аналитических функций. Теорема Лиувилля.

10. Неопределённый интеграл. Теорема Морера.

11. Равномерно сходящиеся ряды аналитических функций.

12. Аналитичность суммы степенного ряда. Теорема Тейлора.

13. Теорема единственности аналитических функций. Нули аналитической функции.

14. Ряды Лорана. Теорема Лорана.

15. Классификация изолированных особых точек. Устранимая особая точка. Полюс.

16. Существенно особая точка. Теорема Сохоцкого-Вейерштрасса.

17. Вычет аналитической функции в изолированной особой точке. Основная теорема о вычетах.

18. Вычисление интегралов с помощью вычетов. Лемма Жордана.

19. Логарифмический вычет. Теорема Руше. Принцип аргумента.

20. Аналитическое продолжение с вещественной оси. Элементарные функции.

21. Аналитическое продолжение с помощью рядов и через границу. Принцип непрерывности.

22. Аналитическое продолжение Гамма-функции Эйлера. Формула дополнения.

23. Основные принципы конформных отображений: принцип соответствия границ и принцип симметрии Римана-Шварца.

24. Свойство аналитической однолистной функции в области.

25. Локальное свойство однолистной функции. Отображение области на область при конформном отображении.

26. Дробно-линейная функция и её свойства.

27. Конформные отображения, осуществляемые элементарными функциями.

28. Задача Дирихле для уравнения Лапласа. Случай круга и верхней полуплоскости.

29. Следствие из решения задачи Дирихле для круга. Теорема Вейерштрасса о приближении непрерывной функции многочленами.

30. Функция Грина (функция источника).

31. Преобразование Лапласа и его основные свойства.

32. Решение обыкновенных дифференциальных уравнений и уравнений в частных производных с помощью преобразования Лапласа.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дискретная математика**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольные работы**

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №1**

**1**. Найти фиктивные переменные функции (1111 1111 1100 1100) и удалить их.

**2**. Построить совершенную ДНФ, совершенную КНФ и полином Жегалкина для функции (0100 0011).

**3**. Полны ли системы: а) {(1000 1001),(1010 1010)},

б) {m(x, y, z), x\_1+ x\_2 +…+ x\_7};

в) {[xy+ (y → z)] → (x V z), x → z};

г) (M \ T0) в объединении с (S \ L).

**4**. Выделить все базисы из системы {0, x + y, x → y, (xy) + (xz)}.

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №2**

1. Написать код заданного корневого дерева. Полученное двоичное число перевести в десятичную систему.
2. В дереве 50 вершин степени 1, остальные вершины степени 2 и 8. Сколько вершин степени 8.
3. В геометрической реализации планарного графа с 20 вершинами на плоскости все грани 5-угольники. Сколько в этом графе ребер?
4. Нарисовать все попарно неизоморфные непланарные графы с 6 вершинами и 11 ребрами.

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №3**

**1**. Существуют ли взаимно однозначные двоичные коды с длинами кодовых слов: а) (2,2,2,3,3,4); б) (1,2,3,3)? Если да, то построить такой код.

**2**. Является ли код однозначно декодируемым: {01, 011, 100, 2100, 101210, 001210}?

**3**. Построить оптимальный двоичный код для вероятностей: {0.5; 0.2; 0.1; 0.09; 0.08; 0.03}.

**4**. В 15-разрядном коде Хэмминга получено сообщение с не более чем 1 ошибкой: (011000101110100). Что передавалось?

**5**. Линейный код порождается векторами: (1001001), (1100010), (0111101), (0010110).

а) Сколько слов в коде?

б) Найдите кодовое расстояние кода.

в) Сколько ошибок этот код может обнаруживать и сколько исправлять?

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №4**

**1**. Построить схему над базисным множеством из элемента единичной задержки и элементов, реализующих булевы функции «дизъюнкция» и «отрицание», для автомата с каноническими уравнениями:

y1(t) = x1(t)∙q1(t-1), y2(t) = (x2(t)+1) V q2(t-1),

q1(t) = x1(t) V q1(t-1), q2(t) = (x2(t)+1) ∙ q2(t-1), q1(0) = 0, q2(0) = 1.

**2**. Построить диаграмму Мура, каноническую таблицу и канонические уравнения для автомата, преобразующего последовательность x(1)x(2)…x(t)… из нулей и единиц в последовательность y(1)y(2)…y(t)… , где y(1)=0 и y(t)=x(t-1) V x(t) при t > 1.

**3**. Построить диаграмму Мура для автомата, преобразующего последовательность x(1)x(2)…x(t)… из нулей и единиц в последовательность y(1)y(2)…y(t)… , причем y(t)=1, если t – четно и в последовательности x(1)x(2)…x(t) встречаются две единицы подряд, и y(t)=0 иначе.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**ВОПРОСЫ К ЭКЗАМЕНУ**

В билете 2 вопроса (один из части А и один из части В) и задача.

**Часть А** – ответ без подготовки, по любым материалам (конспекты, книжки, распечатки лекций и т.д.). Проверяется, насколько осознаны все доказательства (основной вопрос – «почему?»). Определения и формулировки – без конспектов.

1. Сокращенная дизъюнктивная нормальная форма. Метод ее построения по конъюнктивной нормальной форме (метод Нельсона).
2. Алгоритм построения вектора коэффициентов полинома Жегалкина (с обоснованием).
3. Двойственность. Класс самодвойственных функций, его замкнутость.
4. Лемма о нелинейной функции.
5. Теорема Поста о полноте системы функций алгебры логики.
6. Теорема о предполных классах.
7. Теоремы о представлении k-значных функций 2-й формой и полиномами.
8. Деревья. Свойства деревьев.
9. Алгоритм построения кратчайшего остовного дерева (с обоснованием).
10. Теорема о раскраске планарных графов в 5 цветов.
11. Алгоритм распознавания взаимной однозначности алфавитного кодирования (с обоснованием). Теорема Маркова.
12. Неравенство Макмиллана.
13. Существование префиксного кода с заданными длинами кодовых слов.
14. Теорема редукции.
15. Коды с исправлением *r* ошибок. Оценка функции .
16. Коды Хэмминга. Оценка функции .
17. Метод Карацубы построения схемы для умножения, верхняя оценка ее сложности.
18. Схемы из функциональных элементов и элементов задержки. Автоматность осуществляемых ими отображений.
19. Моделирование автоматной функции схемой из функциональных элементов и элементов задержки.
20. Теорема Мура. Пример автомата, на котором достигается оценка теоремы Мура.

**Часть В** – ответ без конспектов и почти без подготовки (3-5 минут), с доказательствами (можно излагать устно).

1. Функции алгебры логики. Равенство функций. Тождества для элементарных функций.
2. Теорема о разложении функции алгебры логики по переменным. Теорема о совершенной дизъюнктивной нормальной форме.
3. Полные системы. Примеры полных систем (с доказательством полноты).
4. Теорема Жегалкина о представимости функции алгебры логики полиномом.
5. Понятие замкнутого класса. Замкнутость классов 
6. Класс монотонных функций, его замкнутость.
7. Лемма о несамодвойственной функции.
8. Лемма о немонотонной функции.
9. Теорема о максимальном числе функций в базисе в алгебре логики.
10. k-значные функции. Теорема о существовании конечной полной системы в *Pk*.
11. Основные понятия теории графов. Изоморфизм графов. Связность.
12. Корневые деревья. Верхняя оценка их числа.
13. Геометрическая реализация графов. Теорема о реализации графов в трехмерном пространстве.
14. Планарные (плоские) графы. Формула Эйлера.
15. Доказательство непланарности графов *K*5 и *K*3,3. Теорема Понтрягина-Куратовского (доказательство в одну сторону).
16. Теорема о раскраске вершин графа в 2 цвета (теорема Кенига).
17. Оптимальные коды, их свойства.
18. Линейные двоичные коды. Теорема о кодовом расстоянии линейных кодов.
19. Схемы из функциональных элементов. Реализация функций алгебры логики схемами.
20. Сумматор. Верхняя оценка сложности сумматора. Вычитатель.
21. Понятие автоматных функций, их представление диаграммой Мура. Единичная задержка.
22. Несуществование эксперимента, определяющего начальное состояние автомата.

По результатам контрольных работ по каждой из четырех тем (алгебра логики, графы, коды, автоматы) у каждого студента должна стоять одна из трех оценок — 0, 1/2 или 1. Оценка 0 означает, что на экзамене студент должен решить *дополнительную* задачу по данной теме, оценка 1/2, — что студент решает задачу по данной теме только в случае, если она выпадает *в билете*. Оценка 1 означает, что на экзамене студент не должен решать по данной теме как дополнительны*е* задачи, так и задачу из билета. Дополнительные задачи решаются до выбора билета. Студенты, не решившие достаточное количество *дополнительных* задач, удаляются с экзамена с оценкой «неудовлетворительно», количество решенных задач может ограничить сверху оценку, получаемую на экзамене.

Задачи решаются без конспектов.

После ответа на билет возможна прогонка по всему материалу (определения, формулировки, идеи доказательств) и *добавочные* задачи на любые темы (не путать с *дополнительными*!).

**Типовые задачи для экзамена.**

|  |  |
| --- | --- |
| 1. Представить функцию алгебры логики f(x, y, z) = (1010101010110110) полиномом Жегалкина.    2. Выяснить, полна ли система функций в P2:  {0, x→y, xy~xz}.  3. Построить упорядоченное корневое дерево по его коду: (01001000110111).  4. Нарисовать все попарно неизоморфные непланарные графы с 6 вершинами и 10 ребрами.  5. Построить оптимальный двоичный код для набора вероятностей: (0,5; 0,2; 0,1; 0,09; 0,08; 0,03). |  |

6. Построить по методу Хэмминга кодовое слово для сообщения: 01110111011.

7. Построить диаграмму Мура и канонические уравнения автомата, реализующего преобразование:

8. Для автомата, заданного каноническими уравнениями:

построить схему из элементов дизъюнкции, отрицания и задержки.

**Экзаменационный билет** состоит из двух вопросов и задачи, например:

1. Двойственность. Класс самодвойственных функций, его замкнутость.
2. Оптимальные коды, их свойства.
3. Задача по графам.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Практикум на ЭВМ**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Типовые задачи теоретического зачёта.**

1. Написать программу для учебной машины УМ-3. Эта программа должна сначала вводить целочисленный массив X длины 100, затем печатать число S, равное количеству одновременно отрицательных и кратных трём элементов массива X. При записи кодов операций использовать мнемонические обозначения.

2. Выписать вид внутреннего машинного представления целой переменной X (в двоичном или шестнадцатеричном виде):

X **dw** -1023

3. Написать полную программу на Ассемблере, которая вводит (по **inint**) целое знаковое число X в формате **dd** и выводит (по **outword**) число значащих *чётных* цифр (т.е. '0', '2', '4', '6', '8')в десятичной записи значения числа X. Цифра является значащей, если её удаление меняет величину числа.

4. Написать полную программу на Ассемблере, которая вводит текст до точки и выводит (по **outword**) сумму *нечётных цифр*, расположенных в этом тексте после первой "\*". Считать, что таких цифр не более MaxLongword.

5. Пусть на Паскале дано описание типа массива:

**const** n=5000; **type** MAS=**array**[1..n,1..n] **of** char;

Написать на Ассемблере *процедуру* со стандартными соглашениями о связях, которая получает в качестве параметров адрес массива типа MAS и длину массива N. Процедура должна все символы-*цифры* на главной диагонали этой матрицы заменить на символы "+". Привести пример вызова этой процедуры.

6. Написать макрос с заголовком

First\_1 **macro** X

параметр которого может быть только форматов m8, m16 или m32. Макрос переставляет в начало все 1 (биты со значением единица)   
во внутреннем машинном представлении X, например, для X **=** 10101010b необходимо получить X **=** 11110000b. Макроопределение должно настраиваться на тип параметра..

7. Написать на Ассемблере неголовной модуль, содержащий описание процедуры без параметров с именем Del\_3. Эта процедура должна уменьшать в три раза значение знаковой переменной размером в слово (**dw**) с именем Perem3, описанной в каком-то другом модуле.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Обыкновенные дифференциальные уравнения**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Решить уравнение и найти особые решения, если они есть:  .  2. Решить задачу Коши:  Решить уравнения:  3. .  4. .  5. . | 1. Решить уравнение и найти особые решения, если они есть:  .  2. Решить задачу Коши:  Решить уравнения:  3. .  4. .  5.. |

**Контрольная работа № 2**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| Найти решения линейных ОДУ и их систем:  1.  2.  3.  4.  5. | Найти решения линейных ОДУ и их систем:  1.  2.  3.  4.  5. |

**Контрольная работа № 3**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Найти  :  2. Решить систему нелинейных ОДУ  3. Исследовать на устойчивость:  4. Найти *a* и *b*, при которых асимптотически устойчиво нулевое  решение уравнения  5. Исследовать на устойчивость все положения равновесия системы | 1. Найти  :  2. Решить систему нелинейных ОДУ  3. Исследовать на устойчивость:  4. Найти *a* и *b*, при которых асимптотически устойчиво нулевое  решение уравнения  5. Исследовать на устойчивость все положения равновесия системы |

**Контрольная работа № 4**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Изобразить эскиз траекторий решений системы в окрестности  положения равновесия системы ОДУ  2. Решить краевую задачу  3. Построить функцию Грина:  4. Решить задачу Коши для ДУ в частных производных 1-го порядка    5. Найти стационарные кривые функционала | 1. Изобразить эскиз траекторий решений системы в окрестности  положения равновесия системы ОДУ  2. Решить краевую задачу  3. Построить функцию Грина:  4. Решить задачу Коши для ДУ в частных производных 1-го порядка    5. Найти стационарные кривые функционала |

**Вопросы к коллоквиуму.**

1. Понятие дифференциального уравнения, примеры. Редукция ОДУ n-го прядка, разрешенного относительно старшей производной, к нормальной системе ОДУ. Определение решения общего ОДУ n-го прядка и его интегральной кривой. Определение решения, интегральной кривой и фазовой траектории нормальной системы ОДУ, примеры.
2. Примеры математических моделей, использующих дифференциальные уравнения: движение материальной точки в пространстве под действием силы, зависящей от времени, положения точки и ее скорости; динамика популяций в рамках модели «хищник-жертва».
3. ОДУ 1 порядка в симметричном виде, определение параметрического решения. Интеграл и общий интеграл, примеры. Уравнения в полных дифференциалах (УПД). Теорема об общем интеграле УПД.
4. Уравнения в полных дифференциалах (УПД). Теорема о необходимом и достаточном условии того, что ОДУ в симметричном виде является УПД.
5. Уравнения в полных дифференциалах. Интегрирующий множитель. Теорема о существовании интегрирующего множителя.
6. Лемма Гронуолла-Беллмана.
7. Постановка задачи Коши для ОДУ 1 порядка, разрешенного относительно производной. Лемма о редукции этой задачи к интегральному уравнению. Условие Липшица по переменной для скалярной функции . Теорема о единственности решения задачи Коши для ОДУ 1 порядка, разрешенного относительно производной.
8. Теорема о существовании решения задачи Коши для ОДУ 1 порядка, разрешенного относительно производной.
9. Постановка задачи Коши для ОДУ 1 порядка, не разрешенного относительно производной, примеры. Теорема о существовании и единственности решения задачи Коши для ОДУ 1 порядка, не разрешенного относительно производной. Особое решение ОДУ 1-го порядка, примеры.
10. Постановка задачи Коши для нормальной системы ОДУ. Условие Липшица по переменным для функции . Теорема о единственности решения задачи Коши для нормальной системы ОДУ.
11. Теорема о существовании решения задачи Коши для нормальной системы ОДУ на произвольном отрезке.
12. Постановка задачи Коши для ОДУ n-го порядка, разрешенного относительно старшей производной. Теорема о существовании и единственности решения этой задачи на произвольной отрезке.
13. Постановка задач Коши для линейного ОДУ n-го порядка и линейной системы ОДУ. Теоремы о существовании и единственности решения этих задач на произвольной отрезке.
14. Линейная зависимость и независимость скалярных функций. Определитель Вронского и его свойства. Примеры. Теорема об альтернативе для определителя Вронского для решений однородного линейного ОДУ n-ого порядка.
15. Фундаментальная система решений линейного ОДУ n-ого порядка. Теорема о существовании ФСР. Теорема об общем решении однородного линейного ОДУ n-ого порядка.
16. Теорема об общем решении неоднородного линейного ОДУ n-ого порядка. Метод вариации постоянных.
17. Теорема о построении ФСР однородного линейного ОДУ n-ого порядка с постоянными коэффициентами. Пример построения однородного линейного ОДУ с постоянными коэффициентами по заданным решениям.
18. Теорема о единственности однородного линейного ОДУ n-ого порядка, имеющего заданную ФСР.
19. Теорема о построении однородного линейного ОДУ n-ого порядка, имеющего заданный набор решений, пример. Формула Остроградского-Лиувилля.
20. Линейная зависимость и независимость векторных функций. Определитель Вронского и его свойства. Примеры. Теорема об альтернативе для определителя Вронского для решений однородной линейной системы ОДУ.
21. Фундаментальная система решений однородной линейной системы ОДУ. Фундаментальная матрица. Теорема о существовании ФСР. Теорема об общем решении однородной линейной системы ОДУ.
22. Теорема об общем решении неоднородной линейной системы ОДУ. Матрицант. Теорема о частном решении неоднородной линейной системы ОДУ (метод вариации постоянных).
23. Теорема о построении ФСР однородной линейной системы ОДУ n-ого порядка с постоянными коэффициентами в случае существования n линейно независимых собственных векторов матрицы системы.
24. Теорема о построении ФСР однородной линейной системы ОДУ n-ого порядка с постоянными коэффициентами в случае отсутствия n линейно независимых собственных векторов матрицы системы.

**Билет для коллоквиума** содержит 4 вопроса, например:

1. Сформулировать и доказать теорему о необходимом и достаточном условии того, что обыкновенное дифференциальное уравнение 1-го порядка в симметричном виде является уравнением в полных дифференциалах.
2. Сформулировать теорему об альтернативе для определителя Вронского для решений линейной однородной системы обыкновенных дифференциальных уравнений.
3. Сформулировать постановку задачи Коши для обыкновенного дифференциального уравнения n-го порядка, разрешенного относительно старшей производной.
4. Функции , ,  являются решениями линейного однородного обыкновенного дифференциального уравнения второго порядка . Исследовать их на линейную зависимость на отрезке  и объяснить результат.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Зачетная работа**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| Решить уравнения  1.  2.  3.  4.  5.  6.  Решить системы уравнений  7.  8. | Решить уравнения  1.  2.  3.  4.  5.  6.  Решить системы уравнений  7.  8. |

**Вопросы к экзамену**.

1. Понятие дифференциального уравнения, примеры. Редукция ОДУ n-го прядка, разрешенного относительно старшей производной, к нормальной системе ОДУ. Определение решения общего ОДУ n-го прядка и его интегральной кривой. Определение решения, интегральной кривой и фазовой траектории нормальной системы ОДУ, примеры.
2. Примеры математических моделей, использующих дифференциальные уравнения: движение материальной точки в пространстве под действием силы, зависящей от времени, положения точки и ее скорости; динамика популяций в рамках модели «хищник-жертва».
3. ОДУ 1 порядка в симметричном виде, определение параметрического решения. Интеграл и общий интеграл, примеры. Уравнения в полных дифференциалах (УПД). Теорема об общем интеграле УПД.
4. Уравнения в полных дифференциалах (УПД). Теорема о необходимом и достаточном условии того, что ОДУ в симметричном виде является УПД.
5. Уравнения в полных дифференциалах. Интегрирующий множитель. Теорема о существовании интегрирующего множителя.
6. Лемма Гронуолла-Беллмана. Условие Липшица для скалярной функции от 1-й переменной. Примеры, иллюстрирующие соотношения между множествами липшицевых, непрерывных и дифференцируемых функций; поведение липшицевых функций на бесконечности.
7. Постановка задачи Коши для ОДУ 1 порядка, разрешенного относительно производной. Лемма о редукции этой задачи к интегральному уравнению. Условие Липшица по переменной для скалярной функции . Теорема о единственности решения задачи Коши для ОДУ 1 порядка, разрешенного относительно производной.
8. Теорема о существовании решения задачи Коши для ОДУ 1 порядка, разрешенного относительно производной.
9. Постановка задачи Коши для ОДУ 1 порядка, не разрешенного относительно производной, примеры. Теорема о существовании и единственности решения задачи Коши для ОДУ 1 порядка, не разрешенного относительно производной. Особое решение ОДУ 1-го порядка, примеры.
10. Постановка задачи Коши для нормальной системы ОДУ. Условие Липшица по переменным для функции . Теорема о единственности решения задачи Коши для нормальной системы ОДУ.
11. Теорема о существовании решения задачи Коши для нормальной системы ОДУ на произвольном отрезке.
12. Постановка задачи Коши для ОДУ n-го порядка, разрешенного относительно старшей производной. Теорема о существовании и единственности решения этой задачи на произвольной отрезке.
13. Постановка задач Коши для линейного ОДУ n-го порядка и линейной системы ОДУ. Теоремы о существовании и единственности решения этих задач на произвольной отрезке.
14. Линейная зависимость и независимость скалярных функций. Определитель Вронского и его свойства. Примеры. Теорема об альтернативе для определителя Вронского для решений однородного линейного ОДУ n-ого порядка.
15. Фундаментальная система решений линейного ОДУ n-ого порядка. Теорема о существовании ФСР. Теорема об общем решении однородного линейного ОДУ n-ого порядка.
16. Теорема об общем решении неоднородного линейного ОДУ n-ого порядка. Метод вариации постоянных.
17. Теорема о построении ФСР однородного линейного ОДУ n-ого порядка с постоянными коэффициентами. Пример построения однородного линейного ОДУ с постоянными коэффициентами по заданным решениям.
18. Теорема о единственности однородного линейного ОДУ n-ого порядка, имеющего заданную ФСР.
19. Теорема о построении однородного линейного ОДУ n-ого порядка, имеющего заданный набор решений, пример. Формула Остроградского-Лиувилля.
20. Линейная зависимость и независимость векторных функций. Определитель Вронского и его свойства. Примеры. Теорема об альтернативе для определителя Вронского для решений однородной линейной системы ОДУ.
21. Фундаментальная система решений однородной линейной системы ОДУ. Фундаментальная матрица. Теорема о существовании ФСР. Теорема об общем решении однородной линейной системы ОДУ.
22. Теорема об общем решении неоднородной линейной системы ОДУ. Матрицант. Теорема о частном решении неоднородной линейной системы ОДУ (метод вариации постоянных).
23. Теорема о построении ФСР однородной линейной системы ОДУ n-ого порядка с постоянными коэффициентами в случае существования n линейно независимых собственных векторов матрицы системы. Обоснование возможности перехода к действительнозначной ФСР в случае вещественной матрицы системы.
24. Теорема о построении ФСР однородной линейной системы ОДУ n-ого порядка с постоянными коэффициентами в случае отсутствия n линейно независимых собственных векторов матрицы системы.
25. Теорема о зависимости от правой части и начального условия решения задачи Коши для ОДУ 1-го порядка, разрешенного относительно производной. Теорема о непрерывной зависимости от параметра решения задачи Коши для ОДУ 1-го порядка, разрешенного относительно производной.
26. Теорема сравнения решений задач Коши для ОДУ 1-го порядка, разрешенного относительно производной (неравенство Чаплыгина).
27. Теорема о дифференцируемости по параметру решения задачи Коши для ОДУ 1-го порядка, разрешенного относительно производной. Метод малого параметра.
28. Основные понятия теории устойчивости, примеры. Редукция общей задачи к задаче для нулевого решения.
29. Лемма об устойчивости нулевого решения однородной линейной системы.
30. Теорема об устойчивости нулевого решения однородной линейной системы ОДУ с постоянными коэффициентами. Теорема об устойчивости по первому приближению (первый метод Ляпунова, только формулировка).
31. Положительно определенные функции и их свойства, примеры. Функция Ляпунова для нормальной системы ОДУ.
32. Теоремы об устойчивости и асимптотической устойчивости нулевого решения нормальной системы ОДУ (второй метод Ляпунова). Пример.
33. Теорема Четаева о неустойчивости нулевого решения нормальной системы ОДУ. Пример.
34. Точки покоя (положения равновесия) нормальной автономной системы ОДУ. Классификация точек покоя (с эскизами фазовых траекторий и обоснованием эскиза узла) линейной однородной системы ОДУ 2-го порядка с постоянными коэффициентами и невырожденной матрицей. Грубые точки покоя, поведение фазовых траекторий нормальной автономной системы ОДУ 2-го порядка в окрестности грубой точки покоя.
35. Постановка краевой задачи для линейного ОДУ 2-го порядка, редукция к дивергентному виду и однородным краевым условиям. Тождество Лагранжа, формула Грина, следствия из них.
36. Определение функции Грина краевой задачи для линейного ОДУ 2-го порядка, теорема о существовании и единственности функции Грина.
37. Теорема о представлении решения краевой задачи для линейного ОДУ 2-го порядка через функцию Грина.
38. Задача Штурма-Лиувилля, теоремы о свойствах собственных значений и собственных функций. Теорема Стеклова (только формулировка).
39. Первые интегралы (ПИ) нормальной системы ОДУ, лемма о производной в силу системы. Геометрический смысл ПИ, теорема о представлении решения задачи Коши для нормальной системы ОДУ с помощью функционально независимых ПИ.
40. Линейное однородное уравнение в частных производных (УЧП) 1-го порядка и соответствующая ему система характеристик. Теорема о связи между решениями линейного однородного УЧП 1-го порядка и первыми интегралами системы характеристик. Теорема об общем решении линейного однородного УЧП 1-го порядка.
41. Квазилинейное неоднородное УЧП 1-го порядка и соответствующая ему система характеристик Теорема о связи между решениями квазилинейного неоднородного УЧП 1-го порядка и первыми интегралами системы характеристик.
42. Квазилинейное неоднородное УЧП 1-го порядка и соответствующая ему система характеристик. Теорема о геометрическом смысле квазилинейного УЧП 1-го порядка.
43. Определение функционала, локального экстремума функционала, допустимой вариации функции, вариации функционала. Теорема о необходимом условии экстремума функционала.
44. Основная лемма вариационного исчисления. Теорема о необходимом условии экстремума функционала вида , уравнение Эйлера.
45. Основная лемма вариационного исчисления. Теорема о необходимом условии экстремума функционала вида .
46. Основная лемма вариационного исчисления в двумерном случае. Теорема о необходимом условии экстремума для функционала вида .
47. Вариационная задача на условный экстремум, теорема о необходимом условии экстремума в этой задаче.
48. Вариационное свойство собственных функций и собственных значений задачи Штурма-Лиувилля.

**Типовые задачи для экзамена**.

|  |  |
| --- | --- |
| 1. Найти  :  2. Решить систему нелинейных ОДУ  3. Исследовать на устойчивость:  4. Найти *a* и *b*, при которых асимптотически устойчиво нулевое решение уравнения  5. Исследовать на устойчивость все положения равновесия системы    6. Изобразить эскиз траекторий решений системы в окрестности положения равновесия системы ОДУ  7. Решить краевую задачу  8. Построить функцию Грина:  9. Решить задачу Коши для ДУ в частных производных 1-го порядка  10. Найти стационарные кривые функционала |  |

**Экзаменационный билет** состоит из двух вопросов и задачи, например

1. Теорема об общем решении неоднородной линейной системы ОДУ. Матрицант. Теорема о частном решении неоднородной линейной системы ОДУ (метод вариации постоянных).

2. Теорема о дифференцируемости по параметру решения задачи Коши для ОДУ 1-го порядка, разрешенного относительно производной. Метод малого параметра.

1. Построить функцию Грина: 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Теория вероятностей и математическая статистика**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Контрольная работа № 1**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Несколько раз бросается игральная кость. Какое событие  более вероятно: {сумма выпавших очков четна} или {сумма  выпавших очков нечетна}?  2. Двое условились о встрече между 10 и 11 часами утра, причем  договорились ждать друг друга не более 10 минут. Считая, что  момент прихода на встречу выбирается каждым «наудачу» в  пределах указанного часа, найти вероятность того, что встреча  состоится.  3. Имеются три урны с белыми и черными шарами, причем отношение числа белых шаров к числу черных равно p1,p2,p3 для 1-й, 2-й,  3-й урн соответственно. Наудачу (с вероятностью 1/3) выбирается  урна и из нее шар. Какова вероятность того, что он белый?  4. Случайная величина X имеет функцию распределения F(x). Найти  функцию распределения случайной величины 0.5(X+|X|). | 1. Сорок участников турнира разбиваются на четыре равные группы.  Найти вероятность того, что четыре сильнейших участника окажутся в разных группах.  2. На отрезок наудачу бросают три точки, одну за другой. Какова  вероятность того, что третья по счету точка упадет между двумя  первыми?  3. Два стрелка стреляют по мишени. Один из них попадает в цель в  среднем в 5 случаях, а второй — в 8 случаях из 10. Перед выстрелом  они бросают правильную монету для определения очередности.  Посторонний наблюдатель знает условия стрельбы, но не знает, кто в  данный момент стреляет. Вот он видит, что стрелок попал в цель.  Какова вероятность того, что стрелял первый стрелок?  4. Пусть X и Y - независимые случайные величины с  непрерывными функциями распределения F(x) и G(x)  соответственно. Найти функцию распределения произведения  XY. |

**Контрольная работа № 2**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Пусть X и Y-- независимые случайные величины,  причем X+Y принимает значения 0, 1, 2 с вероятностями 1/3  каждое. Доказать, что одна из величин X или Y имеет  вырожденное распределение.  2. Найти распределение, которому соответствует характеристическая функция exp(-|t|).  3. Пусть X1,X2,… - Последовательность независимых случайных величин, причем Xn принимает значения -n,0 и n с вероятностями 1/(2n^2),1-1/n^2,1/(2n^2) соответственно. Применим ли к этой последовательности закон больших чисел.  4. Найти приближенное значение для вероятности того, что число успехов в 100 испытаниях Бернулли с вероятностью успеха 0,5 лежит в интервале (35,65). | 1. Доказать,что функция f(z)=|z| не является производящей функцией вероятностного распределения.  2. Найти распределение, которому соответствует характеристическая функция 1/(1+t^2).  3. Пусть X1,X2,… - Последовательность независимых случайных величин, причем Xn принимает значения 2^(-n) и 2^(n) с вероятностями 1/2 Применим ли к этой последовательности закон больших чисел.  4. . Найти приближенное значение для вероятности того, что число успехов в 100 испытаниях Бернулли с вероятностью успеха 0,5 лежит в интервале (47,53). |

**Контрольная работа № 3**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Пусть  независимы и имеют нормальное распределение  Доказать, что  эффективная оценка функции  2. Пусть  независимы и имеют равномерное распределение на отрезке  Найти оценку методом моментов для  и  по первым двум моментам.  3. Пусть  независимы и    Найти одномерную достаточную статистику.  4. Пусть  независимы и распределены с плотностью  Найти оценку максимального правдоподобия для | 1. Пусть  независимы и имеют гамма-распределение  Доказать, что  является эффективной оценкой  2. Пусть  независимы и имеют гамма-распределение Найти оценку методом моментов для  и  по первым двум моментам.  3. Пусть  независимы и    Найти двумерную достаточную статистику.  4. Пусть  независимы и имеют нормальное распределение Найти оценку максимального правдоподобия для |

**Контрольная работа № 4**

|  |  |
| --- | --- |
| **Вариант 1** | **Вариант 2** |
| 1. Пусть  независимы и имеют равномерное распределение на отрезке  Построить кратчайший доверительный интервал для  с коэффициентом доверия  основанный на центральной статистике  2. Пусть  независимы и имеют распределение Пуассона  Построить центральный доверительный интервал с коэффициентом доверия  используя точечную оценку  3. Пусть  независимы и имеют плотность распределения    Построить наиболее мощный критерий размера  для проверки гипотезы  при альтернативе  Найти мощность критерия. | 1. Пусть  независимы и имеют нормальное распределение  Построить кратчайший доверительный интервал для  с коэффициентом доверия  основанный на центральной статистике  2. Пусть  независимы и имеют нормальное распределение с параметрами m и 1. Построить центральный доверительный интервал для m с коэффициентом доверия  используя точечную оценку  3. Пусть  независимы и имеют биномиальное распределение  Построить равномерно наиболее мощный критерий размера  для проверки гипотезы  при альтернативе  Найти функцию мощности. |

**Вопросы для индивидуального собеседования на устном экзамене.**

1. Вероятностное пространство. Операции над событиями. Свойства вероятности. Условная вероятность. Независимость событий. Критерий независимости. Формула полной вероятности. Формулы Байеса.
2. Прямое произведение вероятностных пространств. Независимые испытания Бернулли.
3. Случайная величина. Порожденное и индуцированное вероятностные пространства.
4. Функция распределения, ее свойства. Дискретные, сингулярные и абсолютно непрерывные функции распределения и случайные величины. Плотность распределения.
5. Теорема Лебега о разложении функции распределения.
6. Моменты случайных величин. Их свойства.
7. Совокупности случайных величин. Совместная функция распределения. Независимость случайных величин. Критерии независимости.
8. Виды сходимости последовательностей случайных величин.
9. Неравенства Маркова и Чебышева. Закон больших чисел в форме Чебышева.
10. Лемма Бореля-Кантелли. Неравенство Колмогорова.
11. Усиленный закон больших чисел в форме Колмогорова. Усиленный закон больших чисел для независимых одинаково распределенных случайных величин.
12. Характеристические функции и их свойства.
13. Закон больших чисел в форме Хинчина. Центральная предельная теорема.
14. Условное математическое ожидание.
15. Статистическая структура. Выборка. Статистика. Порядковые статистики. Вариационный ряд. Выборочные моменты и выборочная функция распределения. Их свойства.
16. Точечная оценка. Несмещенность, состоятельность, оптимальность. Теорема о единственности оптимальной оценки.
17. Функция правдоподобия. Достаточные статистики, полные статистики. Теорема факторизации.
18. Неравенство Рао-Крамера. Эффективные оценки.
19. Теорема Рао-Блекуэлла-Колмогорова. Оптимальность оценок являющихся функцией полной достаточной статистики.
20. Метод моментов. Свойства оценок, полученных методом моментов.
21. Метод максимального правдоподобия. Свойства оценок максимального правдоподобия.
22. Доверительные интервалы. Методы центральной статистики и использования точечной оценки.
23. Проверка гипотез. Лемма Неймана-Пирсона.
24. Критерии согласия Колмогорова и -квадрат.

**Экзаменационный билет** состоит из двух вопросов и задачи, например

1. Случайная величина. Порожденное и индуцированное вероятностные пространства. Функция распределения, ее свойства.

2. Функция правдоподобия. Достаточные статистики, полные статистики. Теорема факторизации.

3. Пусть  имеет биномиальное распределение  Найти оценку максимального правдоподобия для 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Введение в численные методы**

**Вопросы к экзамену**

1. Прямые методы решения СЛАУ. Метод Гаусса.
2. Трехдиагональные системы линейных алгебраических уравнений. Метод прогонки.
3. Обусловленность системы линейных алгебраических уравнений. Число обусловленности.
4. Одношаговые итерационные методы решения системы линейных алгебраических уравнений. Достаточные условия сходимости.
5. Метод простой итерации.
6. Метод Зейделя.
7. Метод верхней релаксации.
8. Интерполирование полиномами. Интерполяционные формулы Лагранжа и Ньютона.
9. Погрешность интерполяционного полинома.
10. Интерполирование с кратными узлами. Полиномы Эрмита
11. Интерполирование сплайнами.
12. Квадратурные формулы прямоугольников и трапеций.
13. Квадратурные формулы Симпсона.
14. Квадратурные формулы Гаусса.
15. Сеточные функции. Разностная аппроксимация первой и второй производной.
16. Метод Эйлера.
17. Метод Рунге-Кута.
18. Метод Адамса.
19. Сетки, сеточные функции. Аппроксимация первой и второй производной.
20. Разностная аппроксимация краевой задачи для линейного дифференциального уравнения второго порядка.

**Типовые задачи для экзамена**

|  |
| --- |
| 1. Рассмотреть задачу Коши    Сделать для нее один шаг по методу Рунге-Кутта с при .  Сравнить результат с точным решением.  2. Найти число обусловленности матрицы . |

**Экзаменационный билет** состоит из одного вопроса и задачи, например

1. Одношаговые итерационные методы решения системы линейных алгебраических уравнений. Достаточные условия сходимости**.**

2. Дана система уравнений

2x + y = 4

x + y = 3:

Записать метод простой итерации и указать, при каких значениях итерационного параметра он сходится.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Алгоритмы и алгоритмические языки**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Коллоквиум**

**1**. Пусть определены следующие переменные:

**int** x = 2, y = 4, z = 6;

**short** n = 0x7FFF; **unsigned** **short** m = 0x7FFFU;

**struct** S { **struct** P { **int** x, y; } p; **struct** S \*n; } s = { { -10, -20 },   
&s };

**unsigned** **int** a[10] = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };

**unsigned** **int** \*p, \*\*pp;

Для каждого из  **8** отдельных независимых выражений указать ***его значение*** и ***побочные эффекты*** (если они есть) либо “ошибка”, если выражение ошибочно.

1) x | y ^ z

2) n + 4

3) m <<= 1

4) s.p.x = x + s.n->p.y

5) \*(a + 4 \* x)

6) p = a + 1, \*\*(pp = &p)

7) \*pp = &a[2], \*(\*pp + y + a[2])

8) p = a + a[3], s.p.y = ++p - a

***УКАЗАНИЕ.***

*В задачах 2-4 определите все необходимые переменные и типы.*

*Считайте, что вызовы функций выделения памяти всегда заканчиваются успешно.*

**2.** Напишите функцию, принимающую единственный параметр – массив строк (указателей типа **char \***). Последний элемент массива содержит нулевой указатель. Функция должна возвратить указатель на размещенный в динамической памяти массив структур, в которых первое поле есть указатель на копию одной из входных строк, а второе поле – количество раз, которое данная строка встречалась во входном массиве (с учетом регистра символов). Одинаковые строки (с учетом регистра символов) должны содержаться в выходном массиве только в одном экземпляре, поле с указателем может указывать на любую из них. Выходной массив должен быть отсортирован сначала по убыванию по второму полю, а затем в лексикографическом порядке по первому полю.

**3.** Напишите функцию, которая удаляет из односвязного списка все элементы с нечетными номерами. Порядок следования всех остальных элементов должен остаться неизменным. Память, занятая удаляемыми элементами списка, должна быть освобождена с помощью вызова функции free. Считайте, что заголовочного элемента в списке нет и элементы списка нумеруются с 1. Ваша функция должна соответствовать прототипу

**void** remove\_odd (struct list \*\*);

где struct list описывает элемент списка с целочисленным ключом.

**4.** Напишите функцию, принимающую три параметра – массив целых чисел, его длину и указатель на целое число. Элементы массива образуют (возможно пустое) двоичное дерево (для элемента a[i] его детьми в двоичном дереве являются элементы a[2\*i+1] и a[2\*i+2], элемент a[0] – корень дерева). Функция должна возвратить указатель на построенное в динамической памяти (в виде рекурсивной ссылочной структуры) двоичное дерево, в котором для каждого узла детьми являются те же числа, что и во входном массиве (или нулевой указатель для пустого дерева). По указателю, который является третьим параметром, необходимо записать 1, если построенное дерево является двоичным деревом поиска, и 0 в противном случае.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

## 

## Вариант письменного экзамена по курсу

**1**. Составьте диаграмму машины Тьюринга (МТ), которая в непустом входном слове над алфавитом А3={*0*, *1*, *2*} меняет местами символы в паре подряд стоящих символов (например, входное слово 02121 преобразуется в слово 20211). Пары символов отсчитываются от левого края (первого символа) слова. Лента МТ бесконечна только справа. В начальном состоянии головка МТ обозревает пустую ячейку сразу после входного слова, в конечном состоянии – пустую ячейку сразу после выходного слова. Можно предполагать, что слева от входного слова на ленте есть одна пустая ячейка. При построении можно использовать элементарные машины *l, r, L, R, символ* (сдвиг головки на одну ячейку влево/вправо, сдвиг головки на одно слово влево/вправо, запись символа в ячейку, соответственно).

**2.** Дано описание:

**struct** list {**int** key; **struct** list \*next;};

Напишите функцию

**void** move\_elts\_keys (**struct** list \*\*src, **struct** list \*\*dst),

которая переносит из односвязного списка src в односвязный список dst все элементы с чётными ключами, удаляя их при этом из исходного списка. Перенесенные элементы должны добавляться в начало списка dst в произвольном порядке. Относительный порядок изначальных элементов обоих списков должен остаться неизменным. Считайте, что заголовочного элемента в обоих списках нет, списки могут быть пустыми.

**3.** Пусть **int** x = 42; **int** y = -105;

Для каждого из  **6** отдельных независимых выражений указать ***его значение*** и ***побочные эффекты*** (если они есть) либо “ошибка”, если выражение ошибочно.

1) y %= x / 5

2) x ^ ~y & 65

3) y <<= (--x || --y)

4) (y = 1) && (y = 2)

5) x++ + --y

6) x += y, y = x - y, x -= y

**4.** Что будет напечатано при выполнении программы:

**#include** <stdio.h>

**int** x = 2, y = 1;

**int** a[3] = { 1, 2 };

**int** f (**int** x, **int** \*\*p) {

**static** **int** c = 2;

**if** (c--)

\*\*p += x;

**else**

++x;

**return** (\*p)++ - a - x;

}

**int** main (**void**) {

**int** x = y, c = 0; **int** \*p;

p = &a[x];

**for** (**int** y = 2; y >= c; --y, ++x)

c = f (y, &p);

printf ("%d %d %d %d %d\n", a[0], a[1], a[2], x, y);

**return** 0;

}

**5.** Перепишите приведенный фрагмент программы с использованием единственного оператора **switch** (дополнительные условные операторы использовать запрещено):

**if** (i == -1)

j = i;

**else** **if** (3 >= i && i >= 1)

j = 10 - i;

**else**

abort ();

**6.** С использованием единственного цикла **for** напишите фрагмент программы, который для массива **int** a[N] формирует в переменной pairs количество пар соседних элементов, имеющих разную четность. Другие операторы циклов и операторы перехода использовать запрещено.

**7.** Напишите функцию

**int** scalar (**void**),

которая считывает со стандартного потока ввода непустую последовательность целых чисел, заканчивающуюся нулем (нуль не входит в последовательность, количество ненулевых элементов четно, элементы нумеруются с 1). Функция возвращает значение скалярного произведения двух векторов, составленных из элементов последовательности: первый вектор – из элементов с нечетными номерами, второй вектор – из элементов с четными номерами. В решении можно использовать не более одного оператора цикла. Запрещается использовать операторы перехода.

**8.** Напишите функцию

**void** erase (**char** \*s, **const char** \*w),

на вход которой дается строка ***s*** и слово ***w***. Слово – непустая строка из латинских букв. Строка ***s*** представляет собой последовательность слов, разделенных одним пробелом. Перед первым словом и после последнего пробелов нет. Функция должна удалить из строки ***s*** все слова, в которые слово ***w*** входит как подстрока. Относительный порядок всех остальных слов должен остаться неизменным. Измененная таким образом строка ***s*** должна удовлетворять всем тем же ограничениям, что и входная. Память под все временные массивы должна выделяться динамически и освобождаться в конце работы функции.

**9.** Дано описание:

**struct** avltree {**int** x; **struct** avltree \*left, \*right;};  
Напишите функцию

**struct** avltree \*build (**int** h),

которая строит АВЛ-дерево заданной высоты ***h***, содержащее минимальное количество узлов ***n*** из всех АВЛ-деревьев с этой высотой. Значения ключей в элементах АВЛ-дерева должны быть от 1 до ***n***.

**10.** Нарисуйте дерево цифрового поиска для алфавита **{*В*, *М*, *И*, *К*},** которое содержит следующие ключи: ***КИМ, МИКИ, ВИМ, ИМ, ИК, ВК, МИМ, ВМК, В, МИ, ВМИК, МИМИ.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Архитектура ЭВМ и язык ассемблера**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Типовой вариант коллоквиума**

**1.** Секция .data была размещена в памяти, начиная с базового адреса 0x0804a024. Выпишите значение регистра EAX и флагов после выполнения указанных команд.

|  |  |  |
| --- | --- | --- |
| section .data  a dd b  b dd a | section .text  mov eax, 0xFF00  add ah, 0xCD ; (1)  movsx eax, word [a+3] ; (2) | Ответ:  (1) EAX = \_\_\_\_\_\_\_\_\_\_\_\_\_\_  CF = \_\_, OF = \_\_, ZF = \_\_, SF = \_\_  (2) EAX = \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**2.** По данному фрагменту ассемблерного кода восстановите объявления глобальных Си-переменных и выражение-оператор.

|  |  |  |
| --- | --- | --- |
| mov eax, dword [a]  cdq  idiv dword [t]  mov dword [a], edx | mov ecx, dword [i]  shr dword [z], cl | moval, byte [s]  cmp byte [q], al  setl al  mov byte [m], al |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**3.** Для данного фрагмента ассемблерного кода восстановите пропуски в соответствующем коде на языке Си.

|  |  |
| --- | --- |
| mov edx, dword [a]  mov eax, dword [c]  lea eax, [eax+edx\*8]  mov edx, dword [p1]  add eax, eax  add eax, dword [b]  movsx edx, word [edx]  mov dword [gg+eax\*4], edx | // объявление переменных  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ gg[5][\_\_][\_\_];  inta, b, c;  // выражение-оператор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; |

**4.** Для данного фрагмента ассемблерного кода восстановите пропуски в соответствующем коде на языке Си.

|  |  |
| --- | --- |
| xor eax, eax  xor ebx, ebx  .L3:  mov ecx, dword[b+eax\*4]  mov edx, dword [a+eax\*4]  sub edx, ecx  cmp dword [a+eax\*4], ecx  cmovl edx, ebx  mov dword[c+eax\*4], edx  inc eax  cmp eax, 1024  jne .L3 | #define N 1024  static int a[N], b[N], c[N];  for (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;  } |

**5.** Для данного фрагмента Си-кода приведите соответствующий ассемблерный код, причем значение вычисляемого булевского выражения после выполнения оператора должно оказаться в регистре EAX.

|  |
| --- |
| static int \*p1, \*p2;  static int diff;  p1 && p2 && (diff = p1 – p2++); |

**Типовой вариант коллоквиума №2**

**1.** Реализуйте заданную функцию на языке ассемблера для платформы IA-32/Linux.   
**Дополнительные требования**: (1) при вызове функций сохраняйте выравненность стека по 16 байтной границе, (2) код должен отражать особенности компиляции с ключом -fomit-frame-pointer.

typedef struct link link;

struct link {

long data;

link \*next;

};

typedef \_\_attribute\_\_((fastcall)) \_Bool (\*comparator)(long, long);

link\* first(link\* p, comparator fp, long data) {

if ((0 == p) || fp(p->data, data)) {

return p;

} else {

return first(p->next, fp, data);

}

}

**2.** Для приведенного Си-кода компилятор построил следующий ассемблерный код (платформа IA-32/Linux).

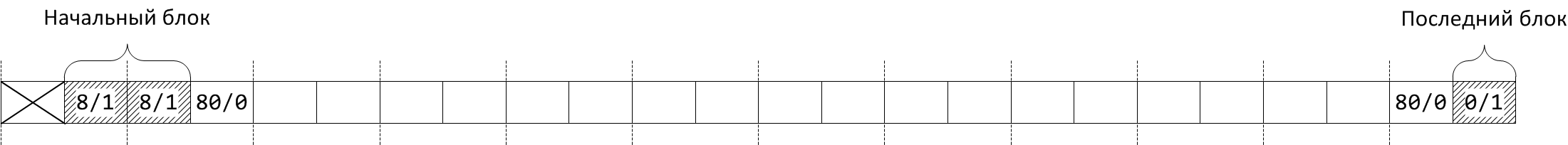
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| #include <stdio.h>  typedef unsigned int uint32\_t;  typedef uint32\_t Elf32\_Word;  typedef uint32\_t Elf32\_Addr;  typedef struct {  Elf32\_Addr r\_offset;  Elf32\_Word r\_info;  } Elf32\_Rel;  unsigned rel\_type(FILE\* fd) {  Elf32\_Rel rel;  fread(&rel, sizeof(rel), 1, fd);  return (rel.r\_info) & 0xff;  }  порядок байт в dword  направление роста адресов | | | | | | rel\_type:  push ebp  mov ebp, esp  sub esp, 24  mov eax, dword [gs:20]  mov dword [ebp-12], eax  xor eax, eax  lea eax, [ebp-20]  push dword [ebp+8]  push dword 1  push dword 8  push eax  call fread  mov edx, dword [ebp-12]  xor edx, dword[gs:20]  movzx eax, byte [ebp-16]  je .L2  call \_\_stack\_chk\_fail  .L2:  leave  ret |
| 0 | 1 | 2 | 3  Нарисуйте устройство фрейма функцииrel\_type:  **(1)** где были размещены сохраненные регистры,  **(2)** автоматические локальные переменные,  **(3)** пространство аргументов вызываемых функций,  **(4)** «канарейку» (если использовалась),  **(5)** неиспользуемые элементы стека. |
| адрес возврата | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |

**3.** Используется 8-ти битный формат, удовлетворяющий требованиям стандарта IEEE 754: знаковый бит, 3 бита – порядок, 4 бита - мантисса. Приведите битовое представление следующих чисел

|  |  |  |
| --- | --- | --- |
| Наименьшее денормализованное | Месяц Вашего рождения | Следующий за ним месяц |
|  |  |  |

Для месяца рождения – укажите его номер (от 1 до 12) в десятичной кодировке: \_\_\_\_\_

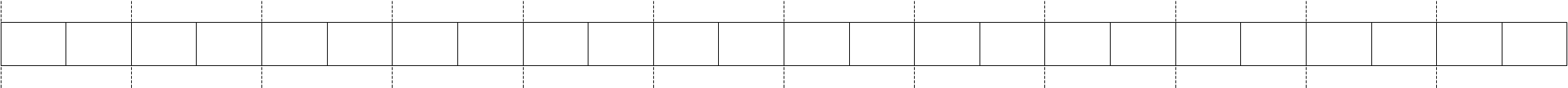
**4.** Модельный менеджер памяти управляет кучей из 24 четырехбайтных машинных слов. Для отслеживания свободных блоков используется неявный список. Начальное состояние кучи показано на рисунке ниже. Свободный блок белого цвета, занятые блоки заштрихованы, неиспользуемая из-за выравнивания память перечеркнута. Начальный и последний блок – служебные, для пользователя они недоступны. Байтовый размер блока хранится в заголовке и граничном теге. В выделенных блоках, за исключением начального блока, граничный тег не используется. Предоставляемая пользователю память выравнивается по 8-ми байтной границе (на рисунке обозначено засечками). Поиск свободного блока начинается с текущей позиции в списке, выбирается первый подходящий. При расщеплении используется первая часть блока, вторая часть – текущая позиция. Слияние проводится незамедлительно.



После выполнения шести обращений к менеджеру динамической памяти  
**А)** запишите значения тегов и **Б)** рассчитайте пиковое использование памяти U6:

U6 = \_\_\_

p1 = malloc(14); p2 = malloc(22); p3 = malloc(8); free(p1); p4 = malloc(4); free(p2);



**5.** Си-программа состоит из двух модулей: t5-1.c и t5-2.c. В результате компоновки: у модуля t5-1 секция .rodata оказалась по адресу 0x080484d0, секция .data по адресу 0x0804a020; у модуля t5-2 секция .textразместилась по адресу 0x0804842b. Символ strcpy@plt исполняемого файла помещен на адрес 0x80484b0. Определите значения указанных ссылок в исполняемом файле после перебазирования.

// t5-1.c

extern void transfer();

char \*rmi[3] = { "MC Solaar",

"3ème Oeil",

"Carré Rouge"};

char \*rsa[3];

int main(void) {

transfer();

}

t5-1.o: file format elf32-i386

RELOCATION RECORDS FOR [.data]:

OFFSET TYPE VALUE

00000000 R\_386\_32 .rodata

00000004 R\_386\_32 .rodata

00000008 R\_386\_32 .rodata

Contents of section .data:

0000 00000000 **0a000000** 15000000 ............

// t5-2.c

#include <string.h>

extern char\* rmi[];

extern char\* rsa[];

void transfer(int pauvre) {

strcpy(rsa[pauvre], rmi[pauvre]);

}

t5-2.o: file format elf32-i386

Disassembly of section .text:

00000000 <transfer>:

0: 55 push ebp

1: 89 e5 mov ebp,esp

3: 83 ec 10 sub esp,0x10

6: 8b 45 08 mov eax,DWORD PTR [ebp+0x8]

9: ff 34 85 **00 00 00 00** push DWORD PTR [eax\*4+0x0]

**c: R\_386\_32 rmi**

10: ff 34 85 00 00 00 00 push DWORD PTR [eax\*4+0x0]

13: R\_386\_32 rsa

17: e8 **fc ff ff ff** call 18 <transfer+0x18>

**18: R\_386\_PC32 strcpy**

1c: 83 c4 10 add esp,0x10

1f: c9 leave

20: c3 ret

|  |  |  |  |
| --- | --- | --- | --- |
| Модуль | Секция | Смещение ссылки | **Новое значение** |
| t5-1 | .data | 0x4 |  |
| t5-2 | .text | 0xс |  |
| t5-2 | .text | 0x18 |  |

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

## Программа экзамена по курсу

Шестнадцатеричная система счисления. Знаковые числа, дополнительный код. Арифметические операции. Флаги: CF, SF, OF и ZF.

Архитектура IA-32: основные регистры, форматы команд. Порядок байт в памяти. Машинные типы данных: байт, слово, двойное слово, четверное слово. Аппаратный стек.

Инструкции в IA-32/nasm: пересылки, в том числе условные, арифметические, логические, битовые, сдвиги и вращения, передачи управления.

Отображение операторов разыменование указателя и взятия адреса из языка Си в язык ассемблера. Размещение различных типов переменных языка Си в памяти компьютера. Работа с указателями. Адресная арифметика.

Массивы: одномерные, многомерные, многоуровневые. Расположение в памяти, способы работы с отдельными элементами. Преобразование индексных выражений в адресную арифметику.

Реализация классов памяти языка Си на уровне языка ассемблера, размещение переменных: глобальных, статических, автоматических. Приведение типов данных. Работа со знаковыми и беззнаковыми числами.

Передача управления. Условные и безусловные переходы. Связь регистра EFLAGS и мнемонических кодов условий. Реализация управляющих операторов языка Си на языке ассемблера.

Организация работы со структурами и объединениями языка Си на уровне языка ассемблера. Доступ к полям. Выравнивание данных в ОС Windows и Linux. Размер агрегатных типов данных.

Соглашение о вызове функций cdecl. Распределение памяти во фрейме функции. Возвращаемое значение в соглашении cdecl. Функции с переменным числом параметров.

Соглашение вызова stdcall, fastcall, реализация вызова функции без использования указателя фрейма, гибридное соглашение вызова.

Выравнивание фреймов в стеке. Организация вызова функций стандартной библиотеки языка Си из ассемблерного кода.

Управление динамической памятью. Пропускная способность и эффективность расходования. Внутренняя и внешняя фрагментация. Управление свободными блоками: неявный список.

Представление чисел с плавающей точкой. Стандарт IEEE 754. Свойства чисел с плавающей точкой. Операции над числами с плавающей точкой. Округление чисел.

Сопроцессор FPU x87. Аппаратный стек регистров. Организация работы с числами с плавающей точкой в языке Си: пересылка данных, основные арифметические операции.

Устройство современного компьютера, запоминающие устройства: организация, емкость. Соотношение временных характеристик доступа на разных уровнях иерархической памяти компьютера.

Кэш-память процессора, способы ее организации: кэш прямого отображения, N-канальный множественно-ассоциативный кэш, полностью ассоциативный кэш.

Организация виртуальной памяти, страничная трансляция адреса. Буфер быстрого преобразования адреса (TLB).

Многомодульные программы. Глобальные, локальные, внешние имена. Сильные и слабые символы, COMMON-символы.

Объектные файлы формата ELF. Статическая компоновка программы: разрешение символов, перемещение кода, модификация символов и ссылок.

Загрузка исполняемого файла в память. Динамическая компоновка, разделяемые библиотеки. Позиционно независимый код. Глобальная таблица смещений. Ленивое связывание в динамической компоновке. Динамическая загрузка.

## 

## Вариант письменного экзамена по курсу

**1.** После выполнения последовательности команд в регистре-аккумуляторе было получено некоторое значение. Выпишите восстановленное содержимое ячеек памяти согласно заданной директиве. Если часть значения не может быть восстановлена, запишите вместо соответствующих шестнадцатеричных цифр символы ‘**?**’.

|  |  |  |
| --- | --- | --- |
| **a dd \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  mov eax, dword [a]  shl eax, 8  ; EAX = 0xae191700 | **b dw \_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_**  mov eax, dword [b]  ror eax, 8  or ax, 0x0f0f  ; EAX = 0xf4c01fcf | **c dw \_\_\_\_\_\_\_\_**  movsx ax, byte [c + 1]  cmp ah, byte [c]  setbe ah  ; AX = 0x01c3 |

**2.** Реализуйте приведенный Си-код, включая объявления переменных.

static int i;

static short \*a[10];

...

a[i++][5] \*= -1;

**3.** Для данного фрагмента ассемблерного кода восстановите соответствующий код на языке Си.

|  |  |  |
| --- | --- | --- |
| section .rodata  .L6:  dd .L1  dd .L2  dd .L3  dd .L4  dd .L1  dd .L4 | mov ecx, dword [x]  cmp ecx, 5  ja .L1  jmp [.L6+ecx\*4]  .L1:  mov eax, 107  add eax, dword [a + 12]  jmp .L5  .L2:  mov eax, 99  .L3:  add eax, dword [a]  jmp .L5  .L4:  mov eax, dword [a + 4]  add eax, dword [a + 8]  .L5: | static int a[N];  static int x;  inty; // размещенавEAX  switch (x) {  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  } |

**4.** Перечислите Си-выражения, имеющие истинное значение при любых значениях переменных заданных типов. Предполагается, что переменные типа float и double имеют значение отличное от NAN.

|  |  |  |  |
| --- | --- | --- | --- |
| signed char c;  short s;  int i;  float f;  double d; | А) i == -(-i);  Б) f\*(f+s) == f\*f+f\*s;  В) i < s ⇒ -i > -s  Г) 32/4.0 == 32/4 | Д) i == (int)(float)i;  Е) d > f ⇒ -f > -d;  Ж) -f \* f < 0.0  З) (c+s)+i == c+(s+i); | **Ответ:** |

**5.** Си-программа, использующая тип данных s, была скомпилирована на платформе IA-32/Linux.

|  |  |  |
| --- | --- | --- |
| typedef struct t\_s st;  struct t\_s {  char b;  union {  st \*n;  int val[2];  } u;  double d;  int r;  };  static st s, \*p;  static int x; | Приведите соответствующий ассемблерный код (В, Г) либо выпишите числовое значение(А, Б):  А)sizeof(s)  Б) Смещениеполяs.u.val[1]  В) s.r = s.u.val[0] + s.u.val[1];  Г)  x = p->u.n->r; | А)  Б)  В)  Г) |

**6.** Компилятор построил для тела Си-функции f следующий ассемблерный код. Исходя из этого кода и того, что было использовано соглашение cdecl, восстановите заголовок функции: типы параметров, их порядок, тип возвращаемого значения.

|  |  |
| --- | --- |
| mov ecx, dword [ebp+16]  mov eax, dword [ebp+8]  mov ebx, dword [ebp+12]  movsx esi, byte [eax]  mov eax, dword [ecx]  cdq  idiv esi  mov edx, dword [ebp+20]  sar dx, 2  movsx edx, dx  mov dword [ecx], eax  add dword [ebx], edx  mov eax, ecx | \_\_\_\_\_\_\_\_\_\_\_\_ f(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ) {  \*z = \*z / \*w;  \*x = \*x + (v >> 2);  return z;  } |

**7.** Реализуйте на языке ассемблера заданную рекурсивную функцию, используя соглашение cdecl. Перед вызовом функции f стек уже выровнен должным образом. **Дополнительные требования**: 1) сохраняйте выравненность стека,2) код должен отражать особенности компиляции с ключом –fomit-frame-pointer.

typedef \_\_attribute\_\_((fastcall)) void (\*fp)(int);

typedef struct link link;

struct link {

fp \*payload;

link \*next;

};

void f(int v, link\* p) {

if (p) {

f(v, p->next);

(\*p->payload)(v);

}

}

|  |  |
| --- | --- |
| Параметр | **Значение** |
| VPN |  |
| Попадание в TLB? (да/нет) |  |
| Страница присутствует? (да/нет) |  |
| PPN |  |
| Номер набора в кэш данных |  |
| Бит валидности |  |
| Попадание в кэш? (да/нет) |  |

|  |  |  |
| --- | --- | --- |
| Кэш данных | | |
| Набор | tag | v |
| 0 | 3 | 1 |
| 1 | 2 | 0 |
| 2 | 1 | 0 |
| 3 | 0 | 1 |
| 4 | 1 | 1 |
| 5 | 1 | 0 |
| 6 | 0 | 1 |
| 7 | 0 | 1 |

**8.** Память модельного компьютера состоит из 512 адресуемых ячеек размером 1 байт. Выполняется страничная трансляция линейных адресов при обращении к физической памяти. Размер страницы – 64 байта. Транслированные адреса сохраняются в TLB, организованный как полностью ассоциативный кэш. Обращение к физической памяти предваряется проверкой кэша данных, имеющего следующее устройство: прямое отображение, 16 байт в строке, 8 наборов. Дано: состояние TLB, фрагмент таблицы страниц, кэш данных. Бит **p** в TLB и таблице страниц показывает присутствие страницы. Заполните столбец «**Значение»** в таблице, определив, как именно будет происходить чтение байта по виртуальному линейному адресу **0x1ce**.

VPN – Номерстраницы виртуальной памяти

PPN – Номерстраницы физической памяти

|  |  |  |
| --- | --- | --- |
| Фрагмент таблицы страниц | | |
| VPN | PPN | p |
| 5 | 4 | 1 |
| 6 | 3 | 1 |
| 7 | 3 | 1 |
| 8 | ‑ | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние TLB | | | | | | | |
| tag | v | PPN | p | tag | v | PPN | p |
| 0 | 0 | ‑ | 0 | 6 | 1 | 3 | 1 |
| 2 | 1 | 1 | 1 | 3 | 1 | ‑ | 0 |

**9.** Си-программа состоит из двух модулей: t9-m.c и t9-e.c. Заполните таблицы, приведенные ниже. Для каждого заданного в таблице имени переменной или функции укажите (да/нет), содержится ли соответствующая запись в таблице символов .symtab объектного файла, Если да, укажите тип связывания символа (local, global, extern), в каком модуле (t9-m.o, t9-e.o, еслиCOMMON-символ присутствует в двух модулях – указывайте оба модуля) и какой именно секции (.text, .bss, .data, COMMON-символ) символ определен. Если ответ дать невозможно – ставьте прочерк.

|  |  |
| --- | --- |
| t9-m.c | t9-e.c |
| #include <sys/stat.h>  char\* fileName;  extern  void printLastAccess(struct stat \*s);  int main(int argc, char \*argv[]) {  static struct stat sb = {0};  int res;  res = stat(fileName = argv[1], &sb);  printLastAccess(&sb);  return res;  } | #include <sys/stat.h>  #include <stdio.h>  #include <time.h>  extern char\* fileName;  FILE \*fd;  void printLastAccess(struct stat \*s) {  char \*formatedTime = ctime(&s->st\_atime);  printf("Last file access: %s", formatedTime);  fd = fopen("/tmp/leak.log", **"**a**"**);  fprintf(fd, "%s\n", fileName);  fclose(fd);  } |

Файл t9-m.c

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя функции/переменной | Присутствует ли в .symtab файла t9-m.o | Тип связывания символа | Модуль, в котором символ определен | Секция, в которой символ определен |
| printLastAccess |  |  |  |  |
| main |  |  |  |  |

Файл t9-e.c

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя функции/переменной | Присутствует ли в .symtab файла t9-e.o | Тип связывания символа | Модуль, в котором символ определен | Секция, в которой символ определен |
| formatedTime |  |  |  |  |
| fd |  |  |  |  |

**10.** Си-программа состоит из двух модулей: t10-m.c и t10-p.c, использующих общий заголовочный файл t10.h. В результате компоновки секция .text из модуля t10-m.c была размещена по адресу 0x0804844b, а ссылки в этой секции получили новые значения d9 fe ff ff, 39 00 00 00, 30 a0 04 08 (приведены в порядке их следования в ассемблерном листинге). Заполните в таблице последний столбец, вычислив адреса размещения указанных символов и секций.

**Модуль t10-m.c**

#include <stdlib.h>

#include "t10.h"

unsigned salt = 0xfaceb00c;

char\* tbl[3] = {"http://cmc.msu.ru",

"https://www.mipt.ru",

"https://cs.hse.ru"};

int main(int argc, char\* argv[]) {

print(atoi(argv[1]));

return 0;

}

int randIndex(int i) {

return (i ^ salt) % 3;

}

t10-m.o: file format elf32-i386

Contents of section .data:

0000 00000000 12000000 26000000 0cb0cefa ........&.......

|  |  |  |
| --- | --- | --- |
| **Модуль** | **Секция / символ** | **Адрес размещения** |
| **‑** | **atoi@plt** |  |
| **t10-p.o** | **.text** |  |
| **t10-m.o** | **.data** |  |

extern char\* tbl[3];

extern unsigned salt;

extern void print(int i);

extern int randIndex(int i);

**Заголовочный файл t10.h**

**Модуль t10-p.c**

t10-m.o: file format elf32-i386

Disassembly of section .text:

00000000 <main>:

0: 8d 4c 24 04 lea ecx,[esp+0x4]

4: 83 e4 f0 and esp,0xfffffff0

7: ff 71 fc push DWORD PTR [ecx-0x4]

a: 55 push ebp

b: 89 e5 mov ebp,esp

d: 51 push ecx

e: 83 ec 10 sub esp,0x10

11: 8b 41 04 mov eax,DWORD PTR [ecx+0x4]

14: ff 70 04 push DWORD PTR [eax+0x4]

17: e8 **fc ff ff ff** call 18 <main+0x18>

**18: R\_386\_PC32 atoi**

1c: 89 04 24 mov DWORD PTR [esp],eax

1f: e8 **fc ff ff ff** call 20 <main+0x20>

**20: R\_386\_PC32 print**

24: 8b 4d fc mov ecx,DWORD PTR [ebp-0x4]

27: 31 c0 xor eax,eax

29: c9 leave

2a: 8d 61 fc lea esp,[ecx-0x4]

2d: c3 ret

0000002e <randIndex>:

2e: 55 push ebp

2f: b9 03 00 00 00 mov ecx,0x3

34: 31 d2 xor edx,edx

36: 89 e5 mov ebp,esp

38: 8b 45 08 mov eax,DWORD PTR [ebp+0x8]

3b: 33 05 **00 00 00 00** xor eax,DWORD PTR ds:0x0

**3d: R\_386\_32 salt**

41: 5d pop ebp

42: f7 f1 div ecx

44: 89 d0 mov eax,edx

46: c3 ret

#include <stdio.h>

#include "t10.h”

... // пропущенныйкод

void print(int i ) {

puts(tbl[randIndex(i)]);

}

t10-p.o: file format elf32-i386

Disassembly of section .text:

...

00000016 <print>:

16: 55 push ebp

17: 89 e5 mov ebp,esp

19: 83 ec 14 sub esp,0x14

1c: ff 75 08 push DWORD PTR [ebp+0x8]

1f: e8 **fc ff ff ff** call 20 <print+0xa>

**20: R\_386\_PC32 randIndex**

24: 8b 04 85 **00 00 00 00** mov eax,DWORD PTR [eax\*4+0x0]

**27: R\_386\_32 tbl**

2b: 83 c4 10 add esp,0x10

2e: 89 45 08 mov DWORD PTR [ebp+0x8],eax

31: c9 leave

32: e9 **fc ff ff ff** jmp 33 <print+0x1d>

**33: R\_386\_PC32 puts**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**История**

**Типовые оценочные средства, необходимые для оценки знаний, умений, навыков при проведении текущей аттестации**

Для оценки практической и самостоятельной работы студентов используется рейтинговая система, при которой каждая форма работы оценивается определенным количеством баллов. Сумма баллов определяет место студента в рейтинге.

Наименование оценочных средств предусмотренных рейтинговой системой:

* Контрольная работа
* Выступление с сообщением по теме семинара с презентацией
* Эссе
* Кейс (ситуационное задание)
* Балльная оценка аудиторной работы

**Пример типовых контрольных заданий**

**Контрольная работа**

1. Князь, пытавшийся провести реформу язычества в 980 г.
2. «Русская правда» - это…
3. 1019-1054 гг. в истории России
4. Ярлык на княжение – это…
5. Московский князь, превративший Москву в центр единого Русского государства, при котором страна окончательно была освобождена от ордынской зависимости
6. Независимость русской митрополии от Византии
7. Территориальные приобретения российского государства при Иване IV
8. 1589 год в истории России
9. Как назывался свод законов Русского государства, принятый Земским собором в 1649 году и действовавший до 1832 года?
10. Закон о порядке государственной службы в Российской империи, учрежденный Петром I
11. 1721 год в истории России
12. Абсолютизм – это…
13. Кто из императорских особ издал «Манифест о вольности дворянской»?
14. В 1772,1793,1795 годах произошли…
15. 1773-1775 годы в истории России

**Эссе**

**Примерные темы эссе**

1. Основные концепции образования Древнерусского государства
2. «Русская Правда» - первое письменное законодательство России
3. Владимир Креститель и его роль в русской истории
4. Раздробленность на Руси: причины, сущность, последствия
5. Свержение монгольского владычества на Рус.
6. Субъективные факторы в становлении российской государственности XV – XVI веков
7. Деятельность «Избранной рады»
8. Опричнина Ивана Грозного: споры в исторической науке
9. Причины Смутного времени на Руси в начале XVII века
10. Религиозная реформа патриарха Никона
11. Социальные движения в России XVII века
12. Реформы Петра I
13. Внешние факторы в развитии российского государства в XVIIII веке
14. Идеология и практика Просвещенного абсолютизма II половины XVIIII века
15. Государственный переворот 1801 года
16. Роль и значение движения декабристов в освободительно движении России
17. Положение сословий российского общества в первой половине XIX века: основные черты, социальные конфликты
18. Теория официальной народности
19. Политический режим Николая I: идеология и практика
20. Этнополитическое развитие России в XIX веке
21. Историческое значение отмены крепостного права в России
22. «Великие реформы» Александра II
23. Освободительное движение в России II половины ХIХ века
24. Россия во внешнеполитических союзах последней четверти XIX века
25. Значение «Манифеста об усовершенствовании государственного порядка» от 17 октября 1905 г. в истории России
26. Император Николай II как государственный деятель
27. Власть в условиях революции 1917 года в России: взаимоотношения Временного правительства и Совета рабочих и солдатских депутатов
28. Значение индустриализации в СССР
29. Коллаборационизм и партизанское движения в годы Великой Отечественной войны
30. Роль СССР во Второй Мировой войне
31. Особенности десталинизации в СССР
32. Ю.В.Андропов во главе советского государства
33. Концептуальное содержание и практическая реализация теории «развитого социализма»
34. Перестройка системы управления советским государством в конце 1980-х годов
35. Переход к рынку в Российской Федерации: проблемы и методы решения
36. Б.Н. Ельцин как политический лидер и управленец

**Кейс (ситуационное задание)**

**Пример кейса**

Тема: Развитие российского государства в период правления Петра I

Задание: сформулировать собственную аргументированную позицию в полемике о реформах Петра I, для чего:

• провести группировку мнений историков, выделив принципиальные разногласия между ними, как в оценке взаимодействия, так и в аргументации;

• установить, в какой мере взгляды, высказанные в отечественной исторической науке, соответствуют известным на данный момент источникам;

• на основе предложенных источников привести аргументацию в пользу одной из предложенных точек зрения или собственной позиции.

Заявленные суждения:

Суждение 1. Реформы Петра I были необходимы для России и ознаменовали начало новой эпохи.

Суждение 2. Реформы Петра I были не продуманны и принесли больше вреда, чем пользы.

Суждение 3. Реформы Петра I не оказали серьезного влияния на историю страны.

**Типовые оценочные средства, необходимые для оценки знаний, умений, навыков при проведении промежуточной аттестации (устный экзамен).**

**Примерный список вопросов для проведения промежуточной аттестации**

1. Древнерусское государство: особенности образования и развития
2. Русские земли в период феодальной раздробленности
3. Борьба русских княжеств и земель против внешней агрессии в начале XIII в. Золотоордынская зависимость и её влияние на положение и развитие русских земель
4. Причины, предпосылки и особенности образования Московского государства, основные этапы объединения земель вокруг Москвы
5. От Руси к России: территориальное расширение российского государства и борьба за выход к морям в XVI в.
6. Смута в России: причины, основные этапы, проявления, последствия
7. Первые Романовы на российском престоле. Государственные институты и их эволюция в XVII в.
8. Реформы Петра I и их влияние на историческую судьбу России
9. Внешняя политика Петра I и изменения в геополитическом положении России
10. «Эпоха дворцовых переворотов» в России в XVIII в.: система власти и внутренняя политика
11. Внешняя политика российского государства во второй половине XVIII века
12. «Просвещенный абсолютизм» в России: содержание, особенности, противоречия
13. Внутренняя политика Александра I
14. Россия в системе европейских международных отношений в первой половине ХIХ в.
15. Социально-экономическое развитие России в первой половине XIX в.
16. Внешняя политика и территориальные приобретения России во второй половине XIX в.
17. «Великие реформы» 60 - 70-х гг. XIX в.: сущность, последствия
18. Политический консерватизм Александра III. Социально-экономическое развитие России в 1880-х - 1890-х гг.
19. Внешняя политика Российской империи в конце XIX - начале XX вв
20. Экономическое и политическое развитие России в начале ХХ века
21. Революция 1905-1907 гг. в России.
22. Начало парламентаризма. Первые Государственные Думы
23. Столыпинская аграрная реформа
24. Россия в Первой мировой войне
25. Развитие России от Февраля к Октябрю 1917 года
26. Формирование советского государства
27. Россия в условиях гражданской войны
28. Переход к новой экономической политике. СССР в годы НЭПа
29. Индустриализация СССР в 1930-е гг.
30. Коллективизация сельского хозяйства
31. Развитие политической системы СССР в 1930-е гг.
32. Внешняя политика СССР в 1920-е - 1930-е гг.
33. СССР в Великой Отечественной войне.
34. Социально-экономическое и политическое развитие СССР в послевоенный период 1946-1953 гг. Усиление идеологического контроля
35. Борьба за политическое лидерство в СССР в 1953-1957 гг. ХХ съезд КПСС и разоблачение культа личности Сталина
36. Развитие СССР во второй половине 1950-х – первой половине 1960-х гг.
37. Социально-экономическое и политическое развитие СССР во второй половине 1960-х – первой половине 1980-х гг.
38. Внешняя политика СССР во второй половине XX в.
39. Перестройка и кризис советской модели общественного устройства
40. Социально-экономическое и политическое развитие России в 1990-е гг.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Классическая механика**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

Список **контрольных вопросов**

1. Сформулируйте законы Ньютона.
2. Что такое сила и масса? Как их измерить?
3. Сформулируйте принцип относительности Галилея, принцип относительности Эйнштейна и принцип постоянства скорости света.
4. Напишите формулы преобразований Лоренца, релятивистское уравнение движения.
5. Сформулируйте закон всемирного тяготения и принцип суперпозиции.
6. Дайте определения работы и потенциальной энергии. Приведите примеры потенциальных и не потенциальных сил.
7. Что такое внутренние и внешние силы? Приведите примеры.
8. Что такое центр масс системы частиц? Сформулируйте закон движения центра масс.
9. Сформулируйте законы сохранения импульса и энергии в механике Ньютона и в теории относительности.
10. Что такое момент импульса и момент силы? Сформулируйте теорему моментов и закон сохранения момента импульса.
11. Что такое момент инерции твердого тела? Приведите примеры. Сформулируйте теорему Гюйгенса – Штейнера.
12. Напишите формулы для импульса, момента импульса и кинетической энергии тела, совершающего плоское движение.
13. Напишите уравнение вращения тела.
14. Что такое силы инерции? Приведите примеры.
15. Что такое связи в механике? Приведите примеры систем со связями и без связей.
16. Что такое число степеней свободы механической системы? Приведите примеры.
17. Что такое идеальные связи? Приведите примеры.
18. Что такое лагранжиан механической системы? Запишите уравнения Лагранжа.
19. Что такое обобщенная сила и обобщенный импульс? Чем определяются их размерности? Приведите примеры.
20. Что такое гамильтониан консервативной механической системы? Запишите уравнения Гамильтона.
21. Напишите уравнение гармонических колебаний. Как найти частоту малых колебаний механической системы?
22. Приведите примеры колебательных систем с двумя степенями свободы. Что такое нормальные колебания и нормальные координаты?
23. Напишите волновое уравнение.
24. Что такое распределение плотности вероятности? Напишите формулу распределения Гиббса.
25. Напишите формулы распределения Максвелла и распределения Больцмана.
26. Сформулируйте теорему о равнораспределении энергии по степеням свободы.
27. Напишите уравнения диффузии и теплопроводности. Дайте определения коэффициентов диффузии и теплопроводности.

**Типовые задачи** для проверки знаний

**Кинематика**. Найти время, за которое свободно падающее тело проходит сотый сантиметр своего пути.

**Динамика.** Найти радиус орбиты спутника Земли, если известно, что период обращения спутника равен одним суткам.

**Законы сохранения.** Найти изменение скоростей двух тел при упругом ударе.

**Динамика твердого тела.** Найти ускорение центра цилиндра, скатывающегося по наклонной плоскости.

**Аналитическая механика.** Записать функцию Лагранжа для математического маятника.

**Колебания и волны**. Найти частоту колебаний струны.

**Статистическая механика**. Используя законы механики, вывести уравнение состояния идеального газа.

**Механика сплошной среды**. Вычислить скорость звука в воздухе при нормальных условиях.

**Теория относительности.** Вывести релятивистские правила сложения скоростей.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

Список **экзаменационных вопросов**

1. Кинематика материальной точки
2. Тангенциальное и нормальное ускорения
3. Относительность механического движения
4. Принцип относительности. Преобразования Галилея и преобразования Лоренца
5. Кинематика твердого тела
6. Матрица поворота тела
7. Кинематика вращающихся систем отсчета
8. Законы Ньютона
9. Силы в механике
10. Релятивистское уравнение движения
11. Неинерциальные системы отсчета. Силы инерции
12. Импульс частицы и системы частиц. Движение центра масс
13. Закон сохранения импульса
14. Реактивное движение
15. Работа и потенциальная энергия
16. Потенциальная энергия механических систем
17. Кинетическая энергия частицы и системы частиц
18. Кинетическая энергия твердого тела
19. Закон сохранения энергии в механике
20. Импульс и энергия в теории относительности
21. Момент импульса частицы и системы частиц. Момент силы
22. Момент импульса твердого тела
23. Теорема моментов. Закон сохранения момент импульса
24. Материальная точка в центральном поле
25. Законы Кеплера
26. Плоское движение твердого тела
27. Момент инерции твердого тела
28. Системы со связями. Степени свободы. Обобщенные координаты
29. Виртуальные перемещения. Виртуальная работа. Идеальные связи
30. Уравнения Лагранжа. Обобщенные силы
31. Функция Лагранжа. Обобщенные импульсы
32. Уравнения Гамильтона. Канонические переменные
33. Гамильтониан консервативной системы
34. Равновесие системы и его устойчивость
35. Колебания в системах с одной степенью свободы
36. Физические эффекты в колебательных системах
37. Нормальные колебания и нормальные координаты
38. Колебания струны.
39. Случайные величины и вероятности
40. Распределение Гиббса
41. Размер и масса молекул
42. Измерение постоянной Больцмана
43. Распределение энергии по степеням свободы
44. Диффузия и теплопроводность
45. Вязкость жидкости
46. Движение вязкой жидкости
47. Уравнения динамики сплошной среды
48. Звуковая волна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Электродинамика**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

Список **контрольных вопросов**

1. Фундаментальные свойства электрического заряда. Закон сохранения заряда. Сформулируйте Закон Кулона.
2. Дайте определение напряженности электрического поля. Сформулируйте принцип суперпозиции электрических полей.
3. Электростатическая теорема Гаусса. Напряженности электростатического поля равномерно заряженных сферы и бесконечной плоскости.
4. Как определяется потенциал электрического поля.
5. Запишите формулы для потенциала электрического поля дискретного и непрерывного распределений заряда.
6. Запишите формулу, показывающую локальную связь между потенциалом и напряженностью электрического поля.
7. Что такое электрический диполь. Чему равны потенциал и напряженность поля электрического диполя.
8. Чему равна циркуляция вектора напряженности электростатического поля. Приведите доказательство для системы точечных зарядов.
9. Чему равен ротор вектора напряженности электростатического поля. Приведите доказательство для системы точечных зарядов.
10. Запишите уравнения Пуассона и Лапласа для потенциала электростатического поля.
11. Свободные и связанные заряды в веществе.
12. Что такое электрическая индукция поля.
13. Сформулируйте теорему Гаусса для электрической индукции в интегральной и дифференциальной формах.
14. Материальные уравнения для электрического поля, диэлектрические восприимчивость и проницаемость.
15. Взаимная энергия системы точечных зарядов. Формулы для энергии электростатического поля и ее объемной плотности.
16. Закон Ома для участка цепи и его дифференциальная форма. Закон Джоуля-Ленца и его дифференциальная форма.
17. Сформулируйте правила Кирхгофа.
18. Запишите закон взаимодействия элементов тока – закон Ампера. Запишите закон Био-Савара-Лапласа.
19. Сформулируйте теорему о циркуляции вектора магнитной индукции в интегральной и дифференциальной формах.
20. Сформулируйте теорему Гаусса для магнитного поля в интегральной и дифференциальной формах
21. Что такое векторный потенциал. Как он связан с магнитной индукцией. Свойства векторного потенциала.
22. Сила Лоренца и характер движения заряда в постоянных электрическом и магнитном полях.
23. Сформулируйте закон электромагнитной индукции Фарадея и правило Ленца.
24. В чем заключается явление самоиндукции.
25. Чему равны собственная энергия проводника с током и энергия системы замкнутых токов.
26. Запишите формулы для энергии магнитного поля и ее объемной плотности.
27. Молекулярные токи и вектор намагниченности. Дайте определение вектора напряженности магнитного поля.
28. Сформулируйте теорему о циркуляции вектора напряженности магнитного поля (в интегральной и дифференциальной формах).
29. Что такое ток смещения.
30. Запишите уравнения Максвелла в дифференциальной и интегральной формах.
31. Запишите уравнения Максвелла в интегральной форме.
32. Дайте определение и запишите выражение для вектора Умова-Пойнтинга.
33. Получите волновое уравнение из системы уравнений Максвелла.Что такое плоская волна. Ее свойства.
34. Чему равны плотность потока энергии, плотность потока импульса и плотность потока момента импульса электромагнитной волны.
35. Излучение электромагнитных волн диполем. Зависимость излучаемой мощности от частоты.
36. Дайте определение квазистационарных электромагнитных процессов.
37. Собственные и вынужденные колебания в колебательном контуре. Формулы для амплитуды и фазы.
38. Опишите и обоснуйте метод комплексных амплитуд.
39. В чем заключается скин-эффект. Чему равна толщина скин-слоя в простейших случаях.
40. Постулаты теории относительности.Преобразования Лоренца для напряженностей электрического и магнитного полей.
41. Четырехвекторы и четырехтензоры в специальной теории относительности. Приведите примеры.
42. Тензор электромагнитного поля.
43. Инвариантная запись уравнений электродинамики.
44. Релятивистская природа силы Лоренца.
45. Инварианты электромагнитного поля.

**Типовые задачи** для проверки знаний

**Электрическое поле. Уравнения электростатики.** Точечный заряд *q* находится на расстоянии *a* от центра проводящей сферы радиусом *R* (*a*>*R*). Заряд сферы равен *Q*. Найдите силу, действующую на заряд *q*.

**Проводники и диэлектрики в электрическом поле.** Диэлектрический шар радиусом *R* равномерно заряжен по объему. Объемная плотность заряда равна *ρ*, диэлектрическая проницаемость материала шара - *ε*. Найдите потенциал поля, создаваемого шаром.

**Магнитное поле в вакууме и веществе.** Проводящая сфера радиуса *R* заряжена с поверхностной плотностью *σ*. Сфера вращается вокруг оси симметрии с угловой скоростью *ω*. Найдите индукцию магнитного поля на оси вращения.

**Закон электромагнитной индукции.** По двум металлическим параллельным рейкам, расположенным в горизонтальной плоскости и замкнутым на конденсатор емкостью , может без трения двигаться металлический стержень массой  и длиной . Вся система находится в однородном магнитном поле с индукцией , направленной вверх. К середине стержня перпендикулярно ему и параллельно рейкам приложена сила *F*. Определить ускорение *a* стержня. Сопротивлением реек, стержня и подводящих проводов пренебречь. В начальный момент скорость стержня равна нулю.

**Уравнения Максвелла.** Заряженный и отключенный от источника плоский конденсатор с круглыми пластинами медленно разряжается объемными токами проводимости, возникающими в диэлектрике между обкладками из-за наличия слабой проводимости. Пренебрегая краевыми эффектами, вычислите напряженность магнитного поля внутри конденсатора.

**Электрические цепи. Квазистационарные токи.** Два гальванических элемента с ЭДС  ии внутренними сопротивлениями  и соединены параллельно. Найдите ЭДС и внутреннее сопротивление полученной батареи.

**Электромагнитные волны:** Плоская монохроматическая световая волна распространяется в вакууме. Максимальное значение напряженности магнитного поля этой волны – *H0*. Какова средняя (за период) энергия, переносимая волной в единицу времени через поверхность полусферы радиуса *R*, основание которой перпендикулярно направлению распространения волны?

**Теория излучения:** Выведите формулу для напряженности электрического поля электромагнитной волны, излучаемой зарядом *q*, колеблющимся с частотой *ω* вдоль некоторой прямой. Амплитуда колебаний заряда – X0.

**Энергия, импульс и момент импульса электромагнитного поля:** Плоская монохроматическая электромагнитная волна нормально падает из вакуума на плоскую поверхность проводника. Чему равно среднее (за период) давление этой волны на проводник, если интенсивность волны – *I*? Считать, что волна полностью поглощается.

**Электродинамика теории относительности:** Вычислите компоненты 4-мерного ускорения. Показать, что 4-мерное ускорение ортогонально 4-мерной скорости.

Результат обучения связан со знанием определений физических понятий, размерностей физических величин и умением формулировать законы механики и электродинамики.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

Список **вопросов к зачёту**

1. Электромагнитное взаимодействие и его место среди других взаимодействий в природе. Закон сохранения электрического заряда.
2. Закон Кулона.
3. Вектор напряженности электрического поля. Принцип суперпозиции. Электростатическая теорема Остроградского–Гаусса.
4. Потенциальность электростатического поля. Связь вектора напряженности электростатического поля и потенциала.
5. Работа сил электростатического поля. Потенциал системы зарядов.
6. Теорема о циркуляции вектора напряженности электрического поля. Уравнения Пуассона и Лапласа. Электрический диполь.
7. Проводники в электростатическом поле. Распределение заряда по поверхности проводника. Связь между зарядом и потенциалом проводника. Электроемкость. Конденсаторы.
8. Диэлектрики. Свободные и связанные заряды. Вектор поляризации. Связь вектора поляризации со связанными зарядами.
9. Вектор электрической индукции в диэлектрике. Материальное уравнение для векторов электрического поля.
10. Теорема Остроградского – Гаусса для диэлектриков. Ее дифференциальная форма.
11. Граничные условия для векторов напряженности и электрической индукции.
12. Взаимодействие токов. Элемент тока. Закон Био – Савара – Лапласа. Вектор индукции магнитного поля. Закон Ампера. Сила Лоренца.
13. Теорема о циркуляции вектора индукции магнитного поля. характер магнитного поля. Векторный потенциал.
14. Элементарный ток и его магнитный момент. Магнитное поле элементарного тока. Вектор намагниченности вещества и его связь с молекулярными токами.
15. Вектор напряженности магнитного поля. Материальное уравнение для векторов магнитного поля. Диамагнетики, парамагнетики и ферромагнетики.
16. Электромагнитная индукция. Закон электромагнитной индукции Фарадея и его дифференциальная форма. Правило Ленца. Явление самоиндукции и взаимной индукции.
17. Система уравнений Максвелла. Ток проводимости и ток смещения. Высокочастотные токи. Скин-эффект.
18. Постоянный электрический ток. Плотность тока. Условие стационарности тока. Электрическое напряжение. Закон Ома в дифференциальной форме.
19. Токи в сплошных средах. Закон Джоуля – Ленца. Правила Кирхгофа.
20. Условия квазистационарности. Переходные процессы в RC- и LC-цепях.
21. Собственные колебания в контуре. Затухающие колебания. Вынужденные колебания в контуре. Метод комплексных амплитуд. Резонанс напряжений.
22. Электромагнитные волны. Волновое уравнение. Скорость электромагнитных волн.
23. Плоские и сферические гармонические электромагнитные волны в непроводящей среде. Связь векторов напряженности электрического и магнитного поля и волнового вектора в плоской волне.
24. Поляризация электромагнитной волны. Отражение электромагнитных волн.
25. Стоячая электромагнитная волна. Давление электромагнитной волны.
26. Условие калибровки Лоренца. Неоднородное волновое уравнение для векторного и скалярного потенциалов. Их решения в виде запаздывающих и опережающих потенциалов. Дипольное приближение.
27. Излучение точечного диполя. Электромагнитное поле в ближней и дальней волновой зоне. Диаграмма направленности и полная мощность излучения.
28. Энергия системы покоящихся электрических зарядов. Энергия электростатического поля и ее объемная плотность.
29. Энергия системы стационарных токов. Энергия стационарного магнитного поля и ее объемная плотность.
30. Плотность энергии нестационарного электромагнитного поля. Вектор Умова-Пойнтинга. Закон сохранения энергии в электродинамике. Плотность и поток энергии в плоской электромагнитной волне.
31. Импульс и момент импульса электромагнитного поля. Законы сохранения импульса и момента импульса электромагнитного поля.
32. Уравнения электродинамики и преобразования Галилея. Опыт Майкельсона-Морли.
33. Принцип относительности Эйнштейна и постулаты теории относительности. Преобразования Лоренца
34. Четырех-векторы и четырех – тензоры.
35. Релятивистски-инвариантная запись закона сохранения заряда и уравнений Максвелла. Релятивистски-инвариантная запись уравнений электродинамики в потенциалах.
36. Электромагнитные волны в движущейся среде. Эффект Доплера. Инварианты электромагнитного поля.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Операционные системы**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Пример вопросов коллоквиума.**

1. Определение виртуального ресурса (устройства).
2. Минимальные требования к аппаратуре для обеспечения корректного мультипрограммирования.
3. Дать определение понятия «аппарат» виртуальной памяти.
4. Основное преимущество протокола UDP по сравнению с протоколом TCP?
5. Сформулировать общее (не являющееся определением процессов в Unix) определение процесса.
6. Указать основное преимущество использование битовых массивов для учета свободных блоков файловой системы.
7. Дать определение семафора Дейкстры.
8. Для доступа взаимодействующих процессов к разделяемому ресурсу ***R*** используется семафор Дейкстры ***S***. На входе в критические секции ресурса **R** выполняется операция ***Down(S)***, на выходе ***Up(S)***. ***S*** имеет начальное значение ***N***. Какую модель доступа к ресурсу ***R*** демонстрирует этот пример?
9. Имеется программная система, реализующая с использованием сокетов модель клиент-сервер. К серверу п Написать программу на Си, выводящую на стандартное устройство вывода текст командной строки, посредством которой данная одключено ***К*** клиентов. Какое количество сокетов создано на сервере в этот момент времен?
10. Написать программу на Си, выводящую на стандартное устройство вывода текст командной строки, посредством которой данная программа была запущена.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Экзаменационные вопросы.**

- *примеры вопросов из письменной части:*

*Вариант 1.*

1.Что определяет количество записей инвертированной таблицы страниц?

2. Характеристика производительности ОЗУ – длительность цикла ОП?

3. NUMA – система. Определение и классификация.

4. Что дает расслоение ОЗУ?

5. Основное преимущество протокола UDP по сравнению с протоколом TCP?

*Вариант 2.*

1. Основное предназначение RAID1?

2. Какова структура IP адреса класса С?

3. Что такое инкрементное архивирование файловой системы?

4. Основное отличие полновесных процессов от легковесных?

5. Основное преимущество при использовании контроллеров прямого доступа при

управлении внешними устройствами (DMA)?

- *список устных вопросов*:

1. Этапы развития вычислительной техники и программного обеспечения.

2. Структура вычислительной системы. Ресурсы ВС - физические ресурсы, виртуальные ресурсы. Уровень операционной системы.

3. Структура вычислительной системы. Ресурсы ВС - физические, виртуальные. Уровень систем программирования.

4. Структура вычислительной системы. Ресурсы ВС - физические ресурсы, виртуальные ресурсы. Уровень прикладных системы.

5. Структура вычислительной системы. Понятие виртуальной машины.

6. Основы архитектуры компьютера. Основные компоненты и характеристики. Структура и функционирование ЦП.

7. Основы архитектуры компьютера. Основные компоненты и характеристики. Оперативное запоминающее устройство. Расслоение памяти.

8. Основы архитектуры компьютера. Основные компоненты и характеристики. Кэширование ОЗУ.

9. Основы архитектуры компьютера. Аппарат прерываний. Последовательность действий в вычислительной системе при обработке прерываний.

10. Основы архитектуры компьютера. Внешние устройства. Организация управления и потоков данных при обмене с внешними устройствами.

11. Основы архитектуры компьютера. Иерархия памяти.

12. Аппаратная поддержка ОС. Мультипрограммный режим.

13. Аппаратная поддержка ОС и систем программирования.. Организация регистровой памяти ЦП (регистровые окна, стек).

14. Аппаратная поддержка ОС. Виртуальная оперативная память.

15. Аппаратная поддержка ОС. Пример организации страничной виртуальной памяти.

16. Многомашинные, многопроцессорные ассоциации. Классификация. Примеры.

17. Многомашинные, многопроцессорные ассоциации. Терминальные комплексы. Компьютерные сети.

18. Операционные системы. Основные компоненты и логические функции. Базовые понятия: ядро, процесс, ресурс, системные вызовы. Структурная организация ОС.

19. Операционные системы. Пакетная ОС, ОС разделения времени, ОС реального времени, распределенные и сетевые ОС.

20. Организация сетевого взаимодействия. Эталонная модель ISO/OSI. Протокол, интерфейс. Стек протоколов. Логическое взаимодействие сетевых устройств.

21. Организация сетевого взаимодействия. Семейство протоколов TCP/IP, соответствие модели ISO/OSI. Взаимодействие между уровнями протоколов семейства TCP/IP. IP адресация.

22. Управление процессами. Определение процесса, типы. Жизненный цикл, состояния процесса. Свопинг. Модели жизненного цикла процесса. Контекст процесса.

23. Реализация процессов в ОС UNIX. Определение процесса. Контекст, тело процесса. Состояния процесса. Аппарат системных вызовов в ОС UNIX.

24. Реализация процессов в ОС UNIX. Базовые средства управления процессами в ОС UNIX. Загрузка ОС UNIX, формирование нулевого и первого процессов.

25. Взаимодействие процессов. Разделяемые ресурсы. Критические секции. Взаимное исключение. Тупики.

26. Взаимодействие процессов. Некоторые способы реализации взаимного исключения: семафоры Дейкстры, мониторы, обмен сообщениями.

27. Взаимодействие процессов. Классические задачи синхронизации процессов. “Обедающие философы”.

28. Взаимодействие процессов. Классические задачи синхронизации процессов. “Читатели и писатели”.

29. Базовые средства взаимодействия процессов в ОС UNIX. Сигналы. Примеры программирования.

30. Базовые средства взаимодействия процессов в ОС UNIX. Неименованные каналы. Примеры программирования .

31. Базовые средства взаимодействия процессов в ОС UNIX. Именованные каналы. Примеры программирования.

32. Базовые средства взаимодействия процессов в ОС UNIX. Взаимодействие процессов по схеме ”подчиненный-главный”. Общаясхема трассировки процессов.

33. Система межпроцессного взаимодействия ОС UNIX. Именование разделяемых объектов. Очереди сообщений. Пример.

34. Система межпроцессного взаимодействия ОС UNIX . Именование разделяемых объектов. Разделяемая память. Пример.

35. Система межпроцессного взаимодействия ОС UNIX . Именование разделяемых объектов. Массив семафоров. Пример.

36. Сокеты. Типы сокетов. Коммуникационный домен. Схема работы с сокетами с установлением соединения.

37. Сокеты. Схема работы с сокетами без установления соединения.

38. Общая классификация средств взаимодействия процессов в ОС UNIX.

39. Файловые системы. Структурная организация файлов. Атрибуты файлов. Основные правила работы с файлами. Типовые программные интерфейсы работы с файлами.

40. Файловые системы. Модели реализации файловых систем. Понятие индексного дескриптора.

41. Файловые системы. Координация использования пространства внешней памяти. Квотирование пространства ФС. Надежность ФС. Проверка целостности ФС.

42. Примеры реализаций файловых систем. Организация файловой системы OC UNIX. Виды файлов. Права доступа. Логическая структура каталогов.

43. Примеры реализаций файловых систем Внутренняя организация ФС. Модель версии UNIX SYSTEM V.

44. Примеры реализаций файловых систем. Внутренняя организация ФС. Принципы организации файловой системы FFS UNIX BSD.

45. Управление внешними устройствами. Архитектура организации управления внешними устройствами, основные подходы, характеристики.

46. Управление внешними устройствами. Планирование дисковых обменов, основные алгоритмы.

47. Управление внешними устройствами. Организация RAID систем, основные решения, характеристики.

48. Внешние устройства в ОС UNIX. Типы устройств, файлы устройств, драйверы.

49. Внешние устройства в ОС UNIX. Системная организация обмена с файлами. Буферизация обменов с блокоориентированными устройствами.

50. Управление оперативной памятью. Одиночное непрерывное распределение. Распределение разделами. Распределение перемещаемыми разделами.

51. Управление оперативной памятью. Страничное распределение.

52. Управление оперативной памятью. Сегментное распределение.

53. Вычислительная система. Кэширование информационных потоков на уровнях аппаратуры и ОС.

54. Язык программирования С. Общая характеристика. Типы, данные, классы памяти. Правила видимости. Структура программы. Препроцессор. Интерфейс с ОС UNIX.

**Примеры экзаменационных билетов.**

**Билет 3.**

**1.**Структура вычислительной системы. Понятие виртуальной машины.

**2.**Система межпроцессного взаимодействия ОС UNIX. Именование разделяемых объектов. Очереди сообщений. Пример.

**3.** Управление внешними устройствами. Организация RAID систем, основные решения, характеристики.

**Билет 16.**

**1.** Основы архитектуры компьютера. Аппарат прерываний. Последовательность действий в вычислительной системе при обработке прерываний.

**2.**Файловые системы. Модели реализации файловых систем. Понятие индексного дескриптора.

**3.**Базовые средства взаимодействия процессов в ОС UNIX. Сигналы. Примеры программирования.

**Билет 23.**

**1.**Операционные системы. Основные компоненты и логические функции. Базовые понятия: ядро, процесс, ресурс, системные вызовы. Структурная организация ОС.

**2.** Взаимодействие процессов. Классические задачи синхронизации процессов. “Читатели и писатели”.

**3.**Управление оперативной памятью. Одиночное непрерывное распределение. Распределение разделами. Распределение перемещаемыми разделами.

**Методические материалы для проведения процедур оценивания результатов обучения**

Используется дифференцированная система оценки знаний и навыков. Оценка основывается на:

* контроле посещаемости занятий;
* результатах сдачи коллоквиума;
* результате сдачи экзамена.

Экзамен состоит из двух частей – письменной и устной. Результаты письменной части экзамена учитываются в дифференцированной оценке. При этом в порядке исключения, учитывая интегральную высокую оценку работы в семестре, часть студентов может получить «отличные» и «хорошие» оценки за экзамен (без сдачи устной части), часть может получить оценку «неудовлетворительно», а остальная часть направляется на сдачу устной части экзамена. Для студентов, получивших оценку «неудовлетворительно» сразу проводится консультация-разъяснение с возможностью апелляции оценки.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Системы программирования**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Письменный коллоквиум**

|  |
| --- |
| **Вариант 1** |
| 1. Вычеркните только те строки функции *main ()*, которые приводят к синтаксической ошибке (если таковые имеются). Обязательно **объясните** причины ошибок. Что будет напечатано в результате работы получившейся программы.   **struct** A { **int** main () {  **int** n; A a;  A(){ n = 9;} B b;  A **operator**+(**const** A & a){ a = a + b;  A t; cout << a.n << '\n';  t.n = n + a.n; b = b + a;  **return** t; } cout << b.n << '\n';  }; **return** 0;  **struct** B:A { }  B **operator**+(**const** B & a){  B t;  t.n = n + a.n + 10;  **return** t; }  };   1. Исправьте **только описание класса А** так, чтобы в приведенной ниже программе не было ошибок, а на экран напечаталось **f ( int, int )**.   **struct** A { **int** n;  A (**int** m) { n = m; }  **operator int** () { **return** 1; }  };  **void** f ( **int** i, **int** j ) { cout << "f(int, int)\n";}  **void** f ( A b, A a ) { cout << "f(A, A)\n";}    **int** main () { A a (1);  f (a, 1);  **return** 0; }     1. Опишите необходимые классы так, чтобы в заданной функции *main ()* не было ошибок, а на экран печаталось **310.**   **int** main () {  T t (3);  P <T> p1(&t), p2(0);  cout << p1 -> n << p2 -> n << endl;  **return** 0 ; }     1. Добавьте в описание класса А **из задачи 1** метод   **virtual** A& **operator** - (){ n = - (n+1)**;** cout << n <<" А::**operator** - ()\n"; **return \*this**;},  а в описание (производного от класса А) класса В из задачи 1 -  В& **operator** - () { cout << n << " B::**operator** - () \n"; **return \*this**;}  Что напечатает программа с нижеприведенной функцией *main()* и получившимися классами А и В?  **int** main () {  B b;  A a, & ra = b;  a = -a;  ra = -ra;  ra = a;  ra = -ra; **return** 0;  }   1. Что такое абстрактный класс в С++? Обязательно приведите **пример** описания и использования абстрактного класса. 2. Укажите все прототипы конструкторов и деструкторов в порядке их выполнения в следующей программе:   **class** A {};  **class** B: **public** A{};  **struct** D { A a; };    **int** main () {  D d;  { d.a = B();  B b; }  A a1, a2 = a1;  **return** 0;  }     1. Добавьте в следующую программу необходимые описания так, чтобы в ней не было ошибок.   **int** main () {  **const** A a;  a.x = 3;  cout << a.y << a.x << a.f(1) << endl;  **return** 0; }  **8**. Модифицируйте функцию *main()*, **ничего в ней не удаляя, не используя комментарии, goto и прочие переходы** так, чтобы программа завершалась нормально (не аварийно).    **struct** B { **virtual** **void** f () { cout << “B::f()\n”; } };  **struct** D : B { **void** f () { cout << “D::f()\n”; } };  **int** main () {  D d, &rd = d;  B b, & rb = b, &rbd = d;  rd = **dynamic\_cast** <D&> (rbd); rd.f();  rd = **dynamic\_cast** <D&> (rb); rd.f();  **return** 0;  }   1. **С++11.** Вычеркните неверные конструкции в следующей программе на С++11. Обязательно **объясните** причины ошибок в вычеркнутых конструкциях.   **struct** A {  **int** i = 10;  **};**  **void** f (A && x) { }  **int** main () {  A a;  **int** && n, m;  **auto** b = A();  **decltype** (a) c;  f (b);  f ( A() );  m = **nullptr**;  **return** 0;  }  **10**. **STL.** Напишите шаблонную функцию, подсчитывающую сумму пяти последних элементов заданного контейнера (списка или вектора, константного или неконстантного). Тип элементов контейнера является числовым типом. Если в заданном контейнере меньше 5 элементов, функция должна вернуть 0 соответствующего типа. |
| **Вариант 2** |
| **1.** Вычеркните только те строки функции *main ()*, которые приводят к синтаксической ошибке (если таковые имеются). Обязательно **объясните** причины ошибок. Что будет напечатано в результате работы получившейся программы.  **struct** Т { **int** main () {  **int** i;  T(){i = 3;} Т a(5);  Т(int t){ i = t;} Р b;  Т **operator**\*(**const** Т & a){ a = a \* b;  Т t; cout << a.i << '\n';  t.i = i \* a.i; b = b \* a;  **return** t; } cout << b.i << '\n';  }; **return** 0;  **struct** Р:Т { }  Р **operator**\*(**const** Р & a){  Р t;  t.i = i \* a.i \* 2;  **return** t; }  };   1. Исправьте **только описание класса А** так, чтобы в приведенной ниже программе не было ошибок, а на экран напечаталось **f ( A, A )**.   **struct** A { **int** n;  A (**int** m) { n = m; }  **operator int** () { **return** 1; }  };  **void** f ( **int** i, **int** j ) { cout << "f(int, int)\n";}  **void** f ( A b, A a ) { cout << "f(A, A)\n";}    **int** main () { A a (1);  f (a, 1);  **return** 0; }   1. Опишите необходимые классы так, чтобы в заданной функции *main ()* не было ошибок, а на экран печаталось **520.**   **int** main () {  C c (5);  B <C> b1(&c), b2(0);  cout << (\*b1). n << (\*b2). n << endl;  **return** 0 ; }   1. Добавьте в описание класса T **из задачи 1** метод   **virtual** T& **operator --** (){ i = i - 1; cout << i <<" T::**operator** -- () \n"; **return \*this**;} ,  а в описание (производного от класса T) класса P из задачи 1 -  P& **operator --** () { cout << i << " P::**operator** -- () \n"; **return \*this**; }  Что напечатает программа с нижеприведенной функцией *main()* и получившимися классами T и P?  **int** main () {  P b;  T a(5), & rt = b;  --a;  --rt;  rt = a;  --rt; **return** 0;  }  **5**. Что такое функция-друг в языке Си++? Приведите **пример** описания и использования функции-друга.   1. Укажите все прототипы конструкторов и деструкторов в порядке их выполнения в следующей программе:   **class** A {};  **class** B: **public** A{};  **struct** T { B \* pb; };    **int** main () {  A a;  { a = B();  T t; }  B b1, b2 = b1;  **return** 0; }   1. Добавьте в следующую программу необходимые описания так, чтобы в ней не было ошибок.   **int** main () {  B::y = 3 ;  **const** B b;  cout << b.x-B::h(2)<< b.h(0)<< endl;  **return** 0; }  **8**. Модифицируйте функцию *main()*, **ничего в ней не удаляя, не используя комментарии, goto и прочие переходы** так, чтобы программа завершалась нормально (не аварийно).    **struct** B { **virtual void** g() { cout << "B::g () \n"; } };  **struct** D : B { };    **int** main () {  D d, \* pd1, \*pd2;  B b, \* pb = &b, \* pbd = &d;  pd1 = **dynamic\_cast** < D\* > (pbd);  pd2 = **dynamic\_cast** < D\* > (pb);  **if** ( **typeid** (\*pd1) == **typeid** (\*pd2) ) pb -> g () ;  **return** 0; }  **9.С++11.** Вычеркните неверные конструкции в следующей программе на С++11. Обязательно **объясните** причины ошибок в вычеркнутых конструкциях.  **struct** S {  **int** x = 1;  **};**  **void**  g (S && s) { cout << s.x << endl;}  **int** main () {  **int** && i;  **auto** k = **nullptr**;  **decltype** ( S ( ) ) a;  **char** \* s = k;  k = s;  g (a);  g ( S() );  **return** 0;  }  **10**. **STL.** Напишите шаблонную функцию, подсчитывающую сумму первых семи элементов заданного контейнера (списка или вектора, константного или неконстантного). Тип элементов контейнера является числовым типом. Если в заданном контейнере меньше 7 элементов, функция должна вернуть 0 соответствующего типа |

**Вопросы к коллоквиуму**

1. Абстрактные типы данных, инкапсуляция, наследование, полиморфизм.
2. Класс, объект, состояние объекта, поведение объекта.
3. С++: Пространства имен. Пространство имен *std*.
4. С++: Конструкторы и деструкторы.
5. С++: Присваивание и инициализация.
6. С++: Ссылки в С++. Передача параметров по ссылке.
7. С++: Манипуляции с состоянием объекта.
8. С++: Работа с динамической памятью.
9. С++: Друзья класса.
10. С++: Статические члены класса.
11. Виды полиморфизма в С++ (статический, динамический, параметрический).
12. С++: Статический полиморфизм. Перегрузка бинарных операций:
    1. с помощью функции-члена класса
    2. с помощью функции-друга класса
13. С++: Статический полиморфизм. Перегрузка унарных операций:
    1. с помощью функции-члена класса
    2. с помощью функции-друга класса
14. С++: Специфика перегрузки операций инкремента и декремента, операции индексации.
15. С++: Статический полиморфизм. Перегрузка функций.
16. С++: Алгоритм поиска оптимально отождествляемой (best-matching) функции.
17. С++: Средства обработки ошибок. Исключения и обработка исключений.
18. Виды отношений между классами (ассоциация, наследование, агрегация, использование).
19. С++: Одиночное наследование. Правила наследования. Видимость при наследовании.
20. С++: Динамический полиморфизм. Виртуальные функции.
21. Принципы реализации виртуальных функций
22. С++: Абстрактные классы.
23. С++: Множественное наследование. Видимость при множественном наследовании. Виртуальные базовые классы.
24. С++: Динамическая информация о типе (RTTI).
25. С++: Параметрический полиморфизм. Шаблонные функции.
26. С++: Шаблонные классы.
27. Стандартная библиотека С++.
28. Стандартная библиотека шаблонов STL.
29. STL: контейнеры, итераторы, алгоритмы, аллокаторы.
30. STL: Шаблонные классы *vector* и *list*.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Экзаменационная работа (письменный экзамен)**

|  |
| --- |
| **Вариант 1** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **1.** Дана грамматика *G*:  *S* → *aS |Sb|aAb| ε*  *A* → *aaAbb|aabb*  *A* → *A* | (a) Описать язык *L*(*G*) в виде теоретико-множественной формулы:  **Ответ:**  *L*(*G*) = | (б) Каким из перечисленных классов грамматик принадлежит *G*? **Ответ:**   |  |  | | --- | --- | | Класс П | *G* ∈ П? (да/нет) | | регулярные |  | | контекстно-зависимые |  | | контекстно-свободные |  | | грамматики типа 0 |  | | неукорачивающие |  | |   (в) Тип языка: найти такое целое *k*, что *L(G)* является языком типа *k*, но не языком типа *k+*1.  **Ответ:** *k* =  **2.** Является ли однозначной данная грамматика *G*, порождающая язык цепочек в алфавите {0,1}, в которых символов 0 и 1 поровну? Ответ обосновать.  *G: S* → *0S1S |1S0S | 01BB| ε*  *B* → *BB|B01*     |  |  |  | | --- | --- | --- | | **3.** | Тесей должен пройти из левого верхнего угла лабиринта в правый нижний, чтобы убить Минотавра. Нить Ариадны оставляет след перемещений Тесея. Он умеет за один шаг перемещаться на клетку вниз (след обозначается символом *a*), вверх ( след *b*) или вправо (след *с*). Влево не умеет. Каждая клетка посещается не более одного раза. Постройте грамматику, порождающую язык в алфавите {*a*, *b, с*} всевозможных путей Тесея, приводящих к цели. Например, цепочка *ссссaaaaac* приведет Тесея к Минотавру. В грамматике должно быть не более 4-х правил вывода, считая альтернативы. |  |   **4.** Дан список слов: *в*, *и*, *исправление*, *обнаружение*, *ошибки*, *программа*. Составьте из него, употребив слова в нужном порядке и форме, определение термина (понятия) из раздела СП и назовите сам термин (он не входит в заданный список).  **5.** Можно ли считать утилиту *make* системы *Unix*  компилятором? Обоснуйте ответ.  **6.** По заданной грамматике *G =* {{*a, b*}*,* {*B, S*}*, P, S*} получить эквивалентную неукорачивающую контекстно-свободную грамматику (использовать алгоритм устранения правил с пустой правой частью).  *P: S → aBb | ε* **Ответ:**  *B → aB | bS | SS*  **7.** Даны 1) функция-анализатор на языке Си++ для леволинейной грамматики *GL*:  **bool** scan\_G ()  { **enum** state { H, B, S, ER }; // множество состояний  state CS= H; // CS —— текущее состояние  **do**  { gc (); // считывает символ в глобальный объект с  **switch** (CS) {  **case** H: **if** ( c == 'a' ) CS = B;  **else** CS = ER; **break**;  **case** B: **if** ( c == 'b' );  **else** **if** ( c == '⊥' ) CS = S;  **else** CS = ER; **break**;  }  }  **while** ( CS != S && CS != ER);  **return** CS == S; // true, если CS != ER, иначе false  }  2) заготовка диаграммы состояний праволинейной грамматики *GR:*   |  |  | | --- | --- | | *S*  *A*  *D*  *F*  *C*  *⊥*  *⊥*  *⊥* | (а) Расставить в заготовке диаграммы для *GR*терминальные символы так, чтобы грамматики *GR* и *GL* были эквивалентны.  (б) Восстановить грамматику *GR.* **Ответ:**  (в) Сколько состояний будет иметь конечный автомат, полученный алгоритмом детерминизации диаграммы для *GR* ? **Ответ:\_\_\_\_** |  |  |  | | --- | --- | | ­­­**8.** Дана КС-грамматика *G*, порождающая язык *L.*  Вставить в грамматику действия вида **〈***cout <<* ′′символы′′ ;**〉** так, чтобы в процессе рекурсивного спуска был реализован перевод *τ* = {(ω, *a* k*b* k+n)| ω∈ *L* , *k* = |ω|*a* , *n* = |ω|*b* }. | *G*:  *S* → *aA |bB |* ε  *A* → *сA | S*  *B* → *сB| S* |   **9.** Дана грамматика *G*. Докажите, что метод рекурсивного спуска непримени́м к ней. Можно ли вычеркнуть один терминальный символ в правилах грамматики так, чтобы к получившейся грамматике метод был бы примени́м?  *G: S* → *aSb | Yb | bb*  *Y* → *cSY | bd | ε*  **10.** Постройте ПОЛИЗ для фрагмента программы на Си. Префиксный ++ в польской записи обозначается как +#, постфиксный как #+.  **for** ( i = 0, j = 10; i == j ? 0 : 1; --j) a += 2 < i++ > j;   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ПОЛИЗ | | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | | 18 | | 19 | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | | 28 | | 29 | | 30 | | 31 | | 32 | | 33 | | 34 | | 35 | | 36 | | 37 | | 38 | 39 | |  | |  |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |  | |
| **Вариант 2** |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **1.** Дана грамматика *G*:  *S* → *сA |сAb|сb|* ε  *сAb* → *сAbb |сb|cсAb*  *ссAbb* → *ссAbb* | (a) Описать язык *L*(*G*) в виде теоретико-множественной формулы:  **Ответ:**  *L*(*G*) = | (б) Каким из перечисленных классов грамматик принадлежит *G*? **Ответ:**   |  |  | | --- | --- | | Класс П | *G* ∈ П? (да/нет) | | регулярные |  | | контекстно-зависимые |  | | контекстно-свободные |  | | грамматики типа 0 |  | | неукорачивающие |  | |   (в) Тип языка: найти такое целое *k*, что *L(G)* является языком типа *k*, но не языком типа *k+*1.  **Ответ:** *k* =  **2.** Является ли однозначной данная грамматика *G* , порождающая язык цепочек в алфавите {*a*,*b*}? Ответ обосновать.  *G: S* → *aSaS |aBa |aSa| A*  *B* → *BB|Ba*  *A* →*b|* ε     |  |  |  | | --- | --- | --- | | **3.** | Баба-Яга дала Иванушке клубочек, который через лесной лабиринт проведет его к сундуку, где находится смерть Кащеева. Клубочек (а за ним и Иванушка) умеет за один шаг перемещаться на клетку вниз (оставляемый след обозначается символом *a*), вверх ( след *b*) или вправо (след *с*). Влево не умеет. Каждая клетка посещается не более одного раза. Постройте грамматику, порождающую язык в алфавите {*a*, *b, с*} всевозможных путей Иванушки, приводящих к цели. Иванушка начинает путь из левого нижнего угла. Цель – в правом верхнем. Например, цепочка *сbbcbbbccc* приведет Иванушку к цели. В грамматике должно быть не более 4-х правил вывода, считая альтернативы. |  |   **4.** Дан список слов: *входные*, *заранее*, *данные*, *для*, *известный, программа, работа*, *результат,с*. Составьте из него, употребив слова в нужном порядке и форме, определение термина (понятия) из раздела СП и назовите сам термин (он не входит в заданный список).  **5.** Можно ли считать утилиту *make* системы *Unix*  интерпретатором? Обоснуйте ответ.  **6.** По заданной грамматике *G =* {{*a, c*}*,* {*B, S*}*, P, S*} получить эквивалентную неукорачивающую контекстно-свободную грамматику (использовать алгоритм устранения правил с пустой правой частью).  **Ответ:**  *P: S → aBc | ε*  *B → aB | cS | SaS*  **7.** Даны 1) функция-анализатор на языке Си++ для леволинейной грамматики *GL*:  **bool** scan\_G ()  { **enum** state { H, A, S, ER }; // множество состояний  state CS= H; // CS —— текущее состояние  **do**  { gc (); // считывает символ в глобальный объект с  **switch** (CS) {  **case** H: **if** ( c == 'a'||c ==’b’ ) CS = A;  **else** CS = ER; **break**;  **case** A: **if** ( c == 'a' );  **else** **if** ( c == '⊥' ) CS = S;  **else** CS = ER; **break**;  }  }  **while** ( CS != S && CS != ER);  **return** CS == S; // true, если CS != ER, иначе false  }    2) заготовка диаграммы состояний праволинейной грамматики *GR:*   |  |  | | --- | --- | | *S*  *B*  *D*  *F*  *C*  *⊥*  *⊥*  *⊥* | (а) Расставить в заготовке диаграммы для *GR*терминальные символы так, чтобы грамматики *GR* и *GL* были эквивалентны.  (б) Восстановить грамматику *GR.* **Ответ:**  (в) Сколько состояний будет иметь конечный автомат, полученный алгоритмом детерминизации диаграммы для *GR* ? **Ответ:\_\_\_\_** |  |  |  | | --- | --- | | **8.** Дана КС-грамматика *G*, порождающая язык *L.*  Вставить в грамматику действия вида **〈***cout <<* ′′символы′′ ;**〉** так, чтобы в процессе рекурсивного спуска был реализован перевод *τ* = {(ω, *a* k*d* k-n)| ω∈ *L* , *k* = |ω|, *n* = |ω|*d* }. | *G*:  *S* → *aA |dD |* ε  *A* → *сA | S*  *D* → *сD| S* |   **9.** Дана грамматика *G*. Докажите, что метод рекурсивного спуска непримени́м к ней. Можно ли вычеркнуть один терминальный символ в правилах грамматики так, чтобы к получившейся грамматике метод был бы примени́м?  *G: S* → *aSb | Xa | bb*  *X* → *cSX |bd | ε*  **10.** Постройте ПОЛИЗ для фрагмента программы на Си. Префиксный ++ в польской записи обозначается как +#, постфиксный как #+.  i = 0, j = 10; **do** a += 2 < i++ > --j; **while** ( i == j ? 0 : 1 );   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ПОЛИЗ | | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | | 14 | | 15 | | 16 | | 17 | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | | 18 | | 19 | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | | 28 | | 29 | | 30 | | 31 | | 32 | | 33 | | 34 | | 35 | | 36 | | 37 | | 38 | 39 | |  | |  |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |  | |

**Вопросы к экзамену**.

1. Этапы жизненного цикла программного продукта.
2. Состав и схема функционирования классической системы программирования.
3. Типы трансляторов, особенности интерпретаторов и компиляторов. Смешанная стратегия трансляции.
4. Общая схема работы компилятора.
5. Основные понятия теории формальных грамматик и языков.
6. Эквивалентные грамматики.
7. Классификация формальных грамматик и языков по Хомскому.
8. Соотношения между типами грамматик.
9. Соотношения между типами языков.
10. Разбор цепочек по КС-грамматикам. Задача разбора. Дерево вывода.
11. Неоднозначность грамматик и языков.
12. Недостижимые и бесплодные символы грамматики. Алгоритмы удаления недостижимых и бесплодных символов. Приведенная грамматика. Алгоритм приведения грамматики.
13. Алгоритм удаления правил с пустой правой частью.
14. Определение недетерминированного конечного автомата (НКА).
15. Диаграмма состояний (ДС) конечного автомата.
16. Леволинейные регулярные грамматики и конечные автоматы.
17. Определение детерминированного конечного автомата (ДКА).
18. Алгоритм построения детерминированного конечного автомата по НКА.
19. Задачи лексического анализа.
20. Лексический анализ на основе регулярных грамматик.
21. Задачи синтаксического анализа.
22. Метод рекурсивного спуска (РС-метод): назначение, семантика процедур метода рекурсивного спуска.
23. Достаточные условия применимости метода рекурсивного спуска для грамматик без ε-альтернатив и для грамматик с ε-альтернативами.
24. Критерий применимости РС-метода.
25. Задачи семантического анализа. Грамматики с действиями.
26. Свойства языка внутреннего представления программы, примеры таких языков.
27. Синтаксически управляемый перевод: идея, принципы организации, примеры. Определение формального перевода.
28. ПОЛИЗ выражений.
29. ПОЛИЗ операторов языков программирования.
30. Генерация ПОЛИЗа выражений и операторов.
31. Интерпретация ПОЛИЗа.
32. Основные стратегии распределения памяти.
33. Машинно-независимая и машинно-зависимая оптимизация. Примеры оптимизирующих преобразований.
34. Интегрированная среда разработки программного обеспечения (ИСР).
35. Основные функции редактора текстов в рамках ИСР. Примеры его интегрированности с другими компонентами ИСР.
36. Отладчики, их возможности. Примеры интегрированности отладчика с другими компонентами ИСР.
37. Редактор внешних связей, его назначение и принципы работы. Загрузчик.
38. Профилировщики. Назначение. Принцип работы.
39. Библиотеки. Основные типы библиотек.
40. Критерии проектирования стандартных библиотек.
41. Способы подключения библиотек функций.

**Типовые задания для экзамена**.

1. Определение типа по Хомскому заданной грамматики.
2. Определение типа языка, порождаемого заданной грамматикой.
3. Построение грамматики [определенного типа], порождающей заданный язык.
4. Определение языка, порождаемого заданной грамматикой.
5. Построение дерева вывода (разбора) заданной цепочки по заданной КС-грамматике.
6. Построение приведенной грамматики.
7. Устранение пустых правых частей заданной КС-грамматики.
8. Построение ДС конечного автомата по заданной леволинейной грамматике.
9. Построение анализатора на C++ по заданной ДС.
10. Восстановление леволинейной регулярной грамматики, порождающей язык, распознаваемый конечным автоматом, заданным в виде ДС.
11. Восстановление леволинейной регулярной грамматики по заданному тексту анализатора на C++.
12. Построение ДС по праволинейной грамматике и построение праволинейной грамматики по заданной ДС.
13. Преобразование леволинейной грамматики в эквивалентную праволинейную с помощью ДС.
14. Преобразование НКА в эквивалентный ДКА с помощью соответствующего алгоритма.
15. Определение и обоснование применимости метода рекурсивного спуска к заданной КС-грамматике.
16. Построение (на C++) анализатора методом рекурсивного спуска для заданной КС-грамматики.
17. Восстановление КС-грамматики по заданному анализатору, построенному методом рекурсивного спуска.
18. Определение языка, порождаемого грамматикой с действиями.
19. Дополнение заданной КС-грамматики действиями, позволяющими учесть дополнительные ограничения на цепочки определяемого ею языка.
20. Вставка в заданную (или построенную) грамматику языка *L*1 действий по переводу цепочек этого языка в цепочки языка *L*1 по заданному закону соответствия между цепочками (т. е. реализовать формальный перевод).
21. Определение языков *L*1 и *L*1 по заданной грамматике с действиями, реализующей формальный перевод языка *L*1 в язык *L*1.
22. Запись в ПОЛИЗе заданного фрагмента программы на заданном языке.
23. Восстановление текста на заданном языке заданного фрагмента программы в ПОЛИЗе.
24. Определение, является ли данная запись ПОЛИЗом заданной конструкции.
25. Вставка в грамматику, порождающую некоторую конструкцию языка программирования, действий, осуществляющих синтаксически управляемый перевод этой конструкции в ПОЛИЗ при анализе РС-методом.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Уравнения математической физики**

**Контрольные работы**

**КОНТРОЛЬНАЯ РАБОТА №1.**

**КЛАССИФИКАЦИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ. ЗАДАЧИ ДЛЯ УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ.**

**1. Найдите области гиперболичности, эллиптичности и параболичности уравнения**

****

**и в каждой из этих областей приведите его к канонической форме.**

**2. Найдите функцию , являющуюся решением задачи**

, , ;

, , ;

, .

**3. Найдите функцию , являющуюся решением задачи**

, , ;

 , .

**4. Найдите функцию , являющуюся решением задачи**

, , ;

, ;

, .

**Выразите**  **через функцию ошибок** . **Найдите** .

**5. Запишите общий вид решения задачи**

, , ; ;

, .

**КОНТРОЛЬНАЯ РАБОТА №2.**

**ЗАДАЧИ ДЛЯ УРАВНЕНИЙ ЛАПЛАСА И ПУАССОНА. ЗАДАЧИ ДЛЯ УРАВНЕНИЯ КОЛЕБАНИЙ**

**1. Найдите функцию** , **являющуюся решением задачи**

**,** , , ,  – полярные координаты на плоскости;

, , ;

, .

**2.**  **Найдите функцию** , **являющуюся решением задачи**

**,** , ,

,  – декартовы координаты на плоскости;

, ,

, .

**3. Методом зеркальных изображений постройте функцию Грина задачи Дирихле в области** ,,  **в пространстве** .

**4. Найдите функцию** , **являющуюся решением задачи**

, , ;

, 

**Постройте график функции** .

**5. Найдите функцию** , **являющуюся решением задачи**

, , ;

, ;

, ,.

**6. Найдите функцию** , **являющуюся решением задачи**

, , ;

, , ;

, ,.

**Список определений и теорем**

1. Определения канонических форм дифференциальных уравнений с частными производными второго порядка линейных относительно старших производных, зависящих от двух независимых переменных.

2. Определение классического решения первой начально-краевой задачи для уравнения теплопроводности.

3. Определение классического решения задачи Коши для уравнения теплопроводности.

4. Определения фундаментальных решений уравнения Лапласа в пространстве и на плоскости.

5. Определение функции Грина для задачи Дирихле.

6. Определения потенциалов простого слоя и двойного слоя.

7. Определение интеграла энергии.

8. Определение сопряженного дифференциального оператора.

9. Теорема существования решения первой начально-краевой задачи для уравнения теплопроводности.

10. Принцип максимума для уравнения теплопроводности.

11. Теорема единственности решения первой начально-краевой задачи для уравнения теплопроводности.

12. Теорема устойчивости решения первой начально-краевой задачи для уравнения теплопроводности.

13. Теорема единственности решения общей начально-краевой задачи для уравнения теплопроводности на отрезке.

14. Теорема существования решения задачи Коши для уравнения теплопроводности на прямой.

15. Свойства гармонических функций.

16. Принцип максимума для гармонических функций.

17. Теорема единственности решения внутренней задачи Дирихле для уравнения Лапласа.

18. Теорема устойчивости решения внутренней задачи Дирихле для уравнения Лапласа.

19. Необходимое условие разрешимости внутренней задачи Неймана для уравнения Лапласа.

20. Теорема единственности решения внутренней задачи Неймана для уравнения Лапласа.

21. Теоремы единственности решений внешних задач Дирихле для уравнения Лапласа в пространстве и на плоскости.

22. Свойства функции Грина для задачи Дирихле.

23. Теорема существования решения внутренней задачи Дирихле для уравнения Лапласа.

24. Теорема существования решения задачи Коши для уравнения колебаний.

25. Теорема единственности решения задачи Коши для уравнения колебаний.

26. Теорема устойчивости решения задачи Коши для уравнения колебаний.

27. Теорема существования решения первой начально-краевой задачи для уравнения колебаний на отрезке.

28. Теорема единственности решения первой начально-краевой задачи для уравнения колебаний на отрезке.

29. Теорема существования решения задачи для уравнения гиперболического типа с данными на характеристиках.

30. Теорема единственности решения задачи для уравнения гиперболического типа с данными на характеристиках.

**Вопросы к экзамену**

1. Классификация уравнений с частными производными второго порядка.

2. Вывод уравнения теплопроводности в пространстве.

3. Уравнение теплопроводности с одной пространственной переменной. Постановка основных задач.

4. Метод разделения переменных для доказательства существования решения первой начально-краевой задачи для уравнения теплопроводности.

5. Принцип максимума для уравнения теплопроводности.

6. Единственность и устойчивость решения первой начально-краевой задачи для уравнения теплопроводности.

7. Единственность решения общей начально-краевой задачи для уравнения теплопроводности на отрезке.

8. Единственность решения задачи Коши для уравнения теплопроводности.

9. Существование решения задачи Коши для уравнения теплопроводности.

10.Метод продолжения решения первой и второй начально-краевой задачи для уравнения теплопроводности на полупрямой.

11. Уравнения Лапласа и Пуассона. Постановка основных задач. Фундаментальные решения уравнения Лапласа.

12. Первая и вторая формулы Грина. Третья (основная) формула Грина.

13. Свойства гармонических функций.

14. Принцип максимума для гармонических функций.

15. Единственность и устойчивость решения внутренней задачи Дирихле для уравнения Лапласа.

16. Единственность решения внутренней задачи Неймана для уравнения Лапласа и необходимое условие ее разрешимости.

17. Единственность решения внешней задачи Дирихле для уравнения Лапласа в двух и трехмерных случаях.

18. Свойства функции Грина для задачи Дирихле.

19 .Потенциалы простого и двойного слоя. Потенциал двойного слоя с единичной плотностью.

20 .Сведение внутренней задачи Дирихле к интегральному уравнению Фредгольма 2 рода.

21. Существование решения внутренней задачи Дирихле.

22. Уравнение колебаний. Постановка основных задач.

23. Формула Даламбера. Существование, единственность и устойчивость решения задачи Коши для уравнения колебаний.

24. Метод разделения переменных для доказательства существования решения первой начально-краевой задачи для уравнения колебаний.

25. Интеграл энергии. Теоремы единственности решения начально-краевых задач для уравнения колебаний.

26. Задачи с данными на характеристиках. Эквивалентная система интегральных уравнений.

27. Существование решения задачи с данными на характеристиках.

28. Единственность решения задачи с данными на характеристиках.

29. Сопряженный дифференциальный оператор. Примеры.

30. Метод Римана.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Базы данных**

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Вопросы к экзамену**.

1. Файловые системы. Особенности организации устройств внешней памяти на магнитных дисках. Структуры файлов на дисках. Способы организации архивов файлов. Принципы именования.

2. Файловые системы. Способы авторизации доступа к файлам. Организация мультидоступа.

3. Области применения файловых систем. Требования к базам данных со стороны информационных систем: согласованность данных, языки запросов, восстановление согласованного состояния после сбоев, реальный режим мультидоступа.

4. Основные функции СУБД, типовая организация СУБД.

5. Дореляционные модели данных

6. Основные черты модели данных SQL

7. Типы данных, наследование типов в SQL

8. Основные черты модели данных ODMG

9. Типы данных, наследование типов в модели данных ODMG

10. Основные черты истинно реляционной модели данных

11. Типы данных, наследование типов в истинно реляционной модели данных

12. Общие понятия реляционного подхода к организации БД. Основные концепции и термины.

13. Фундаментальные свойства отношений.

14. Реляционная модель данных: общее понятие и составные части.

15. Реляционная алгебра Кодда.

16. Алгебра A.

17. Полнота алгебры A.

18. Избыточность алгебры A.

19. Реляционное исчисление кортежей.

20. Реляционное исчисление доменов.

21. Функциональные зависимости, замыкание множества функциональных зависимостей, аксиомы Армстронга, замыкание множества атрибутов. Минимальное покрытие множества функциональных зависимостей.

22. Декомпозиция без потерь и функциональные зависимости, теорема Хита.

23. Проектирование реляционных баз данных с использованием нормализации: первая, вторая и третья нормальные формы.

24. Проектирование реляционных баз данных с использованием нормализации: теорема Риссонена, нормальная форма Бойса-Кодда.

25. Многозначные зависимости, теорема Фейджина, четвертая нормальная форма.

26. Зависимости проекции-соединения, пятая нормальная форма.

27. Семантические модели данных.

28. Семантическая модель Entity-Relationship (Сущность-Связи).

29. Получение реляционной схемы из ER-диаграммы.

30. Диаграммы классов языка UML.

31. Язык объектных ограничений OCL.

32. Основные цели System R и их связь с архитектурой системы.

33. Организация внешней памяти в базах данных System R, B-деревья.

34. Интерфейс ядра System R - RSS.

35. ACID-транзакции. Средства СУБД для поддержки свойств атомарности, согласованности, изолированности и постоянства хранения.

36. Сериализация транзакций, виды конфликтов транзакций и порождаемые ими феномены поведения транзакций. Двухфазный протокол синхронизационных блокировок.

37. Гранулированные и предикатные блокировки.

38. Синхронизационные тупики, способы их обнаружения и разрушения.

39. Сериализация транзакций на основе временных меток.

40. Версионный вариант алгоритма временных меток

41. Версионный вариант двухфазного протокола синхронизационных блокировок

42. Версионно-блокировочный протокол сериализации транзакций для поддержки только читающих транзакций.

43. Ситуации, требующие восстановления базы данных. Индивидуальные откаты транзакций. Понятие журнала.

44. Управление буферами основной памяти.

45. Физическая синхронизация.

46. Протокол Write Ahead Log.

47. Физически согласованное состояние базы данных. Восстановление базы данных после мягкого сбоя.

48. Способы восстановления физически согласованного состояния.

1. Архивация базы данных и журнала. Восстановление базы данных после жесткого сбоя.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Физические основы построения ЭВМ**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Список контрольных вопросов по курсу:**

1 Приведите примеры (не менее трех) полупроводниковых материалов, используемых при создании современных интегральных микросхем.

2 Перечислите основные технологические операции (не менее пяти) при изготовлении интегральных микросхем.

3 Поясните суть метода фотолитографии и его роль в процессе изготовлении интегральных микросхем.

4 Запишите уравнение де Бройля и поясните его смысл.

5 Поясните смысл понятия энергетических зон в твердых телах на примере проводников, полупроводников и диэлектриков.

6 Что является носителями электрического заряда в полупроводниках и каковы их основные характеристики?

7 Для чего и каким образом осуществляется легирование полупроводниковых материалов?

8 Поясните причины возникновения дрейфового и диффузионный токов в p-n переходе и запишите поясняющие формулы.

9 На примере p-n-перехода покажите направления дрейфового и диффузионного токов.

10 Нарисуйте вольт-амперную характеристику p-n-перехода и поясните ее основные особенности.

11 Поясните, какова причина появления и от чего зависит величина барьерной и диффузионной емкости p-n-перехода.

12 Изобразите схематически структуру биполярного транзистора и поясните принцип его работы.

13 Изобразите схематически структуру полевого транзистора и поясните принцип его работы.

14 Изобразите схематически планарную структура биполярного транзистора и поясните, как она создается.

15 Изобразите схематически планарную структуру полевого транзистора и поясните, как она создается.

16 Изобразите планарную схему МОП (или MOS)-структуры и поясните ее суть.

17 Приведите пример схемы из двух транзисторов, изготовленных в соответствии с КМОП(CMOS) технологией.

18 Чем обусловлены преимущества цифрового подхода перед аналоговым при обработке, записи, хранении, передаче и регистрации информации.

19 Чем ограничены минимальные затраты на запись, хранение, обработку бита информации?

20 Изобразите схематически реализации транзисторного ключа на биполярном транзисторе и с использованием КМОП технологии.

21 Поясните термин “динамическая потребляемая мощность”. Выведите выражения для зависимости динамической потребляемой мощности от напряжения питания, топологического размера элемента, частоты переключения.

22 Схематически проиллюстрируйте типы выходов цифровых микросхем. Какие проблемы возникают при объединении выводов микросхем, в том числе, при шинной организации связей?

23 Что такое архитектура Фон-Неймана? Изобразите поясняющую схему.

24 Что такое АЛУ и УУ, какова их роль? Зачем нужен задающий генератор при их взаимодействии.

25 Назовите основные этапы цикла обработки команды процессором. Что такое конвейер и как он работает?

26 Какие вы знаете методы повышения производительности процессора без повышения тактовой частоты?

27 Что такое синхронный и асинхронный режим работы шины? Что такое мультиплексирование шин?

28 Чем шина PCI отличается от шины PCI-Express?

29 Что такое 3D-транзистор? В чем его преимущества? Каковы особенности технологического процесса?

30 Что такое триггер? Приведите пример схемы триггера на транзисторах.

31 Общая схема организации памяти. Основные управляющие сигналы для доступа к ячейке памяти.

32 Устройство ячейки динамической памяти.

33 Нарисуйте схему и поясните принцип работы транзистора с плавающим затвором.

34 Изобразите логические блок-схемы взаимодействия с внешними устройствами: программного, по прерыванием, с использованием прямого доступа к памяти.

35 Что такое последовательные и параллельные шины/интерфейсы? Перечислите их преимущества и недостатки.

36 Устройство и особенности кабеля и разъемов USB 2.0. Конструктивная и логическая схема организации интерфейса USB.

37 Цикл передачи данных по интерфейсу USB 2.0.

38 Назовите основные характеристики интерфейса Thunderbolt.

39 Какие материалы называются ферромагнитными?

40 Что такое «петля гистерезиса»? Поясните ее основные характеристики.

41 Какой физические эффекты используются при записи и считывании данных на магнитных носителях?

42 Параллельный и перпендикулярный способы записи на магнитных дисках. В чем состоят преимущества перпендикулярной записи?

43 Поясните, что такое магниторезистивный эффект.

44 Охарактеризуйте плотность записи данных на современных жестких дисках.

45 Чем когерентное оптическое излучение отличается от некогерентного?

46 Сравните параметры систем записи и считывании CD, DVD и Blu-ray дисков.

47 Чем обусловлены ограничения плотности записи с использованием лазерного излучения?

48 Нарисуйте схему и на ее основе поясните принцип записи голографического изображения.

49 Что такое спектральное представление сигнала? Его связь с временным представлением.

50 Теорема Котельникова.

51 Опишите принцип работы АЦП. Как сигнал, изменяющийся во времени, записывается в память компьютера?

52 Изобразите схематически устройство параллельного АЦП. Объясните как оно работает.

53 Опишите принцип работы ЦАП. Как превратить кодовую числовую последовательность в аналоговый сигнал (ток, напряжение или заряд)?

54 Поясните суть явления внутреннего фотоэффекта и как оно используется при детектировании оптических сигналов

55 Изобразите схематически устройство и опишите принцип работы трехфазного регистра ПЗС (CCD) регистра.

56 Изобразите схематически, как работает матричный ПЗС (CCD) детектор.

57 Изобразите схематически, как работает матричный КМОП (CMOS) сенсор/детектор.

58 Поясните, какие физические явления используются при нанесении изображения лазерным принтером.

59 Поясните, какие технологии применяются для формирования капель красителя в лазерных принтерах.

60 Что из себя представляет и как работает 3D принтер?

61 Методы формирования изображения на экране монитора. Учет принципа Ферма.

62 Что такое поляризация электромагнитных волн? Как получить поляризованный свет? Что можно использовать в качестве поляризатора?

63 Изобразите схематически устройства ячейки TFT-LCD монитора и объясните принцип работы.

64 Перечислите преимущества OLED технологии по сравнению с TFT-LCD.

65 Как работает электронная бумага (E-INK)?

66 Перечислите не менее трех подходов к формированию 3D изображения.

67 Перечислите, какие проводные линии связи используются для передачи данных.

68 Перечислите, какие беспроводные линии связи используются для передачи данных.

69 Каковы преимущества и недостатки проводных и беспроводных линий связи?

70 Охарактеризуйте дальность и максимальную скорость передачи данных с использованием современных линий связи: телефонная линия/ коаксиальный кабель/ витая пара/ оптическое волокно/ радиорелейная линия/ спутниковый канал/ WiFi/ WiMAX/ bluetooth

71 Для чего используется модуляция сигнала?

72 В чем состоят отличия одномодовых оптических волокон от многомодовых?

73 Какие источники света используются для передачи данных по оптическим волокнам?

74 Чем обусловлены основные ограничения скорости передачи данных по оптическим волокнам?

75 Сколько транзисторов входит в состав современных микропроцессоров?

76 Перечислите основные направления, в которых возможно развитие/совершенствование компьютеров

77 Какие ключевые физические задачи необходимо решить при создании квантового компьютера?

**Типовые задачи для проверки знаний:**

1 Найти длину волны, соответствующую фотону и электрону с энергией 1 эВ.

2 Оценить концентрацию свободных носителей заряда в беспримесном кремнии (***Si***) при температуре ***T***=300K. Во сколько раз изменится концентрация при увеличении температуры на 40К? (Замечание: Изменения величин и с температурой считать несущественными).

3 Для ***p-n***-перехода в кремнии (***Si***) (***ni***=3∙1019см-3, ***ND***=5х1019см-3, ***NA***=1015см-3) найти высоту потенциального барьера **Δ*E0***. Результат привести в эВ и в Дж. Пояснить смысл использования внесистемной единицы энергии.

4 Получить выражения, описывающие форму сигнала на сопротивлении R и конденсаторе C после прохождения прямоугольного импульса длительностью Т и амплитудой U0 через RC-цепь. Рассмотреть случаи RC>>T, RC~T, RC<<T.

5 Получите оценку максимальной емкости памяти компакт-диска с внешним и внутренним диаметрами рабочей области 118 мм и 35 мм соответственно. Для записи/чтения используется полупроводниковый лазер с длиной волны 780 нм.

6 Сотовый телефон работает от аккумулятора емкостью ***E***=5000 мА\*час и напряжением ***U***=3,7 вольта. В режиме разговора телефон работает на частоте ***f***=900 МГц, средняя емкость цифровой схемы телефона составляет ***C***=10 нФ, а на антенну выдается сигнал мощностью ***P***=3 Вт. Когда телефон не используется, находясь в режиме ожидания, обработка сигналов не выполняется. При этом, телефон все же потребляет ***I0***=40 мА остаточного тока. Рассчитайте время, на которое хватит полностью заряженного аккумулятора телефона, для случаев: (а) телефон включен, но не используется; (б) телефон используется непрерывно.

7 Получите выражение для формы спектра ***S***(**ω**) прямоугольного импульса длительностью ***T*** и амплитудой ***U***. Найдите произведение длительности этого импульса и ширины его спектра. Ширину спектра считать равной расстоянию между ближайшими к центральному максимуму частотами **ω** при которых ***S***(**ω**) обращается в 0.

8 Цепь переменного тока состоит из последовательно включенных источника синусоидального напряжения ***E***(***t***)=***E*0**sin(***ωt***), конденсатора емкостью С и резистора сопротивлением ***R***. Найдите зависимость амплитуды напряжения на конденсаторе от частоты **ω** и постройте ее график.

9 Оценить минимальный радиус ***R*** изгиба оптического волокна, при котором излучение, распространяющееся вдоль оси, не будет покидать сердцевину волокна. Показатели преломления сердцевины и оболочки ***nc*** и ***no***, соответственно. Считаем, что радиус сердцевины ***r<<R.*** Значения ***nc***=1,479 и ***no***=1,474.

10 Оцените максимальное количество радиостанций, вещающих в FM-диапазоне 87-108 МГц в режиме стерео. Полагаем, что полоса частот каждого канала соответствует диапазону звуковых частот 20 Hz-20kHz, воспринимаемому человеческим ухом. Для исключения наложения сигналов от соседних каналов, частотные каналы разделены промежутками 10 кГц.

11 Оцените площадь и линейный размер области, соответствующей записи одного бита на пластине магнитного диска диаметром 3,5 дюйма и емкостью 1 ТБ.

12 Составьте схему АЛУ устройства для выполнения операции умножения целых чисел.

13 Оцените максимальную длину последовательной компьютерной шины для скоростей передачи 1 Гбит/с и 1 Мбит/с.

**Типовые темы для рефератов и презентаций:**

1 Современные полупроводниковые материалы

2 Полупроводниковый p-n-переход. Физические процессы и их влияние на хар-ки полупроводниковых устройств

3 Основные технологические операции при изготовлении интегральных микросхем

4 UV- и EUV- фотолитография

5 Полупроводниковые источники света. Светодиоды

6 Полупроводниковые фотоприемники

7 Биполярные и полевые транзисторы. Принципы работы, основные параметры, сходства и отличия

8 Перспективные схемы транзисторов

9 АЦП и ЦАП. Принципы работы, основные характеристики

10 Устройство различных типов памяти: статической (регистры), динамической (ОЗУ), FLASH (ППЗУ)

11 FPGA (ПЛИС)  
12 Перспективные типы оперативной памяти (MRAM магниторезистивная память, оптическая, нанотехнологии, и т.д.)

13 ПЗС и КМОП детекторы/сенсоры

14 Полупроводниковые лазеры, лазеры на гетероструктурах

15 Твердотельные устройства памяти

16 Устройства записи на магнитных носителях. Продольная и перпендикулярная запись

17 Магнитооптические устройства записи и хранения

18 Устройства записи и хранения на магнитных носителях. Перспективные устройства и материалы.

19 Оптическая память. Предельная плотность записи информации в оптике

20 Трехмерная оптическая и голографическая память

21 TFT-LCD-дисплеи: физические процессы, формирование изображения.

22 Перспективные типы дисплеев (OLED, FED и др.)

23 3D-дисплеи

24 Электронная бумага. Физические процессы работы, формирование изображения

25 3D принтеры

26 Беспроводные линии связи (WiFi, WiMax, 3/4G, LTE: технологии и физические процессы

27 Связь по телефонным линиям. Виды модуляции сигналов

28 Оптические волокна и волоконно-оптические кабели

29 Волоконно-оптические линии связи

30 Органическая электроника

31 Нанотрубки. Применение в электронике

32 Графен. Применение в электронике

33 Спинтроника

34 Оптический компьютер

35 Нейрокомпьютеры

36 Современные элементы питания и аккумуляторы

37 МЭМС и НЭМС технологии

38 Принцип работы систем спутниковой навигации

39 Микроскопы с разрешение выше оптического

40 Особенности процессоров для мобильных устройств. Специализированные процессоры

41 Интерфейсы USB, Thunderbolt

42 Квантовые каналы передачи данных

43 Квантовый компьютер

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

**Список вопросов к зачету:**

1 Поколения компьютеров и их элементная база. Роль полупроводниковых материалов и устройств в элементной базе современных компьютеров.

2 Преимущества интегральных микросхем перед дискретными компонентами. Экспоненциальное развитие и закон Мура.

3 Основные технологические процессы при производстве современных сверхбольших интегральных микросхем. Фотолитография. Соединение элементов

4 Степень интеграции, технологические нормы при производстве микросхем. Воспроизводимость параметров и минимальный топологический размер.

5 Электроны. Уравнение де Бройля. Соотношение неопределенностей.

6 Волновая функция. Волновая функция электрона в бесконечной потенциальной яме. Спектр электронных состояний атома водорода и многоэлектронных атомов.

7 Спектр электронных состояний в атомах, молекулах и кристаллах. Типы химических связей в твердом теле. Понятие о зонной структуре твердых тел. Принципы разделения веществ на проводники (металлы), полупроводники и изоляторы (диэлектрики).

8 Энергетические зоны в полупроводниках. Энергия (уровень) Ферми. Положение уровня Ферми в электрически нейтральном полупроводнике.

9 Электроны и дырки. Полупроводники n- и p-типа. Технологии легирования полупроводников.

10 Движение свободных носителей заряда в полупроводнике, дрейфовый и диффузионный ток.

11 Электронно-дырочный(**p-n**) переход, распределение поля и потенциала, потенциальный барьер.

12 Вольт-амперная характеристика **p-n**-перехода. Дифференциальное сопротивление.

13 Барьерная и диффузионная емкость **p-n**-перехода.

14 Полупроводниковые диоды. Виды полупроводниковых диодов. Быстродействие полупроводниковых диодов.

15 Контакты металл - полупроводник.Диоды Шоттки. Омические контакты.

16 Понятие о прямозонных и непрямозонных полупроводниках. Полупроводниковые приемники и источники света.

17 Взаимодействие двух близкорасположенных электронно-дырочных переходов. Биполярный транзистор. Изготовление транзисторов с использованием планарной технологии.

18 Режимы работы биполярного транзистора. Основные схемы включения. Усиление тока и напряжения.

19 Режимы работы биполярного транзистора. Ключевой режим. Быстродействие.

20 Полевые транзисторы. Виды полевых транзисторов. МОП (MOS) структуры и КМОП (CMOS) технология.

22 Основные схемы включения и быстродействие полевых транзисторов.

23 Транзисторы в современном компьютере. Перспективные схемы транзисторов.

24 Свойства аналоговой и цифровой информации. Физическое представление информации в ЭВМ. Двоичный код. Примеры реализации элементарных логических функций.

25 Ключевой режим работы коммутирующего элемента. «Высокое» и «низкое»состояния логических схем. Позитивная и негативная логики.

26 Основные характеристики логических элементов. Средняя статическая потребляемая мощность, динамическая потребляемая мощность, время задержки распространения сигнала, коэффициент разветвления по выходу.

27 Архитектура фон Неймана и обобщенная структура системного блока: микропроцессор, память, шина, внешние устройства.

28 Основные характеристики микропроцессора. Цикл микропроцессора и его фазы. Приемы и методы повышения производительности микропроцессора.

29 Взаимодействие микропроцессора и оперативного запоминающего устройства. Способы обмена информацией между микропроцессором и внешними устройствами: синхронный, асинхронный и полусинхронный. Режимы работы процессора: прерывание, прямой доступ к памяти.

30 Компьютерные шины, классификация и основные характеристики (ISA, PCI, PCI-E, HyperTransport, Ring Interconnect). Мультиплексирование. Методы повышения пропускной способности шин.

31 Специализированные микропроцессоры. Мультипроцессорные и многоядерные конфигурации. Процессор Эльбрус.

32 Конденсатор и триггер как простейшие ячейки памяти. Энергозависимая и энергонезависимая память.

33 Энергозависимая и энергонезависимая память. Статическое оперативное запоминающее устройство (SRAM). Принципы работы. Применение SRAM в ЭВМ.

34 Общая организация памяти. Характеристики и виды памяти: стоимость, емкость,быстродействие, потребляемая мощность, возможность доступа.

35 Динамическое оперативное запоминающее устройство (DRAM). Физические процессы, происходящие при записи/ считывании/ регенерации. Применение DRAM в ЭВМ.

36 Постоянное запоминающее устройство (ПЗУ) и Flash-память. Элементы на основе структур с плавающим затвором. Стирание информации, туннельный эффект. Применение Flash-памяти в ЭВМ.

37 Роль и место различных типов памяти в ЭВМ. Перспективные типы компьютерной памяти.

38 Функции интерфейса ввода-вывода. Информационная, электрическая и конструктивная совместимость интерфейсов. Дуплексная и полудуплексная связь. Асинхронная и синхронная связь.

39 Устройство типичного интерфейса. Функциональная и управляющая части интерфейса. Методы доступа FIFO и LIFO. Примеры компьютерных интерфейсов и их характеристики и назначение.

40 Интерфейс USB 2.0 Конструктивное, программное, логическое устройство. Цикл работы шины USB 2.0. Thunderbolt - принципы построения, характеристики, современное состояние.

41 Устройства внешней памяти на основе магнитных материалов. Магнетизм. Магнитные материалы: диамагнетики, парамагнетики, ферромагнетики. Температура Кюри. Магнитная память в природе.

42 Намагниченность. Закон электромагнитной индукции. Кривая намагниченности ферромагнетиков. Магнитомягкие и магнитотвердые ферромагнетики. Доменная структура.

43 История магнитной записи.Принципы записи и считывания информации на магнитных носителях.Запись цифровой информации. Типы магнитных носителей.

44 Жесткий диск: запись и считывание. Продольная и поперечная запись. Скорость доступа к записанной информации.

45 Записывающие и считывающие головки в современных жестких дисках. Использование магниторезистивных эффектов. Предельная плотность записи.

46 Современные магнитные носители и перспективные технологии в устройствах магнитной памяти. Твердотельные устройства внешней памяти и их перспективы как альтернативы устройств магнитной памяти.

47 Взаимодействие света со средой. Электромагнитные волны и свет. Уравнения Максвелла. Кванты света. Когерентное излучение.

Понятие о лазерной генерации. Фокусировка лазерного излучения.

48 Современные источники когерентного света. Полупроводниковые лазеры.

Технологии и устройства оптической памяти. Компакт диск, R,RW CD, DVD, HD-DVD, Blu-ray и HD-DVD технологии.

49 Предельная плотность записи информации в оптике. Магнитная запись с использованием оптических явлений. Магнитооптика. Термомагнитная запись.

50 Перспективные технологии оптической памяти. Трехмерная оптическая память: фоторефрактивные и фотохромные материалы, голография.

51 Принципы отображения и восприятия визуальной информации. Алфавитно-цифровые и графические (аналоговые) мониторы. Методы формирования изображения.

52 Электронно-лучевая трубка (ЭЛТ). Физические процессы в ЭЛТ: термоэлектронная эмиссия, отклонение, фокусировка, люминесценция. Формирование изображения в ЭЛТ. Отображение цвета.

53 TFT-LCD мониторы: жидкокристаллические (LCD) дисплеи. Устройство и принцип работы ячейки TFT-LCD монитора.

54 OLED дисплеи - принципы работы. Преимущества по отношению к TFT-LCD мониторам. Электронная бумага. Перспективные типы дисплеев.

55 Ввод и вывод цифровой и аналоговой информации. Понятие о цифровом методе хранения и передачи аналоговой информации. Понятие о спектральном представлении сигналов, преобразование Фурье. Теорема Котельникова (Найквиста-Шеннона), эффект наложения частот.

56 Аналого-цифровые преобразователи, примеры схем реализации. Цифро-аналоговое преобразователи, пример схемы реализации.

57 Ввод оптического изображения в ЭВМ: приборы с зарядовой связью (ПЗС). Принцип действия ПЗС-камеры. CMOS сенсоры.

58 Принципы отображения информации на твердых носителях. Матричные, струйные, лазерные принтеры. Цветная печать. 3D-принтеры.

59 Основные современные линии связи для передачи данных. Физическая среда, скорость и структура канала передачи данных.

60 Модуляция сигнала. Кодирование информации: амплитудная, фазовая, частотная и другие типы модуляции. Модем.

61 Проводные и кабельные линии связи. Телеграфное уравнение. Волновое сопротивление. Двухпроводная линия. Коаксиальный кабель и витая пара.

62 Беспроводные линии связи. Электромагнитные волны и свет. Наземная (радиорелейная) связь. Спутниковая связь. Каналы ближней радиосвязи.

63 Оптическое волокно. Ввод излучения в оптическое волокно. Критическая длина волны. Распространение света по волокну.

64 Современные оптические волокна. Потери излучения при распространении и окна прозрачности современных оптических волокон. Моды оптического волокна. Одномодовые и многомодовые оптические волокна. Дисперсия оптических волокон и способы ее подавления.

65 Волоконно-оптические линии связи и их структура. Ближние и магистральные линии связи. Предельная скорость передачи данных и способы ее увеличения.

66 Приемники и излучатели света для волоконно-оптических линий связи: свето- и фотодиоды, полупроводниковые и волоконные лазеры и усилители.

67 Технологические нормы и основные технологические процессы и при производстве интегральных микросхем. Возможности развития и физические ограничения. Предельные размеры, быстродействие и энергопотребление.

68 Биты и кубиты. Когерентность состояний. Вычисления в классических и квантовых компьютерах. Разрушение когерентности как источник ошибок при квантовых вычислениях, возможность их коррекции.

69 Как построить квантовый компьютер? Ионные ловушки, ЯМР, поверхностные наноструктуры, сверхпроводящие структуры. Перспективы реализации квантовых компьютеров.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Основы кибернетики**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №1**

1. Определение элементарной конъюнкции и элементарной дизъюнкции, ДНФ и КНФ, их «геометрическая» интерпретация. Критерий единственности ДНФ.
2. Определение пучка, регулярной точки и регулярной грани ФАЛ; ДНФ  и критерий вхождения в неё простых импликант.
3. Определение монотонной ФАЛ и её «нижних» единиц. Утверждение о сокращённой ДНФ монотонной ФАЛ, особенности её тупиковых ДНФ.
4. Определение кратчайшей ДНФ и её связь с тупиковыми ДНФ, определение функции Шеннона  для длины кратчайших ДНФ, реализующих ФАЛ от  БП. Поведение функции Шеннона  (идея получения верхней оценки и пример ФАЛ, на которой достигается нижняя оценка).
5. Определение функции Шеннона  для числа минимальных ДНФ у ФАЛ от  БП, её нижняя оценка и пример ФАЛ, на которой эта оценка достигается.
6. Построить сокращённую ДНФ ФАЛ , столбец значений которой имеет вид .
7. На основе ФАЛ покрытия для редуцированной таблицы Квайна построить ядро, ДНФ Квайна, ДНФ сумма тупиковых и все тупиковые ДНФ функции , заданной сокращённой ДНФ:

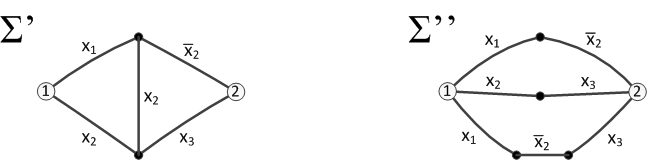
.

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №2**

1. Определение структуры СФЭ в базисе  как помеченного графа специального вида, определение ФАЛ, реализуемой в вершине СФЭ, и системы ФАЛ, реализуемой этой СФЭ. Определение приведённой СФЭ, утверждение о соотношениях между рангом, сложностью и глубиной приведённой СФЭ с 1 выходом в стандартном базисе.
2. Определение эквивалентных формул. Верхняя оценка числа попарно не эквивалентных формул от БП , имеющих сложность не больше, чем , в стандартном базисе, а также в базисе .
3. Определение подсхемы заданной КС из неориентированных контактов с неразделёнными полюсами, принцип эквивалентной замены. Определение тождества для КС из неориентированных контактов с неразделёнными полюсами и его подстановки; описание указанной подстановки одного из основных тождеств, результатом которой является тождество для -схем, моделирующее формульное тождество .
4. Суммарное цикломатическое число КС и его изменение при ЭП КС на базе различных основных тождеств. Идея и основные этапы доказательства утверждения об отсутствии КПСТ в классе всех КС.
5. Тождества перехода от одного базиса к другому; утверждение о моделировании ЭП формул в различных базисах и идея его доказательства.
6. С помощью расширенной системы основных тождеств построить ЭП для формул  и :

,

1. С помощью системы основных тождеств построить ЭП для КС и , указав без каких тождеств вида при этом нельзя обойтись. Упрощённый вариант задачи: преобразовать КС описанным способом в КС, все контакты которой лежат на цепях, соединяющих различные полюса и не имеющих общих внутренних вершин.



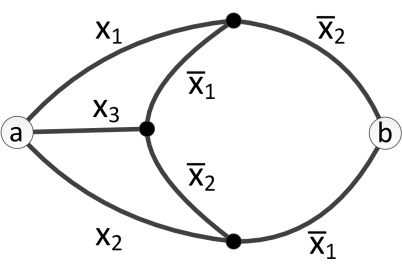
**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №3**

1. Определение сложности  системы ФАЛ  в классе СФЭ над базисом  и формулировка утверждения о простейшей нижней оценке этой сложности. Универсальная система ФАЛ  порядка  и оценка указанного вида для её сложности.
2. Определение функции Шеннона  и её верхняя оценка, получаемая методом Шеннона, с описанием структуры и указанием сложности схемы, построенной для произвольной ФАЛ .
3. Операция присоединения одной вершины СФЭ к другой вершине этой схемы и её свойства, определение строго приведённой СФЭ. Формулировка утверждения о верхней оценке сложности  и идея его доказательства.
4. Определение ‑регулярного множества наборов куба , , от БП , его свойства и задание с помощью системы ФАЛ от БП . Построить 2-регулярное подмножество куба  от БП , на котором ФАЛ ,  и  совпадают с одной из ФАЛ .
5. С помощью моделирования совершенной ДНФ на основе контактного дерева построить –КС, реализующую ФАЛ , где , а затем получить из этой ККС инверсную схему.
6. Построить минимальную –КС для ФАЛ :

.

1. С помощью метода каскадов, последовательно разлагая реализуемые ФАЛ по , построить –КС ∑ для системы ФАЛ  и СФЭ  для ФАЛ , где , .

**ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ №4**

1. Определение ДУМ; описание ДУМ порядка  и ранга , связанного с разбиением куба  на  непустых подмножеств, его мощность. Построить таблицу значений ФАЛ, образующих ДУМ порядка 3 и ранга 4, связанное с каким-либо разбиением куба  на 4 подмножества мощности 1, 2, 2, 3.
2. Нижняя мощностная оценка функции Шеннона  и те соотношения, из которых она выводится (определение , верхняя оценка числа схем и её сравнение с числом ФАЛ).
3. Утверждение о сложности СФЭ, получаемых асимптотически наилучшим методом синтеза, и вытекающая из него верхняя оценка соответствующей функции Шеннона.
4. Описание разложения (представления) ФАЛ, на котором основано доказательство утверждения из п. 3, и структура соответствующей схемы с указанием основного по сложности блока.
5. Определение диагностического теста отделимой по столбцам матрицы , , понятие тупикового и минимального диагностического теста. Формулировка утверждения об оценках длины диагностического теста для почти всех матриц указанного вида, где  и .
6. Применяя методы самокоррекции однородных подсхем к π‑схеме, которая моделирует формулу, подобную формуле , построить эквивалентную ей ‑самокорректирующуюся КС сложности не больше, чем 19.
7. В контактной схеме возможна одна из следующих четырёх неисправностей:
   1. замыкание контактов ;
   2. замыкание контактов ;
   3. обрыв контактов ;
   4. обрыв контактов .

Построить все тупиковые диагностические тесты.

1. Установить асимптотику сложности реализации схемами из функциональных элементов самой сложной из тех ФАЛ , , которые на любой паре противоположенных наборов принимают одинаковые значения.
2. По КНФ , являющейся входом для языка ВЫПОЛНИМОСТЬ построить граф G и число k, являющиеся входом для языка КЛИКА.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

Экзамен проводится как в устной форме, так и в письменной. Для студентов, не сдавших экзамен, учебным планом предусматривается переэкзаменовка.

**Вопросы к экзамену**

1. Представление функций алгебры логики (ФАЛ) дизъюнктивными нормальными формами (ДНФ) и его «геометрическая» интерпретация. Совершенная ДНФ и критерий единственности ДНФ.

2. Сокращённая ДНФ и способы её построения.

3. Тупиковая ДНФ, ядро и ДНФ пересечение тупиковых. ДНФ Квайна, критерий вхождения простых импликант в тупиковые ДНФ и его локальность.

4. Особенности ДНФ линейных и монотонных ФАЛ. Функция покрытия, таблица Квайна и построение всех тупиковых ДНФ.

5. Градиентный алгоритм и оценка длины градиентного покрытия, лемма о «протыкающих» наборах. Использование градиентного алгоритма для построения ДНФ.

6. Задача минимизации ДНФ. Поведение функции Шеннона и оценки типичных значений для ранга и длины ДНФ.

7. Алгоритмические трудности минимизации ДНФ и оценки максимальных значений некоторых связанных с ней параметров. Теорема Ю.И. Журавлёва о ДНФ сумма минимальных.

8. Формулы, способы их задания и эквивалентные преобразования. Оптимизация подобных формул по глубине.

9. Эквивалентные преобразования формул с помощью тождеств. Полнота системы основных тождеств для эквивалентных преобразований формул базиса *Б*0.

10. Задание формул графами, схемы из функциональных элементов (СФЭ). Оценка числа формул и СФЭ в базисе *Б*0={&, ∨, ¬}.

11. Эквивалентные преобразования СФЭ и моделирование с их помощью формульных преобразований. Моделирование эквивалентных преобразований формул и схем в различных базисах, теорема перехода.

12. Контактные схемы (КС) и π-схемы, оценка их числа. Особенности функционирования многополюсных КС.

13. Эквивалентные преобразования КС. Основные тождества, вывод вспомогательных и обобщённых тождеств.

14. Полнота системы основных тождеств. Отсутствие конечной полной системы тождеств в классе всех КС.

15. Некоторые модификации и частные случаи основных классов схем (каскадные КС и BDD, КМОП-схемы, вычисляющие программы и др.)

16. Реализация автоматных функций схемами из функциональных элементов и элементов задержки, схемы с «мгновенными» обратными связями.

17. Задача синтеза. Методы синтеза схем на основе ДНФ и связанные с ними верхние оценки сложности функций.

18. Нижние оценки сложности ФАЛ, реализация некоторых ФАЛ и минимальность некоторых схем.

19. Разделительные КС и лемма Шеннона. Метод каскадов для КС и СФЭ, примеры его применения. Метод Шеннона.

20. Нижние мощностные оценки функций Шеннона.

21. Дизъюнктивно-универсальные множества ФАЛ. Асимптотически наилучший метод О.Б. Лупанова для синтеза СФЭ в базисе *Б*0.

22. Регулярные разбиения единичного куба и моделирование ФАЛ переменными. Методы синтеза формул в базисе *Б*0, поведение функции Шеннона для глубины ФАЛ.

23. Асимптотически наилучший метод синтеза КС. Синтез схем для некоторых дешифраторов и мультиплексоров.

24. Задача синтеза схем для ФАЛ из специальных классов. Асимптотически оптимальные методы синтеза СФЭ и КС для ФАЛ из некоторых классов.

25. Схемы на КМОП-транзисторах и реализация ими простейших функций. Задачи логического и топологического синтеза СБИС, основные этапы и методы их решения.

26. Самокорректирующиеся КС и методы их построения. Асимптотически наилучший метод синтеза КС, корректирующих 1 обрыв (1 замыкание).

27. Задача контроля схем и тесты для таблиц. Построение всех тупиковых тестов, оценки длины диагностического теста.

28. Полиномиальная сводимость языков. Классы P и NP, NP-полнота, формулировка теоремы Кука. Примеры NP-полных проблем.

29. Доказательство теоремы Кука.

**Типовые задачи к экзамену**

**Задачи на ДНФ**

1. По заданной ФАЛ построить её сокращённую ДНФ, ДНФ Квайна, ДНФ сумма тупиковых, все тупиковые ДНФ.

Задачи на эквивалентные преобразования и структурное моделирование

1. По заданным эквивалентным формулам или КС построить эквивалентное преобразование, переводящее их друг в друга с помощью основных тождеств.

2. По заданной формуле построить подобную ей формулу минимальной глубины.

3. По заданной формуле с поднятыми отрицаниями построить моделирующую её π схему и обратно.

4. По данной каскадной КС построить инверсную каскадную КС.

**Задачи на синтез схем**

1. По заданной ФАЛ с помощью простейших методов, метода каскадов или метода Шеннона построить реализующую её СФЭ или КС.

2. Оценить сверху или снизу сложность конкретной ФАЛ или сложность самой сложной ФАЛ из заданного множества в заданном классе схем.

**Задачи на самокоррекцию и тесты**

1. По заданной КС построить эквивалентную ей самокорректирующуюся КС.

2. По заданной таблице или КС и списку её неисправностей построить все тупиковые проверяющие (диагностические) тесты.

**Задачи на сложность алгоритмов**

1. Построить полиномиальное сведение одного языка к другому.

По результатам тестов и контрольных работ с учетом посещаемости студентов, их работы на лекциях и семинарах, а также самостоятельной работы выставляется предварительная оценка.

Для студентов, имеющих предварительную оценку «5», экзамен проводится в форме собеседования по программе курса на определения, формулировки утверждений и идеи их доказательства, методы решения задач. Для студентов, имеющих предварительную оценку «2», экзамен представляет собой письменный тест-контрольную.

Все остальные студенты (с предварительной оценкой «3-», «3» и «4») получают билет с двумя вопросами и одной задачей и после 15-20 минут подготовки отвечают на него сначала на уровне определений, формулировок утверждений и идей их доказательства, а также методов решения задач. Затем студент, по усмотрению экзаменатора, должен раскрыть те или иные детали доказательства утверждений из вопросов билета по конспектам или иным источникам, а также полностью или частично решить задачу билета в течение выделенного специально для этого времени. Студенты, набравшие не менее 80% от суммы баллов по задачам тестов-контрольных соответствующего раздела, от решения билетной задачи данного типа освобождаются. Последний этап экзамена представляет собой описанное выше собеседование по другим вопросам или задачам программы.

Студент, который имеет предварительную оценку «3» или «4» и не претендует на более высокую итоговую оценку, сдаёт экзамен, как правило, по упрощённой процедуре (в форме собеседования по билету и программе без предварительной подготовки) с целью подтверждения этой оценки.

**Экзаменационный билет** состоит из двух вопросов и задачи, например

1. Эквивалентные преобразования СФЭ и моделирование с их помощью формульных преобразований. Моделирование эквивалентных преобразований формул и схем в различных базисах, теорема перехода.

2. Дизъюнктивно-универсальные множества ФАЛ. Асимптотически наилучший метод О.Б. Лупанова для синтеза СФЭ в базисе *Б*0.

3. Построить сокращённую ДНФ ФАЛ , столбец значений которой имеет вид .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Компьютерная графика**

Курс предусматривают выполнение практических заданий, материал к которым выложен на сайте курсов лаборатории компьютерной графики и мультимедиа ВМК МГУ (<https://courses.graphics.cs.msu.ru/>). Задания сдаются на том же сайте посредством загрузки в систему moodle. Проверка заданий осуществляется удаленно персонифицировано для каждого из учащихся. Разбор общих вопросов, типичных ошибок и рекомендации по повышению эффективности алгоритмов и программ дается на семинарских занятиях или лекциях при отсутствии семинарских занятий.

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

**Примерные практические домашние задания.**

1. Реализовать визуализацию (рендеринг) 3D сцены при помощи алгоритма трассировки лучей.
2. Реализовать анимацию и визуализацию 3D сцены содержащей воду (одежду или другой объект физической симуляции) при помощи алгоритма растеризации и современных средств программирования графического конвейера (OpenGL3 или Vulkan).

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Примеры вопросов на экзамене.**

1. Какой промежуток натуральныхчисел может быть представлен в в типе half float (16 бит) (IEEE 754-2008)без ошибки?
2. Дайте определение цифровому цветному изображению.
3. Дайте определение альфа-канала и поясните его назначение.
4. Что такое G-buffer? Приведите пример данных, которые он может хранить.
5. В компьютерной графике используются матрицы 4x4 а не 3x3. Приведите пример как минимум двух операций, которые нельзя записать при помощи матриц 3x3, но можно записать в однородных координатах при помощи матрицы 4x4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Методы оптимизации**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

|  |  |
| --- | --- |
| **Контрольная работа № 1 (у каждого свой вариант). Примеры вариантов:** | |
| **Вариант 1** | **Вариант 2** |
| 1. К какому классу сложности принадлежит задача проверки делимости на 2019? | 1. К какому классу сложности относится задача, дополнительная к БЛН? |
| **Контрольная работа № 2 (у каждого свой вариант). Примеры вариантов:** | |
| **Вариант 1** | **Вариант 2** |
| Сформулировать теорему двойственности линейного программирования (ЛП). Построить двойственную задачу для задачи ЛП:  max (2х + 1)  -1 < x < 3  Решить прямую и двойственную задачи. Найти значения оптимума для прямой и двойственной задач, проверить, что они совпадают. | Сформулировать теорему двойственности линейного программирования (ЛП). Построить по определению двойственную задачу для задачи ЛП:  min (2y1  – 3y2 )  y1= y2  y1 > 0, y2 > 0  Выяснить существование решения прямой и двойственной задачи Если существуют, то найти значения оптимума для прямой и  двойственной задач, проверить, что они совпадают. |
| **Контрольная работа № 3 (у каждого свой вариант). Примеры вариантов:** | |
| **Вариант 1** | **Вариант 2** |
| Выписать 4 итерации метода градиентного спуска с постоянным шагом для задачи поиска минимума функции f (х1, х2) = 2х12 + 3х22  при начальном приближении (х1, х2) = (-1, 2) и шаге *а* = 1. Указать условия на шаг для сходимости метода. Решить задачу методом Ньютона (с тем же начальным приближением). | Выписать 4 итерации метода градиентного спуска с постоянным шагом для задачи поиска минимума функции f (х1, х2) = 2х12 + 4х22  при начальном приближении (х1, х2) = (-3, 2) и шаге *а* = 1. Указать условия на шаг для сходимости метода. Решить задачу методом Ньютона (с тем же начальным приближением). |

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Вопросы к экзамену**.

1. Определения: индивидуальной и массовой задачи, кодировки задачи, алгоритма решения массовой задачи, временной сложности алгоритма.
2. Формула градиентного метода в задаче безусловной минимизации.
3. Задачи распознавания свойств. Классы Р и NР.
4. Формула метода Ньютона в задаче безусловной минимизации.
5. Теорема об экспоненциальной временной оценке для задач из класса NР.
6. Идея метода штрафов.
7. Определение полиномиальной сводимости. Класс NРС.
8. Теорема Кука /без док-ва/.
9. Геометрическое описание симплекс-метода.
10. Критерий NР-полноты. Док-во NР-полноты задачи ЦЛH.
11. Методы глобальной минимизации.
12. Док-во NР-полноты задачи 3-выполнимость. NР-трудные задачи.
13. Формула градиентного метода в задаче безусловной минимизации.
14. Класс со-NР. Пример задачи, допускающей хорошую характеризацию.
15. Доказательство утверждения о взаимоотношении классов NРС и со-NР.
16. Формула метода Ньютона в задаче безусловной минимизации.
17. Взаимоотношение классов Р, NР и NРС, NР и со-NР. Класс РSРАСЕ.
18. Полиномиальный алгоритм округления ε1-приближенного решения системы линейных неравенств.
19. Псевдополиномиальные алгоритмы. Пример для задачи о рюкзаке.
20. Идея метода эллипсоидов.
21. Сильная NР-полнота. Теорема о связи сильной NР-полноты задачи с существованием псевдополиномиального алгоритма ее решения.
22. Идея метода штрафов.
23. Определение комбинаторной задачи оптимизации и приближенного алгоритма ее решения. Утверждение о разнице между приближенным и точным оптимумом для задачи о рюкзаке.
24. Идея метода Ньютона.
25. Определение е-приближенного алгоритма и полностью полиномиальной приближенной схемы /ПППС/. Связь между существованием ПППС и псевдополиномиальностью.
26. Теорема оптимальности для разложимых функций.
27. Теорема об отсутствии ПППС для задач оптимизации, соответствующих сильно NР-полным задачам распознавания.
28. Геометрическая идея симплекс-метода.
29. Определение озЛП. Принцип граничных решений. Алгебраическая и битовая сложность ЛП. Результаты о сложности для задач, близких к ЛП.
30. Идея метода ветвей и границ. Пример для задачи БЛП.
31. Теорема о границах решений задач ЛП с целыми коэффициентами.
32. Метод ветвей и границ для ЦЛП. Различные стратегии метода.
33. Теорема о мере несовместности систем линейных неравенств с целыми коэффициентами.
34. Метод ветвей и границ для глобальной минимизации липшицевых функций.
35. Следствия систем линейных неравенств. Афинная лемма Фаркаша /без док-ва/.
36. Понятие о временной сложности алгоритмов.
37. Лемма Фаркаша о неразрешимости.
38. Понятие о недетерминированно полиномиальных задачах.
39. Теорема двойственности ЛП.
40. Метод динамического программирования для БЛП с неотрицательными коэффициентами.
41. Сведение озЛП к однородной системе уравнений с ограничением x>0.
42. Применение метода динамического программирования для понижения размерности разложимой оптимизационной задачи.
43. Классификация задач математического программирования. Преимущества выпуклого случая.
44. Необходимые условия локального минимума при ограничениях-неравенствах для дифференцируемых функций.
45. Идея метода Кармаркара.
46. Понятие о регулярности ограничений-неравенств в задаче математического программирования.
47. Описание метода эллипсоидов.
48. Теорема о целочисленности решения задачи ЛП с целыми коэффициентами для вполне унимодулярных матриц ограничений.
49. Оценка сложности метода эллипсоидов ε2-приближенного решения озЛП.

**Типовые задачи для экзамена**.

1. Показать, что двойственная задача к двойственной задаче ЛП совпадает с прямой задачей ЛП /для озЛП/.
2. Расшифровать формулировку теоремы Куна-Таккера для случая ocновной задачи ЛП (озЛП).
3. Представить каноническую задачу ЛП в форме основной задачи ЛП.
4. Дать алгоритм распознавания простоты числа, оценить сложность алгоритма.
5. Доказать, что задача распознавания простоты числа принадлежит классу со-NР.
6. Оценить по порядку длину входа задачи коммивояжера.
7. Оценить по порядку битовую длину входа задачи ЛП.
8. Построить двойственную задачу к указанной.
9. Привести пример задачи, имеющей указанный класс сложности.
10. Выписать первые 5 итераций градиентного метода решения указанной задачи с предложенным шагом и начальным приближением, объяснить полученные свойства метода.
11. Решить методом Ньютона задачу минимизации.
12. Выписать алгоритм метода штрафов для конкретной задачи, придумать штрафную функцию для предложенной задачи.

Экзаменационный билет состоит из двух вопросов и задачи, например

1. Теорема о границах решений задач ЛП с целыми коэффициентами.

2. Метод ветвей и границ для ЦЛП. Различные стратегии метода.

3. Доказать, что задача распознавания простоты числа принадлежит классу со-NР.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Основы программной инженерии**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

|  |
| --- |
| **Контрольнаяработа № 1** |
| 1. Заполните следующую таблицу, указав в соответствующих столбцах номера перечисленных ниже характеристик более свойственных сложным программным системам, простым программам, а также тех, которые не связаны со сложностью ПО.   |  |  |  | | --- | --- | --- | | Более свойственно сложным программам | Более свойственно простым программам | Не связано с тем, сложная программа или нет | |  |  |  |  1. Исходный код, написанный на языке C 2. Необходимость разработки архитектуры до перехода к кодированию 3. Много людей вовлечено в создание системы 4. Отсутствие поддержки пользователей 5. Высокая стоимость разработки   2. Обозначим различные виды деятельности при разработке ПО следующим образом.   1. Анализ требований 2. Проектирование 3. Кодирование 4. Тестирование 5. Развертывание 6. Эксплуатация   Поставьте галочки в первом столбце приведенной ниже таблицы рядом с последовательностями выполнения этих видов деятельности, которые могут возникать при использовании итеративной модели жизненного цикла ПО.   |  |  | | --- | --- | |  | ACBCBABCDEF | |  | ABABABCDEF | |  | ABCDEFCDEF | |  | BACDBCDEF | |  | ABABCABCDEF | |  | ABCBADEF | |  | ABCABABCDEF | |  | ABCDEABDF |   3.Заполните следующую таблицу, поместив  **в первый столбец** номера перечисленных ниже техник, используемых при разработке ПО в соответствии с RationalUnifiedProcess (RUP),  **во второй столбец** — номера техник, используемых в ExtremeProgramming (XP),  **в третий столбец** — номера техник, используемых в обоих видах процессов,  **в четвертый столбец** — номера техник, не используемых ни в одном из этих двух процессов   |  |  |  |  | | --- | --- | --- | --- | | Техники, используемые в RUP | Техники, используемые в XP | Техники, используемые и в RUP, и в XP | Техники, не используемые ни в RUP, ни в XP | |  |  |  |  |  1. Использование наиболее простых решений 2. Использование компонентной архитектуры 3. Фиксация требований в виде вариантов использования 4. Совместные инспекции кода 5. Коллективное владение кодом 6. Нацеленность на создание продукта, работоспособного в реальном окружении 7. Использование моделей как основного средства коммуникации   4.Отметьте те из указанных элементов, которые входят в технику «программирование парами».   |  |  | | --- | --- | |  | Разработка кода выполняется парой программистов, один из которых пишет код, другой в это время отдыхает. | |  | Разработка кода выполняется парой программистов, один из которых пишет код, другой в это время дает ему советы и думает над улучшением кода. | |  | Пара программистов в течение всего проекта работает вместе. | |  | Объединение программистов в пары меняется от задачи к задаче. | |  | Разработка кода выполняется парой программистов, один из которых отвечает за все, сделанное ими, и руководит работой другого, а тот пишет код. | |  | Разработка кода выполняется парой программистов попеременно, то одним, то другим, они меняются местами каждые 5-10 минут. |   5.Занесите номера приведенных ниже описаний проблем предметной области в первый столбец таблицы, функций ПО — во второй столбец, требований к ПО— в третий.   |  |  |  | | --- | --- | --- | | Проблемы предметной области | Функции ПО | Требования к ПО | |  |  |  |  1. ПО анализа геологических данных должно повысить точность определения положения рудных тел и снизить ошибки определения их состава. 2. Web-сайт компании должен выдерживать нагрузки до 50000 запросов в час и должен быть недоступен не более 10-ти минут в год. 3. Система контроля городского движения должна существенно снизить количество аварий. 4. Пользователь-аналитик должен уже в первый день работы с системой понимать, как с ее помощью можно получить все виды годовых, квартальных и месячных отчетов о финансовой деятельности компании. 5. ПО контроля процесса производства должно автоматически отсеивать бракованные детали. 6. Пользователь должен иметь возможность установить размер шрифта всех выделенных символов равным любому целому числу пунктов, от 5 до 72.   6.К каким из характеристик качества ПО по ISO 9126 относятся описанные ошибки? Перечислите в первом столбце каждой строки с описанием ошибки номера соответствующих характеристик качества из предыдущего задания.   |  |  | | --- | --- | |  | Внесение первого же изменения, которое было затребовано пользователями, привело к кардинальной перестройке архитектуры системы. | |  | Код системы не содержит комментариев, плохо отформатирован и труден для восприятия. | |  | Для выполнения наиболее часто выполняемой пользователями операции системы — получения аналитического отчета за некоторый временной отрезок — требуется нажать не менее 6-ти кнопок на 4-х диалогах. | |  | Для установки системы необходимо разархивировать архив, содержащий установочный пакет, в директорию, находящуюся в корне диска C:, создать в реестре системы 5 новых записей и занести туда конфигурационные данные системы, создать .dat файл в установочной директории и записать туда в определенном формате конфигурацию драйвера видеокарты. | |  | Система размером около 50000 строк состоит из одного класса, который имеет только один public метод, все остальные 863 метода в этом классе — private. | |

**Вопросы к зачету**

1. Понятие программной системы.
2. Предмет и основные принципы программной инженерии. Свойства правильно спроектированных интерфейсов: адекватность, полнота, простота, минимальность.
3. Жизненный цикл ПО. Виды деятельности по разработке и сопровождению ПО. Основные модели жизненного цикла ПО: водопадная, итеративная, спиральная.
4. Стандарты на технологические процессы разработки и сопровождения ПО. Стандарты ISO 12207 и CMMI.
5. Примеры процессов разработки ПО. Унифицированный процесс разработки (RUP). Экстремальное программирование (XP).
6. Требования к ПО. Уровни абстракции требований: потребности, функции и детальные требования. Характеристики требований по стандарту IEEE 830.
7. Характеристики и атрибуты качества ПО по стандарту ISO 9126. Ошибки в ПО.
8. Методы контроля качества ПО. Тестирование ПО. Виды тестирования и характеристики тестов.
9. Архитектура ПО. Основные архитектурные стили: конвейер, интерактивные системы, вызов-возврат, хранилища данных, интерпретация. Анализ архитектуры с помощью метода SAAM.
10. Графические нотации, используемые при разработке ПО. Диаграммы потоков данных, диаграммы сущностей и связей, основы языка UML.
11. Образцы проектирования ПО. Образцы анализа. Идиомы. Образцы организации работ.
12. Удобство использования ПО. Принципы организации удобного интерфейса и типичные проблемы удобства ПО.
13. Понятие распределенной программной системы. Основные характеристики распределенных систем.
14. Технологии разработки Web-приложений на основе J2EE и .NET.
15. Планирование проектов по разработке ПО. Диаграммы задач и зависимостей. Выделение критического пути на графе задач.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Сложность алгоритмов**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

|  |  |
| --- | --- |
| **Контрольная работа № 1** | |
| **Вариант 1** | **Вариант 2** |
| 1. Указать все вещественные значения δ при которых справедлива оценка  а) TTD(n) = O(nδ),  б) TTD(n) = Ω(nδ),  в) TTD(n) = Θ(nδ),  где TTD(n) – сложность алгоритма пробных делений.  2. Верно ли, что для рассмотрения сложности в среднем некоторого алгоритма требуется задание распределения вероятностей  а) на множестве всех допустимых входов?  б) на каждом из множеств всех входов фиксированного размера?  3. При некоторых значениях n число сравнений, затрачиваемых  при бинарном поиске, не определяется однозначно исходя лишь из  значения n (например, зная лишь, что n = 6, мы не можем указать  точное число сравнений). Но существует бесконечно много n таких,  для которых это значение определяется единственным образом и рав-  но ⌊log2 n ⌋+1. Доказать.  4. Пусть QS (n) и opt (n) – сложности в среднем быстрой сортировки и оптимальной в среднем сортировки. Верно ли, что QS (n) ∼ opt (n)? Если нет, то можно ли подобрать константу c такую, что QS (n) ∼ c opt (n)? | 1. Известно, что мультипликативная сложность метода  Гаусса решения системы n линейных уравнений с n неизвестными  допускает оценки  1) O(n3), 2) + O(n2), 3) Θ(n3).  а) Из какой оценки (указать номер) следуют две остальные?  б) Можно ли из приведенных оценок выбрать такую, которая является следствием любой из остальных?  в) Является ли оценка Ω(n3) следствием какой-либо из оценок 1, 2, 3?    2. Верно ли, что определение усредненных затрат некоторого  рандомизированного алгоритма требует задания распределения вероятностей  а) на множестве всех допустимых входов?  б) на каждом из множеств всех входов фиксированного размера?  3. Верно ли, что сложность по числу сравнений сортировки массива из n элементов бинарными вставками есть n log2 n+ O(1)?  4. Может ли сортировка со сложностью в худшем случае Ω(n log n) по числу перемещений элементов быть оптимальной по порядку сложности по числу сравнений? |
|  | |
| **Контрольная работа № 2** | |
| **Вариант 1** | **Вариант 2** |
| 1.Исследовать битовую сложность вычисления величины 1 +2+ ... +n последовательными сложениями. Размером входа считать битовую длину m целого числа n.  2. Определить сложность алгоритма перекладывания дисков в игре «Ханойские башни» при условии, что затраты на перекладывание со столбика на столбик i-го по величине диска равны i2.  3. Здесь речь идет о линейной сводимости P ≤ Q задач, связанных с мультипликативными операциями над квадратными числовыми матрицами порядка n. Рассматриваются лишь такие алгоритмы решения задачи Q, для сложности по числу арифметических операций каждого из которых выполняется соотношение T(kn) = O(T(n)), k =2, 3.  Требуется показать, что задача умножения произвольных квадратных матриц линейно сводится к задаче умножения симметричных квадратных матриц. | 1. Из определения чисел Фибоначчи видно, что можно вычислить Fn, выполнив n−1 сложение. Доказать, что битовая сложность этого алгоритма допускает оценку O(n2) при рассмотрении n в качестве размера входа.  2. Определить сложность алгоритма перекладывания дисков в игре «Ханойские башни» при условии, что затраты на перекладывание со столбика на столбик i-го по величине диска равны 2i .  3. Здесь речь идет о линейной сводимости P ≤ Q задач, связанных с мультипликативными операциями над квадратными числовыми матрицами порядка n. Рассматриваются лишь такие алгоритмы решения задачи Q, для сложности по числу арифметических операций каждого из которых выполняется соотношение T(kn) = O(T(n)), k =2, 3.  Требуется показать, что задача умножения произвольных квадратных матриц линейно сводится к задаче умножения верхних треугольных матриц. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Формальные языки и автоматы**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Типы заданий контрольной работы №1**

* + - 1. Построить РВ, описывающее заданный язык.
      2. Построить НКА по РВ.
      3. Построить ДКА по РВ.
      4. Построить минимальный ДКА, эквивалентный данному ДКА.
      5. Построить ДКА для симметрической разности двух языков, заданных ДКА.
      6. Построить ДКА, принимающий заданный язык.
      7. С помощью леммы о накачке доказать, что заданный язык не является регулярным.

**Типы заданий контрольной работы №2**

* + - 1. Привести грамматику к нормальной форме Хомского.
      2. Построить МП-автомат для заданного языка.
      3. Построить -анализатор для языка, заданного КС-грамматикой.
      4. При помощи алгоритма Кока-Янгера-Касами определить, принадлежит ли цепочка заданному КС-языку.
      5. Доказать, что заданный язык не является КС.
      6. Построить -анализатор для языка, заданного КС-грамматикой.

## Письменные опросы

Во время лекций проводятся письменные опросы. Каждый опрос включает 3 тестовых вопроса, подобранных случайным образом из множества заготовленных заданий для обеспечения индивидуальных вариантов. Пример варианта опроса:

1. На обратной стороне нарисуйте диаграмму ДКА, распознающего целые десятичные числа больше 1. (Можно использовать запись ``5..9'' для сокращения количества стрелок)
2. Напишите регулярное выражение, задающее множество цепочек из нулей и единиц, содержащих не более одной пары последовательных единиц.
3. Приведите примеры таких языков L\_1, L\_2, что L\_1 \subseteq L\_2 и L\_1 -- регулярный, а L\_2 -- нерегулярный. Будьте готовы доказать регулярность/нерегулярность L\_1 и L\_2.

## 

## Практические задания

1. Построить лексический анализатор для модельного языка при помощи генератора лексических анализаторов LEX. Подготовить репрезентативный набор тестовых данных для поверки корректности результатов и обработки ошибок во входных данных. Пройти автоматизированную проверку разработанного ПО в системе ejudge.
2. Построить синтаксический анализатор с использованием лексического анализатора, разработанного в практическом задании №1. Реализовать транслятор с модельного языка в промежуточное представление вида 3-адресный код. Подготовить репрезентативный набор тестовых данных для поверки корректности результатов и обработки ошибок во входных данных. Пройти автоматизированную проверку разработанного ПО в системе ejudge.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

## Программа экзамена по курсу

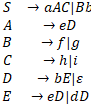
1. Грамматики (определение грамматики, классификация и примеры)
2. Регулярные множества и выражения, конечные автоматы (РМ, РВ, КА).
3. Построение ДКА по НКА. Алгоритм построения КА по РВ.
4. Связь РМ, КА и регулярных грамматик.
5. Автоматы с МП (определения и примеры). Связь МП-автоматов и КС-грамматик. Приведённые грамматики (алгоритмы)
6. Лемма о разрастании (применение при доказательстве не-КС-грамматик).
7. Функции FIRST и FOLLOW. LL(k) –грамматики (определения и примеры).
8. Необходимые и достаточные условия. Построение LL(1)-анализатора.
9. LR(k)-грамматики (определения и примеры).
10. LR(k)-анализатор.
11. Необходимые и достаточные условия LR(k).

## Вариант письменного экзамена по курсу

* + 1. Напишите регулярное выражение, задающее множество всех цепочек в алфавите , которые содержат не более трёх символов  подряд.
    2. Для заданной КС-грамматики  постройте эквивалентную -грамматику. Докажите, что получившаяся грамматика является -грамматикой.

.

* + 1. Постройте ДКА по РВ .
    2. Над алфавитом  задан язык . Докажите, что язык  не является контекстно-свободным.
    3. Постройте -анализатор для грамматики , где 



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Практикум на ЭВМ**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Практическое самостоятельное задание №1**

Задана матрицаA = (aij) порядка n. Требуется найти LU разложение матрицы A и, используя это разложение, решить уравнение Ax=b. МатрицаLдолжнахранитьсянаместенижнеготреугольникаматрицыA, аматрицаUнаместеверхнеготреугольникаматрицыA.

Требования к программе:

* Программадолжназапрашиватьразмерматрицыупользователя (запроснавводсклавиатуры/передачачерезкоманднуюстроку)
* Программадолжнаподдерживатьдваспособавводаматрицы: изфайлаилипоформуле, имеющейсявпрограмме.
* Решениесистемыдолжнобытьоформленоввидеподпрограммыиполучающейвкачествеаргументов:

а) размерность n матрицыA,  
б) матрицуA,  
в) правуючастьb,  
г) векторx, вкоторыйбудетпомещенорешение  
Получатьвэтойподпрограммедополнительнуюинформациюизвнечерезглобальныепеременные, общиеблоки, включаемыефайлыит.п. запрещается.

* Суммарныйобъемоперативнойпамяти, требуемыйпрограмме, недолженпревышать n2 + O(n)
* ПрограммадолжнасодержатьподпрограммувыводанаэкранрешениясистемыиматрицLиU.Вслучае, еслиразмерисходнойматрицыбольше 10, выводматрицLиUнеобходимоосуществлятьвфайл.
* Программадолжнасодержатьфункцию, вычисляющуюнормуневязки, т.е.  ||Ax - b||.
* Вслучае, еслиматрицаAвырожденаневыполнятьLUразложение, авывестисообщениеовырожденностиматрицы.

Практическое самостоятельное задание№2

СимметричнаявещественнаяматрицаA = (aij) порядка n. ТребуетсянайтивсесобственныезначенияисобственныевекторыматрицыA,используятриразличныестратегиивыбораобнуляемогоэлемента.

Требованиякпрограмме:

* Программадолжназапрашиватьразмерматрицыупользователя (запроснавводсклавиатуры/передачачерезкоманднуюстроку)
* Программадолжнапредоставлятьвозможностьвыборастратегииобнуляемогоэлемента (триварианта)
* Программадолжнаподдерживатьдваспособавводаматрицы: изфайлаилипоформуле, имеющейсявпрограмме.
* Решениесистемыдолжнобытьоформленоввидеподпрограммыиполучающейвкачествеаргументов:

     а) размерность n матрицыA,  
     б) матрицуA,  
     в) матрицу, вкоторуюбудутпомещенысобственныевектора  
     г) вслучаенеобходимости, векторx, вкоторыйбудутпомещенысобственныезначения.  
     д) точностьε  
     е) другиевспомогательныепеременныевслучаенеобходимости  
    Получатьвэтойподпрограммедополнительнуюинформациюизвнечерезглобальные  переменные, общиеблоки, включаемыефайлыит.п. запрещается.

* Суммарныйобъемоперативнойпамяти, требуемыйпрограмме, недолженпревышать 2n2 + O(n)
* ПрограммадолжнасодержатьподпрограммувыводанаэкрансобственныхзнченийисобственныхвекторовматрицыA. Вслучае, еслиразмерисходнойматрицыбольше 10, выводсобственныхвекторовнеобходимоосуществлятьвфайл. (собственныезначениявлюбомслучаевыводятсянаэкран)
* Программадолжнавыводитьнаэкранпроизведенноечислоитераций,
* Программадолжнавыводитьнаэкранвремя, затраченноенанахождениесобственныхзначенийисобственныхвекторов.
* Времяработыпрограммынедолжнопревыщать O(n3)

Практическое самостоятельное задание№3

Написатьпрограмму, котораянаходитчисленноерешениеуравнениятеплопроводностилюбымидвумяизтрехразностныхсхем (явная, неявная, Кранка-Николсона)

 Требованиякпрограмме:

1) Программадолжнапредоставлятьвозможностьзадаватьграничныеусловия I и II роданакаждомконцеотрезка (втомчислеисмешанныйвариант, например, налевомконцеграничноеусловие I рода, анаправом - II рода)

2) Программадолжнаработатьвдвухрежимах:  
   а) врежиметестовойпроверки, вэтомслучаепозаданномуточномурешению  u(t,x) (задаетсяпреподавателем) вычисляютсясоответствующиефункции f(t,x), u0(x), φ1(t)/φ2(t)/ψ1(t)/ψ2(t). Этифункции задаютсявпрограммеипонимвычисляетсячисленноерешениепри t = 1.   
   б) врежимевизуализации, вэтомслучаефункции f(t,x), u0(x), φ1(t)/φ2(t)/ψ1(t)/ψ2(t) задаютсяпреподавателем, апрограммавычисляетчисленноерешениедовремени t = t0.

 3) Втестовомрежимепрограммадолжнавыводитьнаэкран:  
    а) абсолютнуюпогрешностьвнорме Ch  
    б) абсолютнуюпогрешностьвнорме l2,h  
    в) относительнуюпогрешностьвнорме Ch  
    г) относительнуюпогрешностьвнорме l2,h

 4) Врежимевизуализациипрограммадолжналибо (снекоторойпериодичностью) сохранятьданныедляпоследующейвизуализациистороннимипрограммами, либосвоимисредствамиосуществлятьвизуализациючисленногорешения.

 5) Программанедолжнатребоватьпамятибольше, чем O(M), где M - числоточекнаотрезке [0,1], вкоторыхзаданафункция.

Практическое самостоятельное задание №4

РеализоватьпараллельнуюпрограммунаС++ сиспользованием OpenMP, котораявыполняетоднокубитноеквантовоепреобразованиенадвекторомсостоянийдлины 2n, где n–количествокубитов, поуказанномуномерукубита k.

1. Дляработыскомплекснымичисламивозможноиспользованиестандартнойбиблиотекишаблонов.
2. Определитьмаксимальноеколичествокубитов, длякоторыхвозможнаработапрограммынасистеме Polus. Выполнитьтеоретическийрасчетипроверитьегоэкспериментально.
3. Протестироватьпрограммунасистеме Polus. ВкачестветестаиспользоватьпреобразованиеАдамарапономерукубита:
   1. которыйсоответствуетВашемуномерувспискегруппыплюс 1.
   2. 1
   3. n
4. Написатьотчет, которыйбудетсодержатьлистингпрограммы, атакжерезультатывыполненияпунктов 2-3.

Практическое самостоятельное задание №5

1. Разработатьсхемураспределенногохраненияданныхипараллельныйалгоритмдляреализацииоднокубитногоквантовогопреобразованиянакластернойсистеме.
2. РеализоватьпараллельнуюпрограммунаС++ сиспользованием MPI, котораявыполняетоднокубитноеквантовоепреобразованиенадвекторомсостоянийдлины 2n, где n–количествокубитов, поуказанномуномерукубита k. Протестироватьпрограммунасистеме Polus Использовать 32-битный mpi. ВкачестветестаиспользоватьпреобразованиеАдамарапономерукубита:
   1. которыйсоответствуетВашемуномерувспискегруппыплюс 1.
   2. 1
   3. n
3. Начальноесостояниевекторадолжногенерироватьсяслучайнымобразом (генерациявекторатожедолжнабытьраспараллелена). Построитьграфикзависимостиускоренияпараллельнойпрограммыотчислапроцессоровдлякаждогоизслучаев a)-c).
4. Провеститестированиеразработанногорешениянакорректность. Необходимозапуститьпрограммунаодномитомжевходномвекторена 1,2,4 и 8 процессорахипоказать, чтовыходнойвекторсовпадаетвовсехэкспериментах. Тестпровестидлямоделирования 16 кубитов. Входнойвектордолженбытьсгенерированслучайнымобразом.
5. Написатьотчет, которыйбудетсодержатьлистингпрограммы, атакжерезультатывыполненияпунктов 1-4. Схемапараллельного
6. алгоритмаихраненияраспределенныхданныхдолжнабытьформальноописана.

Практическое самостоятельное задание №6

1. Реализовать параллельную программу на С++ с использованием MPI, которая выполняет квантовое преобразование n-Адамар с зашумленными вентилями над вектором состояний длины 2n, где n–количествок убитов.
2. Протестировать программу на системе BlueGene/P. Точностье=0.01. Использовать 64-битную адресацию. Результаты представить в виде таблицы.
3. Построить график распределения потерь точности при фиксированной точности е=0.01для количества кубитов 24, 25, 26, 27, 28. Для построения каждого распределения использовать не менее 60 экспериментов. Входной вектор в экспериментах должен генерироваться случайным образом. (Всего должно быть пять распределений, соответствующие разному количеству кубитов).
4. Построить график распределения потерь точности при фиксированном количестве кубитов n=26 и различных значениях точности: е=0.1, е=0.01, е=0.001. Для построения каждого распределения использовать не менее 60 экспериментов. Входной вектор в экспериментах должен генерироваться случайным образом.

Практическоесамостоятельноезадание№7-8

Реализовать двухкубитное квантовое преобразование с использованием технологий MPI и OpenMP, с тестами на корректность.

* 1. Реализовать гейты H (Адамара), n-Адамара, Rw (поворота на w), NOT, C-NOT, контролируемого поворота на w.
  2. Для каждого из них написать тесты на корректность.
  3. Код выполнения операций должен быть оформлен в виде библиотеки, тесты должны быть в отдельных от самой библиотеки файлах.
  4. Формат входного вектора (порядок записи) должен быть одинаковый и не зависеть от параметров гейтов.
  5. Используя результаты заданий 4-7 реализовать квантовое преобразование Фурье.

Практическое самостоятельное задание№9

Реализовать унитарную эволюцию по уравнению Шредингера с логгированием состояний.

Требования к программе:

В качествев ходных данных использовать:

* Размерность системы N
* R - изначальная матрица плотности, в виде либо матрицы N\*N, либо вектора размера N, либо числа (номера базисного состояния системы).
* dT - шаг по времени
* H - гамильтониан системы
* количество шагов— n

Программа должна выводить последовательность диагоналей матрицы плотности, с шагом по времени dT в виде:

Abs(Diag(R(0))

Abs(Diag(R(dT)))

Abs(Diag(R(2\*dT)))

…

Abs(Diag(R(n\*dT)))

Код должен быть оформлен в виде библиотеки на C/C++, матричные операции должны быть реализованы с использованием библиотеки Scalapack.

Тип чисел: complex<double>.Программа должна по максимуму использовать выделенные при запуске распределённые процессы.

Практическое самостоятельное задание №10-11

НаписатьпрограммувычисленияраспределенногогамильтонианаинеунитарнойэволюциииспользуяуравнениеЛинбада. Заданиедаетсяповариантам. Примеродногоизвариантов:

использоватьцепочкуиздвухуровневыхатомоввоптическихполостяхсостоком (обменфотонами), приэтомпрограммадолжнаполучатьвкачествевходныхданных:

* N —количествоатомов (иполостей)
* a —амплитудаобменафотонамимеждусоседнимиполостями
* b —взаимодействиемеждуфотонамииатомами
* wa —частотаатомов
* wc —частотафотонов
* k —размерностьстока
* Emin, Emax —минимальныеимаксимальныезначенияколичествафотоновивозбуждений (всуммесэнергиейстока) ввыбираемыхсостояниях

Вкачествевыходныхданныхпрограммадолжнавыдаватьматрицугамильтонианаиегоразмер.

Используяполученныерезультатыизмеритьконкретныепараметрысистемы (длякаждоговариантасвои). Результатыпредставитьввидетаблициграфиков.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Конструирование компиляторов**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости**

## Комплексное индивидуальное практическое задание

В качестве практического задания студентам предлагается попробовать самостоятельно реализовать одно из оптимизирующих преобразований, используя инфраструктуру компилятора LLVM и его промежуточное представление – биткод.

Примеры оптимизирующих преобразований:

* Нахождение доступных выражений (для исключения избыточных вычислений)
* Планирование кода в суперблоке
* Распространение копий
* Вынос инвариантных вычислений в предзаголовок цикла.
* Распространение констант (с вычислением или без)

На первом этапе выполнения задания (первая - вторая недели) студент должен изучить среду LLVM, ее промежуточное представление (биткод), и, воспользовавшись компилятором Clang, получить биткод своего задания.

На втором этапе, пользуясь возможностями LLVM, он должен составить фазу (программу), выполняющую требуемое преобразование, и включить ее в состав учебного компилятора на базе LLVM.

Для проверки задания используется автоматическая система на удаленном сервере.

Срок выполнения задания – 12 недель. В конце семестра выставляются итоговые технические баллы.

## Пример учебно-методического обеспечения для практического задания

### Постановка задачи

Целью работы является создание оптимизирующего преобразования, позволяющего повысить качество кода, генерируемого учебным компилятором на базе LLVM. Задача оптимизации кода состоит в повышении его быстродействия и/или сокращения его размера.

Предлагается разработать преобразование, которое на вход получает программу в промежуточном представлении LLVM, а на выходе генерирует ее версию, оптимизированную по размеру и скорости выполнения путем удаления недостижимого и бесполезного кода. Например:

## Первоначальная программа:

**int main (void) {**

**printf(“First message \n”);**

**int a = 34+48;**

**if (0)**

**printf(“Unreachable code\n”);**

**a = a \* 2;**

**printf(“Second message\n”);**

**return 0;**

**}**

## Программа после удаления мертвого и недостижимого кода:

**int main (void) {**

**printf(“First message \n”);**

**printf(“Second message\n”);**

**return 0;**

**}**

### Решение задачи

## Теоретические аспекты

Предполагается реализация алгоритмов удаления мертвого и недостижимого кода. *Недостижимым кодом*называют часть кода программы, которая ни при каких условиях не может быть исполнена, поскольку является недостижимой в графе потока управления.  
*Мертвый (неиспользуемый, бесполезный) код* - команды, вычисляющие никогда не использующиеся значения.

## Практические аспекты

Решения должны быть написаны на языке С++ с использованием контейнеров и алгоритмов стандартной библиотеки, а также средств, предоставляемых LLVM. Оптимизирующие преобразования должны быть выполнены в учебном компиляторе, который представляет собой модифицированную версию LLVM, не содержащую оптимизирующих преобразований.

Пример сборки LLVM в ОС Linux:

В каталоге с исходными кодами следует создать каталог с названием **build** и перейти в него:

**$ cd llvm/**

**$ mkdir build**

**$ cd build**

Затем следует осуществить сборку LLVM с указанием каталога сборки и желаемого типа сборки (Debug/Release). Сборка осуществляется с помощью утилит **configure** и **make**. Ключи **configure**:

--prefix=<путь к каталогу инсталляции>

--disable-assertions/--enable-assertion – включить/выключить проверку утверждений

--enable-optimized/--disable-optimized – выбрать тип сборки: оптимизированная (Release) или без оптимизаций (Debug)

Для ускорения компиляции рекомендуется использовать ключ “**-jN**” утилиты **make**, где N -- число желаемых потоков сборки (как правило, равное количеству ядер процессора или количеству ядер процессора + 1).

Пример сборки Debug с выключенной проверкой утверждений и установкой в каталог </home/user/llvm> и сборкой в 8 потоков:

**$ ../configure –prefix=/home/user/llvm –disable-optimized – disable-assertions**

**$ make –j8**

**$ make install**

После завершения установки и компиляции требуется добавить путь до каталога с LLVM в переменную окружения PATH:

**$ export PATH=/home/user/llvm/bin:$PATH**

Получить промежуточное представление LLVM для программы можно, выполнив команду:

**$ clang –c –O0 –emit-llvm test.c –o test.bc**

Перевод бинарного представления в ассемблер LLVM

**$ llvm-dis test.bc –o test.ll**

Ассемблирование в бинарное представление

**$ llvm-as test.ll –o newTest.bc**

Запуск преобразования pass\_name из динамической библиотеки **pass\_name.so**

**$ opt -load <путь/до/динамической/библиотеки/>pass\_name.so -<pass\_name> test.bc –o transformedTest.bc**

test.bc - файл, содержащий бинарную версию промежуточного представления LLVM.

test.ll – файл, содержащий ассемблер LLVM в читаемом виде.

TH

**Написание оптимизирующего прохода LLVM**

Для выполнения работы могут потребоваться проходы следующих типов: FunctionPass (по функциям), ModulePass (по модулям), BasicBlockPass (по базовым блокам).

В качестве примера приведен проход по функциям и именем Hello, вызываемым из командной строки с помощью ключа “–hello”:

**#include "llvm/Pass.h"**

**#include "llvm/IR/Function.h"**

**#include "llvm/Support/raw\_ostream.h"**

**using namespace llvm;**

**namespace {**

**struct Hello : public FunctionPass {**

**static char ID;**

**Hello() : FunctionPass(ID) {}**

**virtual bool runOnFunction(Function &F) {**

**errs() << "Hello: ";**

**errs().write\_escaped(F.getName()) << '\n';**

**return false;**

**}**

**};**

**}**

**char Hello::ID = 0;**

**static RegisterPass<Hello> X("hello", "Hello World Pass", false, false);**

**Листинг 1. Пример компиляторного прохода**

Данный код выводит на экран сообщения вида “Hello: <function\_name>”.

Более детально изучить построение оптимизирующих проходов можно посмотреть в документации на компилятор (страница WritingAnLLVMPass.html в каталоге docs в поставке учебного компилятора).

### Тестирование

На личной странице расположена форма загрузки файла, а также информация о результатах тестирования и минимальный набор синтетических тестов.

**Загрузка решения.** Загружаемый файл должен представлять собой текст программы на языке С++, содержащий компиляторный проход, запускаемый по ключу “-**dce**”. После загрузки решение будет скомпилировано и запущенно с помощью утилиты “**opt**” на тестовом наборе данных.

Оценивается последнее присланное решение.

Для тестирования на локальной машине во время разработки предлагается осуществлять компиляцию программы с уровнем оптимизации “**O0**”. Локальное тестирование предлагается осуществлять с помощью программ с открытыми исходными кодами и минимального набора синтетических тестов.

## Оценка

Обязательное условие:

Присланное решение должно проходить проверку на корректность на программах **SQlite** и **Lzma.**

Оценка эффективности будет производиться на расширенном синтетическом наборе тестов.

«**Удовлетворительно**» – реализовано удаление бесполезного кода на уровне функций.

«**Хорошо**» - реализовано удаление бесполезного и недостижимого кода на уровне функций. Удаление избыточных вызовов “pure” функций (не имеющих побочных эффектов и зависящих только от аргумента), например в выражении **double y=sin(x),** если y нигде более не используется вызов **sin(x)** можно удалить.

«**Отлично**» - реализовано удаление бесполезного и недостижимого кода на уровне функций. Реализован алгоритм удаления частично избыточного кода, алгоритм описан в публикации: Jens Knoop, Oliver Rüthing, and Bernhard Steffen. 1994. Partial dead code elimination.*SIGPLAN Not.* 29, 6 (June 1994), 147-158. DOI=10.1145/773473.178256 http://doi.acm.org/10.1145/773473.17825.

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Вопросы к экзамену**

1. Структура оптимизирующего компилятора.. Построение промежуточного представления программы.
2. Базовые блоки и граф потока управления. Биткод среды LLVM – пример промежуточного представления.
3. Локальная оптимизация. Метод нумерации значений: представление базового блока в виде направленного ациклического графа.
4. Анализ потока данных – основной метод глобальной оптимизации. Примеры анализа потока данных – анализ достигающих определений
5. Анализ живых переменных. Исключение мерьвого кода.
6. Вынесение инвариантных вычислений за пределы цикла.
7. Граф потока управления: остовное дерево, обход, нумерация вершин, классификация дуг.
8. Отношение доминирования и построение дерева доминаторов
9. Построение естественных циклов и гнезд циклов.
10. SSA-форма промежуточного представления и ее построение. Граница доминирования.
11. Анализ потока данных в SSA-форме. Выявление доступных выражений. Исключение избыточности.
12. Обоснование анализа потока данных: полурешетки, передаточные функции, общий итерационный алгоритм.
13. Методы ускорения анализа потока данных. Суперблоки и другие области графа потока управления.
14. Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков. Пример – анализ достигающих определений.
15. Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков на примере анализа достигающих определений.
16. Глобальный метод нумерации значений – использование дерева доминаторов.
17. Глобальный анализ указателей. Псевдонимы (алиасы). Недостаточность глобального анализа.
18. Межпроцедурный анализ. Использование графа вызовов.
19. Межпроцедурный анализ. Методы учета контекста вызова.
20. Задачи, решаемые на этапе машинно-ориентированной оптимизации.
21. Планирование кода.
22. Распределение регистров.
23. Оптимизация потока управления, возвраты из рекурсивных функций.
24. Раскрутка циклов.
25. Открытая вставка функций.
26. Генерация объектного кода методом переписывания дерева

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Математическая логика и логическое программирование**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Вопросы к коллоквиуму.**

1. Синтаксис и семантика логики предикатов. Термы, формулы, интерпретация. Отношение выполнимости формулы на интерпретации.
2. Выполнимость, общезначимость, противоречивость формул логики предикатов. Примеры общезначимых и противоречивых формул логики предикатов. Модель. Логическое следствие. Теорема о логическом следствии.
3. Проблемы выполнимости и общезначимости. Пример формулы, не имеющей конечных моделей.
4. Семантические таблицы в логике предикатов. Табличный вывод. Теорема корректности табличного вывода.
5. Теорема полноты табличного вывода.
6. Теорема Лёвенгейма-Сколема. Теорема компактности Мальцева.
7. Равносильные формулы. Примеры равносильных формул. Теорема о равносильной замене.
8. Предваренная нормальная форма. Теорема о приведении формулы к предваренной нормальной форме.
9. Сколемовская стандартная форма. Теорема о приведении формулы к сколемовской стандартной форме.
10. Эрбрановский универсум, эрбрановский базис, эрбрановские интерпретации. Теорема об эрбрановской модели для сколемовской стандартной формы. Сведение проблемы общезначимости формул к проблеме противоречивости систем дизъюнктов. Теорема Эрбрана.
11. Подстановки. Применение подстановок к термам и формулам. Композиция подстановок. Унификатор. Наиболее общий унификатор.
12. Сведение задачи унификации к задаче решения системы термальных уравнений. Лемма о связке. Алгоритм унификации. Теорема о корректности и завершаемости алгоритма унификации.
13. Метод резолюций для логики предикатов: правила резолюции и склейки, резолютивный вывод. Теорема корректности резолютивного вывода.
14. Лемма о подъеме. Теорема полноты резолютивного вывода для логики предикатов.
15. Общая схема доказательства общезначимости формул логики предикатов методом резолюций. Стратегии резолютивного вывода.

**Билет для коллоквиума** содержит 12 вопросов, например:

**Вопрос 1.** Используя константные, функциональные и предикатные символы алфавита (см. Приложение 2), построить замкнутую формулу логики предикатов, соответствующую следующему утверждению.

“ Какова бы ни была последовательность действительных чисел и отрезок [a,b] действительных чисел, если бесконечно много элементов этой последовательности содержится в данном отрезке, то хотя бы одна предельная точка данной последовательности также содержится в этом отрезке”

**Вопрос 2.** Для заданной формулы ϕ выяснить, применяя метод семантических таблиц, является ли эта формула общезначимой.

**∃x (∀y P(x,y) → R(x)) →∀z ∀y (P(z,y) ∨ R(y))**

**Вопрос 3.** Для заданной формулы ϕ выяснить (применяя методрезолюций), является ли эта формула общезначимой.

**∀x (∀y∃v∀u ((A(u,v) →B(y,u)) & (¬∃wA(w,u) →∀zA(z,v))) →∃yB(x,y))**

**Вопрос 4.**Какая формула называется логическим следствием множества замкнутых формул **Γ**? Сформулируйте теорему о логическом следствии. Существует ли такое множество предложений **Γ**, логическим следствием которого является любая замкнутая формула?

**Вопрос 5.** Известно, что для семейств замкнутых формул **Γ, Δ**семантическая таблица **〈Γ,Δ〉** не имеет ни одного успешного табличного вывода. Какие из приведенных ниже утверждений всегда справедливы и почему?

1. Система формул **Γ**имеет хотя бы одну модель, потому что....
2. Система формул **Δ**имеет хотя бы одну модель, потому что....
3. Ни одна формула **ψ∈Δ** не является общезначимой, потому что....
4. Ни одно из перечисленных выше утверждений не верно, потому что...

**Вопрос 6.** Пусть **S** – некоторое множество дизъюнктов, а **S′** – множество всех формул, которые встречаются хотя бы в одно резолютивном выводе из множества дизъюнктов **S**. Какие из приведенных ниже утверждений справедливы и почему?

1. Если **S** – непротиворечивое множество формул, то и **S′**– непротиворечивое множество.
2. Если **S** – противоречивое множество формул, то и **S′**– противоречивое множество.
3. Если **S′** – непротиворечивое множество формул, то и **S**– непротиворечивое множество.
4. Если **S′** – противоречивое множество формул, то и **S**– противоречивое множество.
5. Одно из утверждений неверно, но какое именно, зависит от конкретного множества формул **S**.

**Вопросы к экзамену**

**Язык логики предикатов**

1. Синтаксис и семантика логики предикатов. Термы, формулы, интерпретация. Отношение выполнимости формулы на интерпретации.
2. Выполнимость, общезначимость, противоречивость формул логики предикатов. Примеры общезначимых и противоречивых формул логики предикатов. Модель. Логическое следствие. Теорема о логическом следствии.
3. Проблемы выполнимости и общезначимости. Пример формулы, не имеющей конечных моделей.
4. Семантические таблицы в логике предикатов. Табличный вывод. Теорема корректности табличного вывода.
5. Теорема полноты табличного вывода.
6. Теорема Лёвенгейма-Сколема. Теорема компактности Мальцева.
7. Равносильные формулы. Примеры равносильных формул. Теорема о равносильной замене.

#### Метод резолюций

1. Предваренная нормальная форма. Теорема о приведении формулы к предваренной нормальной форме.
2. Сколемовская стандартная форма. Теорема о приведении формулы к сколемовской стандартной форме.
3. Эрбрановский универсум, эрбрановский базис, эрбрановские интерпретации. Теорема об эрбрановской модели для сколемовской стандартной формы. Сведение проблемы общезначимости формул к проблеме противоречивости систем дизъюнктов. Теорема Эрбрана.
4. Подстановки. Применение подстановок к термам и формулам. Композиция подстановок. Унификатор. Наиболее общий унификатор.
5. Сведение задачи унификации к задаче решения системы термальных уравнений. Лемма о связке. Алгоритм унификации. Теорема о корректности и завершаемости алгоритма унификации.
6. Метод резолюций для логики предикатов: правила резолюции и склейки, резолютивный вывод. Теорема корректности резолютивного вывода.
7. Лемма о подъеме. Теорема полноты резолютивного вывода для логики предикатов.
8. Общая схема доказательства общезначимости формул логики предикатов методом резолюций. Стратегии резолютивного вывода.

#### Основы логического программирования

1. Использование метода резолюций для нахождения ответов на запросы. Истолкование резолютивного вывода как вычисления. Примеры вычислительных возможностей резолютивного вывода.
2. Хорновские дизъюнкты. Синтаксис языка логического программирования: логические программы и запросы. Декларативная семантика логических программ. Правильный ответ.
3. SLD-резолюция. SLD-резолютивные вычисления (опровержения) логических программ. Процедурная интерпретация SLD-выводов. Примеры SLD-опровержений успешных, тупиковых и бесконечных. Вычислимый ответ. Операционная (процедурная) семантика логических программ.
4. Теорема корректности SLD-резолютивных вычислений логических программ.
5. Теоремы полноты SLD-резолютивных вычислений логических программ.
6. Правило вычислений и его роль. R-вычислимый ответ. Переключательная лемма. Теорема о независимости правила вычислений. Теорема сильной полноты SLD-резолюции.
7. Дерево SLD-вычислений логических программ. Стратегии вычислений. Полные и неполные стратегии вычислений. Стандартная стратегия исполнения логических программ. Неполнота стандартной стратегии.
8. Управление исполнением логических программ. Оператор отсечения. Операционная семантика оператора отсечения.
9. Отрицание в Прологе. Допущение замкнутости мира. Отрицание как неудача. Эффект немонотонности вычислений логических программ с оператором отрицания.
10. Встроенные предикаты и функции. Операционная семантика встроенных средств.
11. Теорема о вычислительной универсальности чистого Пролога. Теорема Чёрча о неразрешимости логики предикатов первого порядка.

#### Неклассические прикладные логики

1. Интуиционистская логика. Модели Крипке для интуиционистской логики. Примеры интуиционистски общезначимых и необщезначимых формул. Модальные логики. Модели Крипке для модальных логик. Эпистемические логики. Темпоральные логики.
2. Проблема верификации последовательных программ. Операционная семантика типовых программных конструкций. Предусловие и постусловие. Частичная корректность программ. Тройки Хоара и их содержательный смысл. Правила вывода в логике Хоара для доказательства частичной корректности последовательных программ.
3. Моделирование программ системами переходов. Темпоральная логика высказываний линейного времени (LTL): синтаксис и семантика. Применение темпоральных логик для спецификации поведения реагирующих программных систем.
4. Задача проверки выполнимости формул LTL на конечной модели. Равносильные преобразования формул LTL. Табличный алгоритм проверки выполнимости формул LTL на конечной модели: основные этапы.

Экзаменационная контрольная работа состоит из 5 задач, проверяющих навыки использования методов математической логики и логического программирования для решения практических задач, 5 вопросов, проверяющих знание основных понятий и математических утверждений, и 4 задач, проверяющих умение применять понятия и методы математической логики для решения теоретических задач

**Пример контрольной экзаменационной работы.**

**Задача 0.** Слово – это конечный непустой список букв фиксированного конечного алфавита. Текст – это конечный непустой список слов. Построить логическую программу, которая для заданного текста **L** вычисляет одно из слов **X**, имеющее наибольшую длину среди всех тех слов текста, которые встречаются в этом тексте наиболее часто. Запрос к программе должен иметь вид ?**G(L,X)**.

**Задача 1.** Используя константные, функциональные и предикатные символы алфавита (см. Приложение 2), построить замкнутую формулу логики предикатов, соответствующую следующему утверждению.

“Каковы бы ни были две последовательности действительных чисел, если одна из них сходится к нулю, а другая ограничена, то и произведение этих последовательностей сходится к нулю”

**Задача 2.** Для заданной формулы ϕ выяснить, применяя метод семантических таблиц, является ли эта формула общезначимой.

**(∀x R(x) &∃y (P(y) →¬R(y))) →¬∀x P(x)**

**Задача 3.** Для заданной системы дизъюнктов **S** выяснить, применяя метод резолюций, является ли система **S**противоречивой или нет.

**S={ Q(v,v); ¬Q(v, u) ∨¬P(u, v); Q(u,v) ∨¬P(a, u) ∨ R(v,v); ¬R(a,u) ∨ Q(v, u); P(v, f(u)) }**

**Задача 4.** Для заданного запроса **G=? A(x), B(x)**  к заданной логической программе построить на основе стандартной стратегии вычислений (с использованием операторов отсечения и отрицания) дерево SLD-резолютивных вычислений и определить множество вычислимых ответов.

**: A(b) ← ;**

**A(a) ← not(B(b)), !;**

**A(f(b)) ← B(b), !;**

**A(c) ←not(B(c)), !, A(b); B(f(x)) ← A(x);**

**B(c) ←A(a), !, B(b);B(b) ← A(b);**

**B(a) ← ;**

**Вопрос 5.** Привести определение интерпретации для динамической логики.

**Вопрос 6.** Что означает корректность резолютивного вывода ?

**Вопрос 7.** Что такое множество успехов хорновской логической программы?

**Вопрос 8.** Привести пример полной стратегии построения SLD-вычислений хорновской логической программы?

**Вопрос 9.** Известно, что множества замкнутых формул **Γ** и **Δ** не имеют ни одной общей модели, не будучи при этом противоречивыми. Какие из приведенных утверждений справедливы и почему?

1. Семантическая таблица **〈Γ,Δ〉**не имеет успешного табличного вывода, потому что...
2. Семантическая таблица **〈Γ,Δ〉**не является выполнимой, потому что...
3. Семантическая таблица **〈∅,Γ∪Δ〉** имеет успешный табличный вывод, потому что...
4. Не существует формулы, которая являлась бы логическим следствием как множества формул **Γ,** так и множества формул **Δ**, потому что...
5. Все приведенные выше утверждения неверны.

**Вопрос 10.** Известно, что в семантическом дереве для семейства дизъюнктов **S** на каждом ярусе имеется непустое множество опровергающих узлов. Какие из приведенных ниже утверждений наверняка неверны и почему?

1. Семейство дизъюнктов **S** имеет эрбрановскую модель.
2. Формула **∀xP(x) →∃x¬P(x)** не является логическим следствием множества дизъюнктов **S**.
3. Ни в каком семантическом дереве для семейства дизъюнктов **S** нельзя выделить бесконечное множество опровергающих узлов.
4. Справедливость или несправедливость всех приведенных выше утверждений зависит конкретного множества **S**.

**Вопрос 11.** Пусть  **-** хорновская логическая программа, **T** - оператор непосредственного логического следования для , и **I′**, **I″** - эрбрановские интерпретации для программы . Какие из приведенных ниже утверждений справедливы и почему?

1. Всегда справедливо соотношение **T (I′∪I″) ⊂T ( I′ ) ∪T ( I″),** потому что....
2. Всегда справедливо соотношение **T (I′∪I″) ⊃T ( I′ ) ∪T ( I″),** потому что....
3. Множества **T (I′∪I″)** и **T ( I′ ) ∪T ( I″)** несравнимы**,** потому что...
4. Справедливость указанных выше соотношений между множествами **T (I′∪I″)** и **T ( I′ ) ∪T ( I″)** зависит от конкретных интерпретаций **I′**и **I″** , потому что ...

**Вопрос 12.** Известно, что запрос **?P(x)**  кхорновской логической программе не имеет успешных вычислений. Каким может быть ответ на запрос **? not(P(c))** к логической программе ? Выберите из предложенных вариантов ответа на этот вопрос правильные и обоснуйте их.

1. Ответ на запрос **? not(P(c))** будет положительный независимо от программы , потому что....
2. Ответ на запрос **? not(P(c))** будет отрицательный независимо от программы , потому что....
3. Ответ на запрос **? not(P(c))** может быть как положительным, так и отрицательным, в зависимости от программы , потому что....
4. На запрос **?not(P(c))** может быть вообще не получено никакого ответа, потому что....

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Распределенные системы**

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

**Вопросы к экзамену**

1. Какие аппаратные механизмы необходимы для организации мультипрограммного режима? Как обеспечить мультипрограммный режим без этих механизмов? Как обеспечить, если отсутствует только один из этих механизмов?

2. Имеется механизм двоичных семафоров. Опираясь на него, реализуйте P-операцию и V-операцию для общего (считающего) семафора. Активное ожидание освобождения семафора не допускается.

3. Имеется команда TSL и команда объявления прерывания указанному процессору. Опираясь на них, реализуйте на мультипроцессоре P-операцию и V-операцию для двоичного семафора. Активное ожидание освобождения семафора не допускается.

4. Имеется механизм двоичных семафоров. Опираясь на него, реализуйте операторы POST(имя переменной-события) и WAIT(имя переменной-события). Активное ожидание события не допускается. Оцените, во сколько раз нижеприведенный алгоритм метода последовательной верхней релаксации можно выполнить быстрее, чем последовательный, если число процессоров мультипроцессора = N, время выполнения одного оператора присваивания (A[i][j]=....) равно 1, временами выполнения остальных операторов можно пренебречь.

float A[ L1 ][ L2 ];

semaphore s[ L1 ][ L2 ]; /\* массив двоичных семафоров с нулевым начальным значением \*/

for ( j = 0; j < L2; j++) { post( s[ 0 ][ j ]) }

parfor ( i = 1; i < L1-1; i++)

for ( j = 1; j < L2-1; j++)

{ wait( s[ i-1 ][ j ]);

A[ i ][ j ] = (A[ i-1 ][ j ] + A[ i+1][ j ] + A[ i ][ j-1 ] + A[ i ][ j+1 ]) / 4;

post( s[ i ][ j ]);

}

5. Имеется механизм двоичных семафоров. Опираясь на него, реализуйте задачу читателей и писателей (алгоритмы предоставления прав доступа процессам-читателям и процессам-писателям):

Процесс-писатель должен получать исключительный (монопольный) доступ к базе данных (других писателей или каких-либо читателей быть не должно). Произвольное число процессов-читателей может работать одновременно, но любой читатель может получить доступ только при отсутствии работающих писателей.

Запросы на доступ должны удовлетворяться "справедливо" - в порядке их поступления (можно исходить из "справедливости" удовлетворения запросов на двоичные семафоры).

6. Какие модели консистентности памяти удовлетворяют алгоритму Деккера (алгоритм без каких-либо изменений будет работать правильно), а какие нет? Объясните ответ.

7 Какие модели консистентности памяти удовлетворяют алгоритму Петерсона (алгоритм без каких-либо изменений будет работать правильно), а какие нет? Объясните ответ.

8. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо выполнить операцию передачи сообщения длиной N байт всем процессам от одного (MPI\_BCAST) - процесса с координатами (0,0). Сколько времени потребуется для этого, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

9. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо выполнить операцию сбора данных (длиной один байт) от всех процессов для одного (MPI\_GATHER) - процесса с координатами (0,0). Сколько времени потребуется для этого, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

10. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо выполнить операцию рассылки данных (длиной один байт) всем процессам от одного (MPI\_SCATTER) - процесса с координатами (0,0). Сколько времени потребуется для этого, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

11. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо выполнить операцию нахождения максимума среди 16 чисел (каждый процесс имеет свое число). Сколько времени потребуется для получения всеми максимального числа, если все процессы выдали эту операцию редукции одновременно. А сколько времени потребуется для нахождения максимума среди 64 чисел в матрице 8\*8? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

12. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо переслать очень длинное сообщение (длиной L байт) из узла с координатами (0,0) в узел с координатами (3,3). Сколько времени потребуется для этого, если передача сообщений точка-точка выполняется в буферизуемом режиме MPI? А сколько времени потребуется при использовании синхронного режима и режима готовности? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

13. В транспьютерной матрице размером 4\*4, в каждом узле которой находится один процесс, необходимо переслать сообщение длиной L байт из узла с координатами (0,0) в узел с координатами (3,3). Сколько времени потребуется для этого при использовании а) неблокирующих и б) блокирующих операций MPI? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

14. Все 16 процессов, находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на вход в критическую секцию. Сколько времени потребуется для прохождения всеми критических секций, если используется древовидный маркерный алгоритм (маркером владеет нулевой процесс). Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

15. Все 16 процессов, находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на вход в критическую секцию. Сколько времени потребуется для прохождения всеми критических секций, если используется децентрализованный алгоритм с временными метками. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

16. Все 16 процессов, находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на вход в критическую секцию. Сколько времени потребуется для прохождения всеми критических секций, если используется широковещательный маркерный алгоритм (маркером владеет нулевой процесс). Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

17. 15 процессов, находящихся в узлах транспьютерной матрицы размером 4\*4, одновременно выдали запрос на вход в критическую секцию. Сколько времени потребуется для прохождения всеми критических секций, если используется централизованный алгоритм (координатор расположен в узле 0,0)? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

18. Сколько времени потребует выбор координатора среди 16 процессов, находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), если используется алгоритм "задиры"? "Задира" расположен в узле с координатами (0,0) и имеет уникальный номер 0. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

19. Сколько времени потребует выбор координатора среди 16 процессов, находящихся в узлах транспьютерной матрицы размером 4\*4, если используется круговой алгоритм? Время старта равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Процессорные операции, включая чтение из памяти и запись в память считаются бесконечно быстрыми.

20. Какие принципиальные решения приходится принимать при обеспечении файлового сервиса?

21. Интерфейс сервера директорий.

22. Семантика разделения файлов.

23. Серверы с состоянием и без состояния. Достоинства и недостатки.

24. Алгоритмы обеспечения консистентности кэшей в распределенных файловых системах.

25. Способы организации размножения файлов и коррекции копий.

26. Последовательная консистентность памяти и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует модификация 10 различных переменных 10-ю процессами (каждый процесс модифицирует одну переменную), находящимися на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания) и одновременно выдавшими запрос на модификацию. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

27. Причинная консистентность памяти и алгоритм ее реализации в DSM с полным размножением при условии, что никаких сведений от компилятора о причинной зависимости операций записи не имеется. Сколько времени потребует модификация 10 различных переменных, если все 10 процессов (каждый процесс модифицирует одну переменную), находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на модификацию своей переменной. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

28. Процессорная консистентность памяти и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует модификация 10 различных переменных, если все 10 процессов (каждый процесс модифицирует одну переменную), находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на модификацию своей переменной. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

29. PRAM консистентность памяти и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует 3-кратная модификация 10 различных переменных, если все 10 процессов (каждый процесс 3 раза модифицирует одну переменную), находящихся на разных ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания), одновременно выдали запрос на модификацию. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

30. Слабая консистентность памяти и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует модификация одним процессом 10 обычных переменных, а затем 3-х различных синхронизационных переменных, если DSM реализована на 10 ЭВМ сети с шинной организацией (с аппаратными возможностями широковещания). Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

31. Консистентность памяти по выходу и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует трехкратное выполнение каждым процессом критической секции, в которой модифицируются 10 переменных, если DSM реализована на 10 ЭВМ сети с шинной организацией (с аппаратными возможностями широковещания). Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

32. Консистентность памяти по входу и алгоритм ее реализации в DSM с полным размножением. Сколько времени потребует трехкратное выполнение критической секции и модификация в ней 10 переменных каждым процессом, если DSM реализована на 10 ЭВМ сети с шинной организацией (с аппаратными возможностями широковещания). Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память считаются бесконечно быстрыми.

33. Алгоритм надежных и неделимых широковещательных рассылок сообщений. Дайте оценку времени выполнения одной операции рассылки для сети из 10 ЭВМ с шинной организацией (без аппаратных возможностей широковещания), если отправитель сломался после посылки 5-го сообщения. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

34. Протоколы голосования. Алгоритмы и применение. Дайте оценку времени выполнения одним процессом 2-х операций записи и 10 операций чтения N байтов информации с файлом, расположенным (размноженным) на остальных 10 ЭВМ сети с шинной организацией (без аппаратных возможностей широковещания). Определите оптимальные значения кворума чтения и кворума записи для N=300. Время старта (время "разгона" после получения доступа к шине для передачи) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса (при одновременных запросах - в порядке номеров ЭВМ). Операции с файлами и процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

35. Консистентное и строго консистентные множества контрольных точек. Дайте оценку накладных расходов на синхронную фиксацию строго консистентного множества контрольных точек для сети из 10 ЭВМ с шинной организацией (без аппаратных возможностей широковещания), если накладные расходы на синхронную фиксацию консистентного множества равны Т1. Время старта (время "разгона" после получения доступа к шине для передачи сообщения) равно 100, время передачи байта равно 1 (Ts=100,Tb=1). Доступ к шине ЭВМ получают последовательно в порядке выдачи запроса на передачу (при одновременных запросах - в порядке номеров ЭВМ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Языки программирования**

**Типовые контрольные задания или иные материалы для проведения промежуточной аттестации**

|  |
| --- |
| **Экзаменационная работа (письменный экзамен)** |
| **Вариант 1** |
| 1. Замените знаки вопроса (?????) в приведенной ниже программе на языке С# так, чтобы в стандартный вывод выдавалась сумма квадратов эдементов списка s. Никакой код куда-либо еще добавлять нельзя. Примечание: метод ForEach списка имеет прототип List<int>.ForEach(Action<int> action) и выполняет действие action для каждого элемента списка.  List<int> l = new List<int>() { 1, 2, 3, 5, 7, 9 };  int s = 0;  l.ForEach(?????);  System.Console.WriteLine(s);  2. Пусть s описана как переменная строкового типа (string), и в программе есть присваивание s= ”кафе” (все буквы - кириллические). Какова будет длина этой строки в языках C++(s.size()), Java (s.length), Go (len(s)), Swift (s.count)? Для каждого языка дайте ответ и объясните различия (если они есть).  3. Объясните, допустимо ли присваивание aobj = afoo в следующем фрагменте программы на Java. Если допустимо, то какие проблемы с aobj и afoo могут возникнуть в дальнейшем при выполнении программы?  class Foo { public void foo () {} }  class Bar { public void bar () {} }  Object [] aobj; Foo [] afoo = new Foo[10];  aobj = afoo;  4. Написать на языке С++ шаблонную функцию int MakeWord(), которая возвращает целое число, являющееся побитовой суммой (операция суммирования — побитовое «или» | ) аргументов шаблона. Аргументы шаблона должны быть целочисленными.  Например std::cout << MakeWord<1,2>(); должно выдать 3, а  std::cout << MakeWord<2,4,9>(); должно выдать 15.  5. Являются ли реализации обобщенных классов на языке Java более эффективными, чем их необобщенные аналоги. Например, будет ли реализация обобщенного класса Stack<int> более эффективной, чем реализация класса Stack из Object?  6. Напишите какой-нибудь пример генератора (но не сопрограммы) на языке Python. Чем сопрограммы языка Python отличаются от простых генераторов в этом языке?  7. Дайте определение вырезки из массива. Приведите примеры вырезок в языке Python и Go. Чем они отличаются?  8. В приведенной ниже программе на языке Go есть ошибка, которая проявляется во время выполнения. В чем она заключается, и как ее исправить (указание — надо переставить одну строку в другое место)? Что будет выдано после исправления?  package main; import "fmt"  func fibonacci(c, quit chan int) {  x, y := 0, 1  for {  select {  case c <- x:  x, y = y, x+y  case <-quit:  fmt.Println("quit")  return  }  }  }  func main() {  c := make(chan int)  quit := make(chan int)  go func() {  for i := 0; i < 10; i++ {  fmt.Print(←c);fmt.Print(" ")  }  }()  quit <- 0  fibonacci(c, quit)  } |
| Вариант 2 |
| 1. Замените знаки вопроса (?????) в приведенной ниже программе на языке С# так, чтобы в стандартный вывод выдавалась сумма квадратов эдементов списка s. Никакой код куда-либо еще добавлять нельзя. Примечание: метод ForEach списка имеет прототип List<int>.ForEach(Action<int> action) и выполняет действие action для каждого элемента списка.  List<int> l = new List<int>() { 1, 2, 3, 5, 7, 9 };  int s = 0;  l.ForEach(?????);  System.Console.WriteLine(s);  2. Пусть s описана как переменная строкового типа (string), и в программе есть присваивание s= ”кафе” (все буквы - кириллические). Какова будет длина этой строки в языках C++(s.size()), Java (s.length), Go (len(s)), Swift (s.count)? Для каждого языка дайте ответ и объясните различия (если они есть).  3. Объясните, допустимо ли присваивание aobj = afoo в следующем фрагменте программы на Java. Если допустимо, то какие проблемы с aobj и afoo могут возникнуть в дальнейшем при выполнении программы?  class Foo { public void foo () {} }  class Bar { public void bar () {} }  Object [] aobj; Foo [] afoo = new Foo[10];  aobj = afoo;  4. Написать на языке С++ шаблонную функцию int MakeWord(), которая возвращает целое число, являющееся побитовой суммой (операция суммирования — побитовое «или» | ) аргументов шаблона. Аргументы шаблона должны быть целочисленными.  Например std::cout << MakeWord<1,2>(); должно выдать 3, а  std::cout << MakeWord<2,4,9>(); должно выдать 15.  5. Являются ли реализации обобщенных классов на языке Java более эффективными, чем их необобщенные аналоги. Например, будет ли реализация обобщенного класса Stack<int> более эффективной, чем реализация класса Stack из Object?  6. Напишите какой-нибудь пример генератора (но не сопрограммы) на языке Python. Чем сопрограммы языка Python отличаются от простых генераторов в этом языке?  7. Дайте определение вырезки из массива. Приведите примеры вырезок в языке Python и Go. Чем они отличаются?  8. В приведенной ниже программе на языке Go есть ошибка, которая проявляется во время выполнения. В чем она заключается, и как ее исправить (указание — надо переставить одну строку в другое место)? Что будет выдано после исправления?  package main; import "fmt"  func fibonacci(c, quit chan int) {  x, y := 0, 1  for {  select {  case c <- x:  x, y = y, x+y  case <-quit:  fmt.Println("quit")  return  }  }  }  func main() {  c := make(chan int)  quit := make(chan int)  go func() {  for i := 0; i < 10; i++ {  fmt.Print(←c);fmt.Print(" ")  }  }()  quit <- 0  fibonacci(c, quit)  } |

**Вопросы к экзамену**

1. Основные позиции при рассмотрении ЯП. Схема рассмотрения ЯП: базис, средства развития и средства защиты.
2. Основные понятия языков программирования: данные, операции и связывание.
3. Понятие о виртуальной машине языка
4. Классификация базисных скалярных типов данных
5. Арифметические типы данных: целые, плавающие, фиксированные. Проблемы представления чисел и способы их решения в ЯП.
6. Символьные и логические типы данных.
7. Порядковые типы: диапазоны и перечисления. Особенности реализации перечислений в современных ЯП.
8. Ссылки и указатели. Управление памятью. Автоматическая сборка мусора. Объектно-референциальная модель в современных ЯП.
9. Составные типы данных. Массивы и их особенности в современных ЯП. Динамические массивы и списки. Вырезки из массивов.
10. Записи. Недостатки системы типов в традиционных ЯП. Объединения как средство преодоления этих недостатков. Проблемы, связанные с объединениями.
11. Ассоциативные массивы и записи.
12. Понятие о структурном программировании. Разновидности управляющих конструкций в современных языках программирования. Оператор перехода, связанные с им проблемы и способы их решения в современных ЯП.
13. Условные операторы и многовариантные развилки. Циклы. Особенности реализации циклов-итераторов в современных ЯП
14. Статический полиморфизм и перегрузка имен подпрограмм.
15. Функции и функциональные типы. Функции как объекты первого порядка. Понятие замыкания и анонимные функции. Лямбда-функции и замыкания в современных ЯП (C#, Python, C++, Java, JavaScript).
16. Сопрограммы и подпрограммы. Сопрограммы в современных ЯП (на примере итераторов в C# и генераторов Python). Го-программы в языке Go.
17. Понятие логического модуля. Использование модулей для определения новых типов данных. Особенности понятия модуля в современных ЯП.
18. Импорт и экспорт имен. Видимость имен: непосредственная и потенциальная. Управление видимостью. Области видимости и пространства имен. Модульность и технология программирования: проектирование «сверху-вниз» и «снизу-вверх».
19. Понятие класса. Класс как тип данных. Члены класса: функции, данные. Статические и нестатические члены. Члены - вложенные классы. Статические и нестатические классы. Классы и области видимости.
20. Понятие специальных функций-членов. Проблема инициализации объектов и способы ее решения. Конструкторы, деструкторы, операторы using и try-finally.
21. Преобразование типов и классы. Явные и неявные преобразования. Управление преобразованиями в современных ЯП: проблемы и способы их решения.
22. Классы и перегрузка имен. Перегрузка встроенных знаков операций. Итераторы и индексаторы. Классы и стандартные библиотеки. Встроенные классы стандартной библиотеки.
23. Понятие инкапсуляции. Понятие абстрактного типа данных (АТД) и его достоинства. Инкапсуляция и логические модули.
24. Реализация АТД в модульных языках программирования (Ада, Оберон, Модула). Инкапсуляция и классы. Управление видимостью и управление доступом.
25. Пространства имен и инкапсуляция. Реализация АТД с помощью понятия класса. Принцип разделения определения, реализации и использования (РОРИ). Эволюция принципа РОРИ в современных ЯП.
26. Модульность и раздельная трансляция.
27. Исключительные ситуации и обработка ошибок.
28. Иерархии типов, статические и динамические типы в объектно-ориентированных ЯП.
29. Наследование и области видимости имен. Замещение, перегрузка и скрытие имен при наследовании. Наследование и инкапсуляция. Управление видимостью и доступом при наследовании. Запрещение наследования для классов и методов.
30. Наследование и специальные функции. Понятие о множественном наследовании.
31. Статическое и динамическое связывание методов. Динамический тип данных и динамическое связывание. Замещение функций и динамическое связывание. Особенности динамического связывания в современных ЯП. Достоинства и недостатки динамического связывания. Снятие динамического связывания.
32. Понятие о мультиметодах.
33. Понятие абстрактного класса (АК). Необходимость понятия АК при проектировании иерархий классов. Воплощение концепции АК в современных ЯП. Абстрактные классы и интерфейсы. Интерфейс как языковая конструкция.
34. Связь интерфейсов и других языковых конструкций (итераторов, сохраняемых объектов и т.д.). Интерфейсы и иерархии классов. Множественное наследование интерфейсов. Реализация интерфейсов и ее особенности современных ЯП. Явная и неявная реализация интерфейсов.
35. Функциональные интерфейсы и функциональные типы данных.
36. Понятие о динамической идентификации типа
37. Понятие о статической параметризации и родовых объектах. Достоинства статической параметризации. Статическая параметризация и ООП. Родовые модули и подпрограммы в языке Ада.
38. Механизм шаблонов в языке Си++. Шаблоны-классы и шаблоны-функции. Параметры шаблонов. Вывод параметров шаблонов. Генерация кода по шаблонам. Полная и частичная специализация шаблонов. Обобщенное программирование на языке Си++. Шаблоны и функциональное программирование.
39. Особенности обобщенного программирования в языках С# и Java.
40. Ковариация и обобщенные конструкции. Реализация ковариации в языках С# и Java.

**Типовые задания для экзамена**

1. Написать на конкретном языке программирования объявления, функционально эквивалентные приведенным объявлениям на другом языке.
2. Найти возможные ошибки в приведенном фрагменте программы, если есть, то исправить их и объяснить, что будет выдано в стандартный канал вывода.
3. Дать определение и примеры какой-либо конструкции (исключительной ситуации, абстрактного класса, виртуальной функции, динамического связывания и т.д.).
4. Перечислить языки (из заданного списка), в которых есть некоторое понятие. Для каждого языка дать пример объявления и использования данного понятия.
5. Сравнить реализации некоторого понятия в двух конкретных языках программирования с некоторой точки зрения (эффективности реализации, удобства программирования, надежности, сложности использования, простоты понимания и т.п.
6. В заданном фрагменте программы внести (небольшие) изменения так, чтобы фрагмент выполнил определенную задачу, например, вывод заданной строки в стандартный канал вывода.
7. Переписать на некотором языке программирования заданный фрагмент программы на другом языке программирования.
8. Дать определение и объяснить назначение какой-либо конструкции в некотором языке программирования. Есть аналоги этой конструкции в языках программирования из приведенного списка. Если есть, то дать короткий пример объявления и использования для каждого языка.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Машинное обучение и анализ данных**

**Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.**

**Вопросы и задания по лекциям**

**Тест 1 (с указанием правильных ответов)**

Случайная величина принимает значение из отрезка [0,1], её плотность линейная функция на этом отрезке, в нуле обращается в ноль. Чему равно матожидание с.в.? Совет: здесь и ниже, кроме аналитического решения напишите на Python программу для оценки названных параметров.

0

1/2

\* 2/3

3/4

4/5

1

Чему равна мода этой с.в.?

0

1/2

2/3

3/4

4/5

\* 1

нет правильного ответа

Чему равна медиана этой с.в.?

0

1/2

2/3

3/4

4/5

1

\* нет правильного ответа

Чему равна дисперсия этой с.в.?

1/18

Предположим, что в задаче бинарной классификации с одним признаком объекты класса 1 распределены так, как описано выше, а объекты класса 0 распределены равномерно на отрезке [0,1]. Оба класса равновероятны. Какой оптимальный порог для отнесения объектов к классу 1 (выше него считаем, что они из класса 1), если оба класса равновероятны?

1/3

\* 1/2

2/3

3/4

4/5

Что лучше использовать для определения монотонной зависимости между переменными?

Корреляционный коэффициент Пирсона

\* Коэффициент корреляции Спирмена

оценку ММП (MLE)

Парзеновский подход используется для параметрического оценивания плотности

\* непараметрического оценивания плотности

оценки плотности смеси распределений

Чему равна оценка плотности в точке 1.5 для выборки {1, 2, 3} парзеновским методом с треугольным ядром (радиус основания ядра 1)?

0

1/6

\* 1/3

2/3

1

нет правильного ответа

Пусть случайная величина равна сумме двух равномерно распределённых величин на отрезке [0,1]. Как выглядит её плотность распределения?

«колокольчик»

\* треугольник

трапеция

прямоугольник

К чему стремится угол между соседними диагоналями n-мерного гиперкуба при увеличении размерности?

\* 0

pi/4

pi/2

нет правильного ответа

Запишите сумму квадратов сингулярных чисел для матрицы [[0,1], [0,1]].

2

При минимизации функции x^2 методом градиентного спуска с темпом 1.0 и начальной точкой 1.0, какая будет оценка argmin после 4й итерации?

1

Выберете верные фразы:

+ в SGD случайный порядок объектов

+ SGD может использоваться при онлайн-оптимизации (обучении)

+ SGD может использоваться для минимизации суммы ошибок на объектах моделей классификации / регрессии

SGD – метод оптимизации второго порядка

SGD – это метод условной оптимизации

Решите задачу условной оптимизации: x^2 + y^2 -> min при условии x+2y>=1 (ответ - значение функции в точке минимума).

0.2

**Тест 2. Тема «метрические и линейные алгоритмы»**

Дана обучающая выборка в однопризнаковой задаче бинарной классификации. Объекты первого класса – {0, 4, 4.5}, второго – {3, 7, 8}. Какая оценка точности алгоритма 1NN методом LOO(leave-one-out)? Ответ умножьте на 6.

4

В предыдущей задаче оцените точность алгоритма ближайшего центроида.

4

Пусть в однородной области метрического пространства объектов бинарной задачи классификации вероятность того, что случайно выбранный объект принадлежит классу 1, равна 1/4. Чему равна ошибка алгоритма ближайшего соседа в этой области?

0.375

Что такое ленивый алгоритм (Lazy learner)?

- медленно классифицирует

+ не формирует модель описания данных во время обучения

- синоним для алгоритма kNN

- который по умолчанию не использует метод предсказания меток (predict)

Задачу регрессии можно решать с помощью (выбрать все верные варианты)

+ kNN

+ метода Надарая-Ватсона

- метода логистической регрессии

Какие расстояния численно наибольшие для пары точек (1,1) и (2,2):

- Евклидово

- Чебышева

+ Манхэттенское

Пусть даны векторы (1,1,2,2,3,3), (1, 4). Чему равно расстояние DTW?

4

Решите матричное уравнение Xw=y, W=[[1, 1], [1, 2], [1, 3]], y=[1,2,1] с помощью минимизации невязки. В ответ запишите скалярное произведение вектора w и вектора (3, -1).

4

В регуляризации по Тихонову:

+ к оптимизируемому функционалу добавляется специальное слагаемое

- вводится ограничение на норму вектора-решения

- оптимизируемый функционал оценивается сверху

В гребневой регрессии (выберите верные утверждения)

+ Используется регуляризация по Тихонову

+ матрица Грамма в псевдообратной становится невырожденной

– происходит гарантированное зануление элементов вектора решения

– происходит гарантированная селекция признаков

С помощью персептронного алгоритма решите систему уравнений a+b>0, 3a-b>0, a-b<0. Начальное приближение (a,b) = (0,0), неравенства просматриваются слева направо. В ответ запишите значение b/a.

2

Выберите верные фразы:

+ Для селекции признаков можно использовать LASSO

+ Устойчивая регрессия (Robust Regression) хороша в задаче с выбросами

- В логистической регрессии минимизируют среднее квадратичное отклонение ответов модели от истинных меток

– решение линейной регрессии робастно (устойчиво к выбросам)

**Тест 3. Тема «Деревья и контроль».**

Что особенного в деревьях решений вида «oblique decision trees»?

- ограничение на глубину

- использование предварительной обрезки (pre-pruning)

+ специальный предикат ветвления

- возможность распараллеливания при построении

Почему при построении дерева используют рекурсивную жадную стратегию?

- алгоритма оптимизации не существует

+ задача построения оптимального дерева очень сложна (в одном частном случае это NP-полная проблема)

- это, как правило, быстрее градиентного спуска

Рассмотрим 10 объектов, если их упорядочить по первому признаку, то их метки будут чередоваться следующим образом: [0,0,1,0,1,0,1,1,1,0]. Найдите максимальное значение критерия расщепления Missclassification criteria.

0.2

В предыдущей задаче найдите максимальное значение критерия расщепления Gini (основанного на мере неоднородности Gini).

0.125

В предыдущей задаче найдите максимальное значение энтропийного критерия расщепления. Ответ округлите до первой цифры после запятой (например, 0.1).

0.2

Отметьте верные утверждения:

+ Обрезку (post-pruning) используют крайне редко

+ Деревья особенно эффективны в ансамбле

+ Деревья – нестабильный (неустойчивый) алгоритм

- Деревья часто используют для экстраполяции

Что такое C5.0?

+ алгоритм построения деревьев решений

- специальный критерий расщепления

- запатентованный способ обрезки деревьев

- метод регуляризации при построении деревьев

Пусть есть категориальный признак со значениями [A, A, B, B, C, C, D, D], с целевыми значениями [2, 0, 2, 8, 3, 5, 0, 4]. Какое будет расщепление со стандартным критерием, использующим дисперсию?

- A, B | C, D

- A, C | B, D

+ A, D | B,C

- A | C, B, D

- A, B, C | D

Что из перечисленного можно использовать для выбора модели (Model Selection):

+ бутстреп

- регуляризацию

+ разбиение на фолды

В какой из перечисленных функций библиотеки scikit-learn схема контроля гарантирует определённую пропорцию объектов разных классов?

- ShuffleSplit

- GroupShuffleSplit

+ StratifiedShuffleSplit

- KFold

- GroupKFold

+ StratifiedKFold

- PredefinedSplit

Пусть по транзакциям пользователя мы предсказываем его покупательскую активность в следующем методе. Какие способы контроля следует выбрать (в данных – статистика по всем клиентам банка)?

- LOOCV

+ out of time

+ out of sample

Кривые обучения (Learning Curves) могут

+ оценить достаточность объёма выборки

+ оценить, не слишком ли простая модель использована

- подобрать оптимальные значения всех параметров

+ оценить переобучение алгоритма

Пусть дана выборка целевых значений: 1, 3, 2, 1 (упорядочено по времени получения меток). Используется модель константных алгоритмов (ответ равен среднему по всем меткам обучения). Функция ошибки – MAE (средний модуль отклонения). Чему равна средняя ошибка при контроле по времени (Out-of-time)?

1

В предыдущей задаче – чему будет равна ошибка LOOCV (контроля по одному)?

1

**Тест 4. Тема «Ансамбли».**

Чему равна ошибка комитета большинства над тремя алгоритмами бинарной классификации с вероятностями ошибки 0.3, 0.2, 0.1 (действует сильное предположение о независимости ответов алгоритмов)?

0.0098

В какой модели разнообразие базовых алгоритмов повышается за счёт варьирования обучающей выборки?

+ бэгинг (Bagging)

+ метод случайных подпространств (Random Subspaces)

+ случайные леса (Random Forests)

- нейросети

В какой модели производится перекодировка целевого признака?

- комитеты

+ ECOC

- стекинг

- бустинг

В какой модели применяется взятие бустреп-подвыборок?

+ бэгинг (Bagging)

+ случайные леса (RF)

- стекинг

- Feature-Weighted Linear Stacking

Для чего можно использовать OOB-предсказания?

+ для оценки качества модели

+ для реализации стекинга

- для регуляризации

- для кодирования целевого вектора

+ для вычисления рейтинга (важности) признаков

Какие из перечисленных ниже моделей являются последовательными ансамблями (Sequential ensembles)?

- бэгинг (Bagging)

+ Adaboost

+ градиентный бустинг

- нейронные сети

- случайные леса (RF)

Пусть случайные величины одинаково распределены (среднее равно 1, дисперсия – 2), корреляция между любой парой величин равна 0.1. К чему стремится среднее арифметическое этих величин при увеличении числа наблюдений (т.е. увеличении числа этих величин)?

0.2

В каких моделях увеличение числа базовых алгоритмов не приводит к переобучению?

- стекинг

- бустинг

+ случайные леса

Что происходит при увеличении глубины деревьев (считаем, что в ансамблях достаточное число деревьев)?

+ как правило, увеличивается качество случайного леса на тесте

+ как правило, увеличивается качество случайного леса на обучении

- как правило, увеличивается качество бустинга над деревьями на тесте

+ как правило, увеличивается качество бустинга на обучении

В одном из подходов к оценке важности признаков используют перестановку значений. Почему именно перестановку (а, например, не анализ качества алгоритмов без соответствующего признака)?

+ это не меняет распределение по признаку

+ это позволяет не переучивать модель

- это гарантирует стабильность модели

- это гарантирует такое же распределение ответов модели

Чем экстремальные леса (Extreme Random Trees) отличаются от случайных (Random Forest)?

- не используем критерии расщепления (типа gini и энтропийного)

+ быстрее построение ансамбля

- нужен градиент функции ошибки

+ качество, как правило, чуть хуже

Что используется в продвинутых методах реализации градиентного бустинга (как в библиотеке XGBoost)?

+ принцип минимальной длины (MDL)

+ вторые производные функции ошибки

- автоматический выбор ключевых параметров, например learning\_rate

На какие слагаемые раскладывается квадратичная ошибка регрессора (матожидание квадрата разности прогноза и истинного значения)?

+ шум (noise)

- квадрат шума

+ разброс (variance)

- квадрат разброса

- смещение (bias)

+ квадрат смещения

При повышении числа соседей k метода kNN...

- увеличивается сложность модели

+ увеличивается качество на обучении

- увеличивается качество на контроле

- увеличивается разброс (variance)

+ увеличивается стабильность

Что из перечисленного приводит к уменьшению переобучения?

+ аугментация

+ регуляризация

+ увеличение объёма выборки

**Пример практического задания**

Задача машинного обучения с реальными данными, выложенная на https://inclass.kaggle.com/c/dayofweek/

***Описание***

Для 300000 пользователей дана статистика посещений ресурса за 1099 дней. Необходимо предсказать день недели следующего визита.

***Метрика качества***

Используется простой процент правильных ответов. Например,

performance([1,2,2,7], [3,2,2,7]) = 0.75

***Формат ответа***

В загружаемом файле по строкам перечислены идентификаторы пользователей и номера дней их первых визитов по версии вашего алгоритма:

id,nextvisit  
1, 7

***Данные***

В файле train.csv перечислены даты визитов пользователей. Каждая строка - информация по одному пользователю. Сначала идёт id, потом через пробел номера дней, когда были визиты. Нумерация идёт от некоторого фиксированного момента. Номера могут быть от 1 до 1099 (т.е. статистика охватывает период примерно 3 года). Первый день в нумерации - понедельник.

Необходимо предсказать день недели первого визита после 1099го дня, т.е. для каждого пользователя вычислить

0 - нет визита

1 - понедельник

2 - вторник

3 - среда

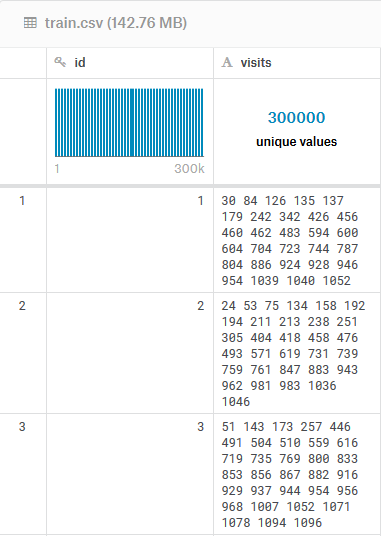
4 - четверг

5 - пятница

6 - суббота

7 - воскресенье

***Данные в просмотрщике платформы inclass.kaggle***



**Пройденные темы (варианты вопросов для возможного устного опроса)**

* Терминология: Наука о данных (Data Sciense), Статистика (Statistics), Искусственный интеллект (Artificial Intelligence), Анализ данных (Data Mining), Машинное обучение (Machine learning), Большие данные (Big Data)
* Обучение с учителем (с размеченными данными / метками): целевая функция, объект, метка, классификация, прогнозирование
* Пространство объектов, признаковое пространство, извлечение признаков, визуализация задач
* Функции ошибки, эмпирический риск, обучающая выборка, задачи оптимизации в обучении, обобщающая способность
* Модель алгоритмов, алгоритм, обучение, схема решения задачи машинного обучения
* Обучение без учителя / c неразмеченными данными, обучение c частично размеченными данными, трансдуктивное обучение
* Обучение с подкреплением, структурный вывод, активное обучение, онлайн-обучение, Transfer Learning, Multitask Learning, Feature Learning
* Математика в машинном обучении: бритва Оккама, теорема о бесплатном сыре, футбольный оракул, теория информации, проклятие размерности, сингулярное разложение матрицы (SVD), матричное дифференцирование
* Сведения из ТВиМС: задание распределений, средние и отклонения, условная плотность, маргинализация и обуславливание, точечное оценивание, оценка максимального правдоподобия, дивергенция Кульбака-Лейблера, ковариация и корреляция, нормальное распределение, центральная предельная теорема
* Оценка плотности: гистограммный подход, Парзеновский подход
* Оптимизация: методы безусловной оптимизации, нулевого порядка, первого порядка, второго порядка, метод градиентного спуска в машинном обучении, стационарные точки, метод Ньютона, квази-ньютоновские методы, оптимизация с ограничениями
* Градиентный спуск, наискорейший градиентный спуск, стохастический градиентный спуск, обучение: Пакетное, онлайн, по минибатчам
* Метрические алгоритмы (distance-based), ближайший центроид (Nearest centroid algorithm), подход, основанный на близости, kNN в задаче классификации / регрессии, обоснование 1NN, ленивые (Lazy) и нетерпеливые (Eager) алгоритмы
* Весовые обобщения kNN, регрессия Надарая-Ватсона
* Различные метрики: Минковского, Евклидова, Манхэттенская, Махалонобиса, Canberra distance, Хэмминга, косинусное, расстояние Джаккарда, DTW, Левенштейна, приложения метрического подхода: нечёткий матчинг таблиц, Ленкор, в DL, классификация текстов, эффективные методы поиска ближайших соседей
* Линейные методы: линейная регрессия, обобщённая линейная регрессия, проблема вырожденности матрицы, регуляризация, основные виды регуляризации, гребневая регрессия (Ridge Regression), LASSO (Least Absolute Selection and Shrinkage Operator), Elastic Net
* Селекция признаков, ошибка с весами, устойчивая регрессия (Robust Regression)
* Линейные скоринговые модели в задаче бинарной классификации, логистическая регрессия, Probit-регрессия, многоклассовая логистическая регрессия
* Линейный классификатор, персептрон, оценка функции ошибок через гладкую функцию
* Деревья решений (СART), предикаты / ветвления, птветы дерева, критерии расщепления в задачах классификации: Missclassification criteria, энтропийный, Джини, критерии остановки при построении деревьев, проблема переобучения для деревьев, подрезка (post-pruning), классические алгоритмы построения деревьев решений: ID3, C5.0
* Важности признаков, проблема пропусков (Missing Values), категориальные признаки, сравнение: деревья vs линейные модели
* Проблема контроля качества, выбора модели (Model Selection) в широком смысле, правила разбиения выборки, кривые обучения (Learning Curves)
* перебор параметров
* Отложенный контроль (held-out data, hold-out set), скользящий контроль (cross-validation), бутстреп (bootstrap), контроль по времени (оut-of-time-контроль), локальный контроль
* Ансамбли алгоритмов: примеры и обоснование, комитеты (голосование) / усреднение, бэгинг, кодировки / перекодировки ответов, ECOC
* Стекинг и блендинг, бустинг: AdaBoost, Forward stagewise additive modeling (FSAM), «Ручные методы», однородные ансамбли
* Случайный лес, его параметры, их настройка, бэггинг и OOB (out of bag), важность признаков, близость (Proximity) с помощью RF, Extreme Random Trees
* Градиентный бустинг над деревьями, его параметры, современные реализации, продвинутые методы оптимизации
* Рекомендательные системы, персонализация, онлайн и оффлайн рекомендации, рекомендация по контенту (content based methods), One-class recommendation, использование дополнительной информации, современные тренды в практике построения рекомендательных систем
* Коллаборативная фильтрация: GroupLens-алгоритм, SVD, SVD++, timeSVD++, адаптация SVD под социальные связи
* Факторизационная машина, факторизационная машина с полями (FFM – field-aware factorization machine)
* Простые методы рекомендаций: FPM – Frequent Pattern Mining, Deep Semantic Similarity Model (DSSM), контекст рекомендации, Knowledge-based Recommendations, важность объяснений (explanations)
* Сложность алгоритмов, переобучение, смещение и разброс: проблема обобщения, переобучение, недообучение, сложность алгоритмов, смещение и разброс, способы борьбы с переобучением

**Вариант теста для письменного зачёта (с указанием правильных ответов).**

Пусть случайная величина равна сумме двух независимых равномерно распределённых величин, одна – на отрезке [0,1], вторая – на отрезке [0,2]. Как выглядит её плотность распределения?

|  |  |  |  |
| --- | --- | --- | --- |
| «колокольчик» | треугольник | трапеция | прямоугольник |

В предыдущей задаче пусть указанные распределения – распределения классов 0 и 1 в задаче бинарной классификации. Оба класса равновероятны. Какая вероятность, что объект x=1 принадлежит классу 0?

|  |  |  |  |
| --- | --- | --- | --- |
| 1/2 | 2/3 | 3/4 | 1 |

При минимизации функции x^2 методом градиентного спуска с темпом 0.5 и начальной точкой 1.0, какая будет оценка argmin после 1й итерации?

|  |  |  |  |
| --- | --- | --- | --- |
| – 0.5 | 0 | 0.5 | 1 |

Выберите верные фразы

|  |  |
| --- | --- |
| Для селекции признаков обычно используют L2-регуляризацию | Логистическая регрессия – ленивый алгоритм |
| Евклидово расстояние – частный случай расстояния Махалонобиса | С помощью персептронного алгоритма можно решать системы линейных уравнений |

Чему рано максимальное значение MC (Missclassification criteria)?

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0.5 | e | 1 |

Пусть дана выборка целевых значений: 1, 2, 3 (упорядочено по времени получения меток). Используется модель константных алгоритмов (ответ равен среднему по всем меткам обучения). Функция ошибки – MAE (средний модуль отклонения). Чему равна средняя ошибка при контроле LOOCV (контроля по одному)?

|  |  |  |  |
| --- | --- | --- | --- |
| 0.5 | 2/3 | 1 | 3/2 |

В каком ансамбле следует использовать неустойчивые модели?

|  |  |  |  |
| --- | --- | --- | --- |
| бэгинг | случайные леса | бустинг | ECOC |

Что происходит при увеличении числа деревьев в градиентном бустинге (отметьте все варианты)?

|  |  |  |  |
| --- | --- | --- | --- |
| ошибка на обучении падает | ошибка на контроле падает | ошибка на обучении возрастает | ошибка на контроле возрастает |

Выберите верные фразы:

|  |  |
| --- | --- |
| Критерий gini используется для построения деревьев в задаче регресии | В экстремальных лесах (Extreme Random Trees) используется вычисление градиента ошибки |
| Контроль по фолдам используется для отбора модели | Аугментация – способ увеличения обучающей выборки |

В течение семестра студенты пишут онлайн-тесты и выполняют практические задания на решение задач машинного обучения. Для получения зачёта «автоматом» необходима сдача в срок всех промежуточных заданий. На тест по темам последних лекций (1-2) даётся 1 неделя, на практическое задание - 2-4 недели. В течение семестра студенты должны пройти 4 теста (результат не важен) и сделать 2 практических задания (необходимо превзойти «бенчмарк» – базовое решение). По практическому заданию необходимо предоставить код написанной программы (иногда в виде т.н. публичного кернела на платформе inclass.kaggle.com). Не получившие зачёт-автоматом сдают зачёт письменно, вариант зачёта – см. выше варианты заданий.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Численные методы**

**Вопросы к экзамену**

1. Связь метода Гаусса разложение матрицы на множители.

2. Обращение матрицы методом Гаусса-Жордана.

3. Метод квадратного корня решения системы линейных алгебраических уравнений.

4. Примеры экономических двухслойного традиционного метода решения системы линейных алгебраических уравнений.

5. Теорема о сходимости двухслойных и традиционных методов.

6. Достаточные условия сходимости методов: Якоби , Зейделя, простой итерации.

7. Теорема об оценке скорости сходимости итерационных методов.

8. Попеременно треугольной итерационный метод. Реализация метода теорема о сходимости попеременно треугольного итерационного метода.

9. Теорема об оценке скорости сходимости попеременно треугольного итерационного метода.

10. Степенной метод решения частичная проблема собственных значений.

11. Метод обратных итераций и обратных итераций со сдвигом решения.

12. Приведение матрицы к верхней почти треугольной форме при помощи преобразования элементарного отражения.

13. Понятие о курьер алгоритмы решения полной проблемы собственных значений не ухудшение верхний почти треугольной формы матрицы при QR алгоритм

14. Метод простой итерации решения нелинейных уравнений. Достаточное условие сходимости метода.

15. Метод Ньютона решения нелинейных уравнений и систем нелинейных уравнений. Метод секущих.

16. Сходимость метода Ньютона. Оценка скорости сходимости.

17. Постановка задачи интерполирования. Интерполяционная формула Лагранжа. Погрешность формулы.

18. Разделенные разности. Интерполяционная формула Ньютона.

19. Понятие об интерполирование с кратными узлами. Построение полинома Эрмита. Оценка погрешности полиномы Эрмита.

20. Применение полиномы Эрмита для получения точной оценки погрешности квадратурная формула Симпсона.

21. Наилучшее среднеквадратичное приближение функций в гильбертовом пространстве. Существование и единственность.

22. Явная разностная схема для первой краевой задачи уравнения теплопроводности. Аппроксимация, сходимость, устойчивость.

23. Чисто неявная разностная схема. Аппроксимация, сходимость.

24. Симметричная разностная схема. Аппроксимация, сходимость.

25. Разностная схема с весами. Вывод погрешности аппроксимации.

26. Основные понятия теории разностных схем: аппроксимация, устойчивость, сходимость. Теорема Филиппова.

27. Сходимость разностной задачи Дирихле для уравнения Пуассона.

28. Методы решения разностной задачи Дирихле.

29. Примеры численных методов решения задачи Коши для ОДУ. Погрешность аппроксимации 2-этапного метода Рунге-Кутта.

30. Общая формулировка m-этапного методы Рунге-Кутта. Оценка точности 2-этапного метода Рунге-Кутта.

31. Многошаговый разностные методы. Погрешность аппроксимации. Понятие устойчивости.

32. Жесткие системы обыкновенных дифференциальных уравнений.

33. Примеры разных схем для интегрирования жестких систем обыкновенных дифференциальных уравнений.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Кафедральный практикум**

**Типовые варианты практических заданий и вопросов при их сдаче для проведения текущего контроля успеваемости**

**Практическое задание №1**

Первое практическое задание заключается в знакомстве со средой pgAdmin и написании простейших SQL-запросов с использованием оператора SELECT. Для модельной базы данных «KingCorporation» должны быть составлены 4 запроса согласно индивидуальному варианту. Скрипт для создания и заполнения модельной базы данных и её описание: http://sp.cs.msu.ru/prak3/sample\_db.sql

После составления запросов следует убедиться в их правильности при помощи более простых запросов.

**Темы для проработки в рамках задания:**

Основные понятия реляционных и SQL-ориентированных баз данных.

• Синтаксис SQL https://postgrespro.ru/docs/postgrespro/11/sql-syntax

• SELECT-запросы https://postgrespro.ru/docs/postgrespro/11/queries

• Функции и операторы https://postgrespro.ru/docs/postgrespro/11/functions

• Оконные функции https://postgrespro.ru/docs/postgrespro/9.5/tutorial-window

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

• Структура учебной базы данных «KingCorporation»

• Работа в среде pgAdmin

**Примеры вопросов по заданию**

• Объяснить, как работают написанные запросы.

• Рассказать про операцию соединения (JOIN) и различные её разновидности.

• Рассказать про агрегатные функции, предложения GROUP BY и HAVING.

• Как выбрать только уникальные значения какого-либо столбца?

• Как осуществить сортировку по возрастанию/убыванию по значению какого-либо столбца?

• Как агрегатные функции ведут себя по отношению к неопределённым значениям?

• Рассказать о теоретико-множественных операциях в SQL.

• Чем отличаются UNION и UNION ALL?

• Чем отличаются COUNT(\*) и COUNT(field)?

• Как подсчитать количество уникальных значений столбца?

• Как можно осуществить проверку на неопределенное значение?

• Рассказать про предикат LIKE.

• Как можно выбрать только определенное количество строк?

• Чем SQL-таблица отличается от отношения?

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Округлить результирующее значение до 3 знаков после точки.

• Округлить вещественное число до целого без нулей после точки.

• Переписать запрос, не используя функцию MAX (MIN).

• Изменить формат вывода данных (например, формат даты и времени).

• Написать или модифицировать запрос по сформулированному заданию.

**Блок практических заданий** №№ 2.1-2.4

Блок практических заданий 2.1-2.4 призван сформировать у студента понимание особенностей хранения данных приложения в РСУБД и умение настраивать и поддерживатьэто хранение.

*Практическое задание №2.1 Проектирование схемы базы данных*

Задание связано с проектированием схемы базы данных для работы приложения (WEB/Mobile/Desktop). Каждый индивидуальный вариант содержит предметную область, из которой должна быть проектируемая база данных. Задачей студента является решить, для чего будет использоваться создаваемая база данных, и, исходя из этого, построить её концептуальную схему. Результатом выполнения задания является схема базы данных (в виде ER-диаграммы, содержащей таблицы и связи между ними, без уточнения типов столбцов). При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

https://www.lucidchart.com/pages/examples/er-diagram-tool

*Типовые индивидуальные варианты (сквозные, т. к. используются в блоках практических заданий №№2.1-2.4, 3.1-3.4):*

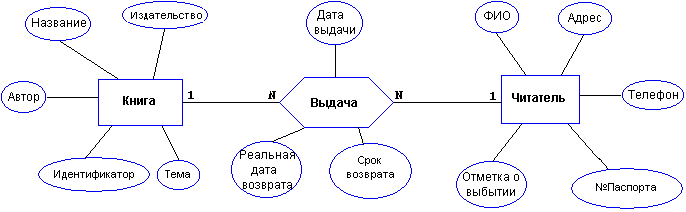
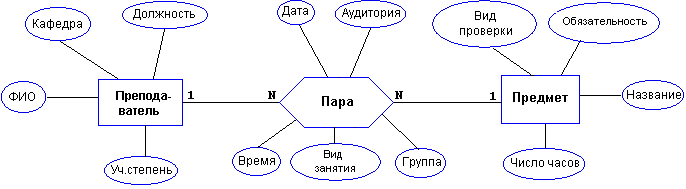
Вариант №1. Библиотека

Рис. Пример схемы базы данных для варианта №1

Список запросов для реализации на языке SQL:

1. Выбрать значения задержки поставки товара после заказа для всех покупателей, живущих в штате 'CA' (Калифорния)
2. Выбрать названия и города тех отделов, у которых нет начальников (должность - 'MANAGER')
3. Выбрать даты и имена покупателей тех заказов, которые обслуживал продавец 'SMITH' и в которых сумма заказа превышала кредит покупателя.
4. Выбрать названия продукта, для которого в 1991 г. было продано минимальное количество экземпляров продукта.

## Вариант №2. Университет

**Рис. Пример схемы базы данных для варианта №2**

Список запросов для реализации на языке SQL:

1. Выбрать цены на сегодняшний день всех продуктов, в названии которых есть слова 'WIFF SOFTBALL'.
2. Выбрать названия и города отделов, в которых есть аналитики, и их количество (должность - 'ANALYST').
3. Выбрать названия и города тех отделов, в которых есть сотрудники, не являющиеся продавцами (должность - не 'SALESPERSON'), но получающие комиссионные.
4. Выбрать минимальную сумму продаж, которая приходится на одного сотрудника, работающего в городе 'NEW YORK'.

*Темы для проработки в рамках задания*

• Модель "сущность-связь" (ER-модель).

• Первичные и внешние ключи.

• Типы связей и их моделирование.

• Нормальные формы и нормализация.

*Требования к созданной схеме базы данных*

• Схема должна соответствовать поставленной задаче.

• Связи между сущностями должны быть правильно смоделированы.

• Таблицы должны удовлетворять, по крайней мере, третьей нормальной форме.

• Желательно придерживаться какой-либо системы в именовании таблиц и столбцов.

*Практическое задание №2.2 Создание и заполнение таблиц*

Задание заключается в подготовке SQL-скрипта для создания таблиц согласно схеме, созданнойпри выполнении предыдущего задания (с уточнением типов столбцов). Необходимо определить первичные и внешние ключи, а также декларативные ограничения целостности (возможность принимать неопределенное значение, уникальные ключи, проверочные ограничения и т. д.). Таблицы следует создавать в отдельной базе данных. Кроме того, нужно подготовить данные для заполнения созданных таблиц. Объем подготовленных данных должен составлять не менее 10 экземпляров для каждой из стержневых сущностей и 20 экземпляров для каждой из ассоциативных. На основе этих данных необходимо создать SQL-скрипт для вставки соответствующих строк в таблицы БД.

*Темы для проработки в рамках задания*

• Язык DDL, операторы CREATE TABLE и ALTER TABLE https://postgrespro.ru/docs/postgrespro/11/ddl-basics

https://postgrespro.ru/docs/postgrespro/11/ddl-default

https://postgrespro.ru/docs/postgrespro/11/ddl-alter

• Типы данных https://postgrespro.ru/docs/postgrespro/11/datatype

• Декларативные ограничения целостности https://postgrespro.ru/docs/postgrespro/11/ddl-constraints

• Оператор INSERT https://postgrespro.ru/docs/postgrespro/11/dml-insert https://postgrespro.ru/docs/postgrespro/11/dml-returning

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания*

• Объяснить, что делают написанные запросы.

• В чём различие типов CHAR и VARCHAR? VARCHAR и TEXT?

• Что такое внешний ключ?

• Какие существуют способы поддержания ссылочной целостности?

• Что такое уникальный ключ?

• Что такое SERIAL?

• Рассказать о значениях по умолчанию и неопределенных значениях.

• Как можно хранить даты и время?

• Рассказать о числовых типах данных.

• Каким образом можно вставить несколько строк с помощью одного оператора INSERT?

• Как ведет себя оператор INSERT, если в списке столбцов перечислены не все столбцы?

• Добавить какие-либо ограничения целостности.

• Добавить SERIAL.

• Исправить выявленные при проверке недочеты.

*Практическое задание №2.3. Операторы манипулирования*

Задание посвящено манипулированию данными с помощью операторов SQL. В ходе выполнения задания необходимо:

• Подготовить еще 3-4 выборки, которые имеют осмысленное значение для предметной области, и также составить для них SQL-скрипты.

• Сформулировать 3-4 запроса на изменение и удаление из базы данных

Запросы должны быть сформулированы в терминах предметной области. Среди запросов обязательно должны быть такие, которые будут вызывать срабатывание ограничений целостности. Составить SQL-скрипты для выполнения этих запросов.

*Темы для проработки в рамках задания:*

• Оператор SELECT https://postgrespro.ru/docs/postgrespro/11/queries

• Оператор UPDATE https://postgrespro.ru/docs/postgrespro/11/dml-update

• Оператор DELETE https://postgrespro.ru/docs/postgrespro/11/dml-delete

• Декларативные ограничения целостности https://postgrespro.ru/docs/postgrespro/11/ddl-constraints

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Объяснить, как работают написанные запросы.

• Примеры вопросов по оператору SELECT см. в практическом задании №1.

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Написать или модифицировать запрос по сформулированному заданию.

*Практическое задание №2.4 Контроль целостности данных*

Задание посвящено контролю целостности данных, который производится с помощью механизма транзакций и триггеров. Транзакции позволяют рассматривать группу операций как единое целое, либо отрабатывают все операции, либо ни одной. Это позволяет избегать несогласованности данных. Триггеры позволяют проверять целостность данных в момент выполнения транзакций, поддерживать целостность, внося изменения, и откатывать транзакции, приводящие к потере целостности.

Необходимо подготовить SQL-скрипты для проверки наличия аномалий (потерянных изменений, грязных чтений, неповторяющихся чтений, фантомов) при параллельном исполнении транзакций на различных уровнях изолированности SQL/92 (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE). Подготовленные скрипты должны работать с одной из таблиц, созданных в практическом задании №2.1. Для проверки наличия аномалий потребуются два параллельных сеанса, операторы в которых выполняются пошагово:

• Установить в обоих сеансах уровень изоляции READ UNCOMMITTED. Выполнить сценарии проверки наличия аномалий потерянных изменений и грязных чтений.

• Установить в обоих сеансах уровень изоляции READ COMMITTED. Выполнить сценарии проверки наличия аномалий грязных чтений и неповторяющихся чтений.

• Установить в обоих сеансах уровень изоляции REPEATABLE READ. Выполнить сценарии проверки наличия аномалий неповторяющихся чтений и фантомов.

• Установить в обоих сеансах уровень изоляции SERIALIZABLE. Выполнить сценарий проверки наличия фантомов.

Необходимо составить скрипт для создания триггера, а также подготовить несколько запросов для проверки и демонстрации его полезных свойств:

• Изменение данных для сохранения целостности.

• Проверка транзакций и их откат в случае нарушения целостности.

*Темы для проработки в рамках задания:*

• Понятие транзакции, свойства транзакций. https://postgrespro.ru/docs/postgrespro/11/tutorial-transactions

• Уровни изолированности и аномалии https://postgrespro.ru/docs/postgrespro/11/transaction-iso

• Триггеры и триггерные функции https://postgrespro.ru/docs/postgrespro/11/trigger-definition

https://postgrespro.ru/docs/postgrespro/11/plpgsql-trigger

• Сообщения и ошибки https://postgrespro.ru/docs/postgrespro/11/plpgsql-errors-and-messages

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Рассказать об аномалиях доступа к БД.

• Перечислить аномалии, возникающие на каждом из уровней изолированности.

• Рассказать о свойствах транзакций.

• Рассказать об управлении транзакциями.

• Что такое тупики? Как бороться с тупиками?

• На каком уровне изолированности возможны тупики?

• Как обеспечивается изолированность транзакций в СУБД?

• Как бороться с проблемой фантомов?

• Что такое журнал транзакций?

• Как обеспечивается постоянство хранения (durability) в СУБД?

• Объяснить принцип работы написанного триггера.

• Какие бывают типы триггеров?

• Когда может срабатывать триггер?

• В каком порядке срабатывают триггеры?

• Можно ли менять порядок срабатывания триггеров?

• Сработает ли триггер, если оператор, выполненный пользователем, не затрагивает ни одну строку таблицы?

• Продемонстрировать откат транзакции при возникновении ошибок.

• Продемонстрировать возникновение тупика.

• Исправить неверные сценарии проверки аномалий

• Исправить ошибки в работе триггера.

• Модифицировать триггер каким-либо образом.

**Блок практических заданий** №№ 3.1-3.4

Блок практических заданий №№ 3.1-3.4 призван сформировать у студента понимание особенностей создания аналитических баз данных и умение их настраивать и поддерживать.

*Практическое задание №3.1 Проектирование схемы базы данных*

Задание связано с проектированием схемы базы данных для аналитики. Будем исходить из того, что приложение, для которого была сделана база данных в задании №2.1, стало очень популярным и по нему каждый день можно собирать большой объем статистической информации. Что это будет за статистика? Почему именно ее необходимо собирать, обрабатывать и анализировать? Задачей студента является ответить на эти вопросы, и, исходя из этого, разработать базу данных и заполнить ее данными. Результатом данного практического задания является схема базы данных, скрипты создания базы данных и ее заполнения, обладающие следующими свойствами:

• Как минимум одна таблица должна содержать не меньше 100 млн. записей, которые со временем теряют актуальность.

• Другая таблица, связанная с первой, должна содержать не меньше 1 млн. записей.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с текстом, по которой будет необходимо настроить полнотекстовый поиск.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с данными в json-формате.

• В одной из таблиц с количество записей больше 1 млн. должна быть колонка с массивом.

При выполнении задания важно учитывать плюсы и минусы денормализации схемы данных и использования массивов и json-формата. При сдаче задания студент должен обосновать соответствие созданной схемы поставленной задаче.

Для проектирования схемы и построения диаграммы можно использовать любые средства, один из вариантов использовать сайт:

https://www.lucidchart.com/pages/examples/er-diagram-tool

*Темы для проработки в рамках задания:*

• Денормализация https://habr.com/ru/company/latera/blog/281262/ https://habr.com/ru/post/64524/

• Массивы https://postgrespro.ru/docs/postgrespro/11/arrays https://postgrespro.ru/docs/postgrespro/11/functions-array

• JSON https://postgrespro.ru/docs/postgrespro/11/datatype-json https://postgrespro.ru/docs/postgrespro/11/functions-json

• Наполнение базы данных https://postgrespro.ru/docs/postgrespro/11/populate

*Практическое задание №3.2 Оптимизация*

Задание посвящено ускорению выполнения запросов. Для этого могут быть использованы механизмы секционирования, наследования и индексов. Для выполнения задания необходим достаточно большой объем данных, чтобы оптимизация была целесообразной (порядка 1 млн. строк в каждой таблице). Необходимо подготовить два запроса:

• Запрос к одной таблице, содержащий фильтрацию по нескольким полям.

• Запрос к нескольким связанным таблицам, содержащий фильтрацию по нескольким полям.

Для каждого из этих запросов необходимо провести следующие шаги:

• Получить план выполнения запроса без использования индексов.

• Получить статистику (IO и Time) выполнения запроса без использования индексов.

• Создать нужные индексы, позволяющие ускорить запрос.

• Получить план выполнения запроса с использованием индексов и сравнить с первоначальным планом.

• Получить статистику выполнения запроса с использованием индексов и сравнить с первоначальной статистикой.

• Оценить эффективность выполнения оптимизированного запроса.

Также необходимо продемонстрировать полезность индексов для организации полнотекстового поиска, фильтрации с использованием массива и json-формата. Для таблицы объемом больше 100 млн. записей произвести оптимизацию, позволяющую быстро удалять старые данные, ускорить вставку и чтение данных.

*Темы для проработки в рамках задания:*

• EXPLAIN https://postgrespro.ru/docs/postgrespro/11/using-explain https://postgrespro.ru/docs/postgrespro/11/planner-stats

https://postgrespro.ru/docs/postgrespro/11/explicit-joins

• ANALYZE https://postgrespro.ru/docs/postgrespro/11/routine-vacuuming#VACUUM-FOR-STATISTICS

• Индексы https://postgrespro.ru/docs/postgrespro/11/indexes

• Полнотекстовый поиск https://postgrespro.ru/docs/postgrespro/11/textsearch

• Наследование таблиц https://postgrespro.ru/docs/postgrespro/11/ddl-inherit

• Секционирование таблиц https://postgrespro.ru/docs/postgrespro/11/ddl-partitioning

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• В чем отличие первичного ключа и уникального индекса?

• В каких случаях имеет смысл создавать индексы? Какие колонки следует включать в индекс и почему?

• Какие существуют способы внутренней организации индексов?

• Рассказать о проблеме фрагментации индексов. Как бороться с фрагментацией?

• Имеет ли значение порядок указания колонок при создании индекса?

• В чем разница между IndexScan и IndexSeek?

• В чем разница между секционированием и наследованием?

• Зачем нужен ANALYZE?

• Исправить ошибки в подготовленных выборках.

• Могут ли индексы ухудшить производительность? Если да, то продемонстрировать это.

• На что влияет порядок сортировки (ASC\DESC) при создании индекса? Продемонстрировать это.

• Продемонстрировать полезность индекса по выражению.

• Продемонстрировать полезность частичного индекса.

*Практическое задание №3.3 Представления и функции*

Задание посвящено упрощению работы аналитика и ограничения его возможностей с помощью созданию и использованию представлений и функций: требуется составить SQL-скрипты для создания нескольких представлений и функций, которые позволяли бы упростить манипуляции с данными или позволяли бы не показывать аналитику всю базу данных, предоставив только необходимую часть информации.

*Темы для проработки в рамках задания:*

• Представления https://postgrespro.ru/docs/postgrespro/11/sql-createview https://postgrespro.ru/docs/postgrespro/11/rules-views

https://postgrespro.ru/docs/postgrespro/11/rules-materializedviews

• Функции https://postgrespro.ru/docs/postgrespro/11/xfunc-sql

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Объяснить, как работают написанные запросы.

• Рассказать о CHECK OPTION.

• Рассказать о модификации данных через представления.

• Рассказать о вставке данных через представления.

• Примеры вопросов по оператору SELECT см. в задании №1.

• Исправить неверно работающий запрос (запросы).

• Упростить один или несколько запросов.

• Продемонстрировать изменение и вставку данных через представления.

• Написать или модифицировать запрос по сформулированному заданию.

• Продемонстрировать полезность материализованного представления.

*Практическое задание №3.4 Управление доступом*

В рамках выполнения задания происходит освоение способов управления доступом. При выполнении задания необходимо:

• Создать пользователя test и выдать ему доступ к базе данных.

• Составить и выполнить скрипты присвоения новому пользователю прав доступа к таблицам, созданным в практическом задании №3.1. При этом права доступа к различным таблицам должны быть различными, а именно:

o По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT, INSERT, UPDATE в полном объеме.

o По крайней мере, для одной таблицы новому пользователю присваиваются права SELECT и UPDATE только избранных столбцов.

o По крайней мере, для одной таблицы новому пользователю присваивается только право SELECT.

• Присвоить новому пользователю право доступа (SELECT) к представлению, созданному в практическом задании №3.1

• Создать стандартную роль уровня базы данных, присвоить ей право доступа (UPDATE на некоторые столбцы) к представлению, созданному в практическом задании №3.3, назначить новому пользователю созданную роль.

• Выполнить от имени нового пользователя некоторые выборки из таблиц и представления. Убедиться в правильности контроля прав доступа.

• Выполнить от имени нового пользователя операторы изменения таблиц с ограниченными правами доступа. Убедиться в правильности контроля прав доступа.

*Темы для проработки в рамках задания:*

• Роли и пользователи https://postgrespro.ru/docs/postgresql/11/user-manag

https://postgrespro.ru/docs/postgrespro/11/app-createuser

• Директивы GRANT и REVOKE https://postgrespro.ru/docs/postgrespro/11/ddl-priv

https://postgrespro.ru/docs/postgrespro/11/ddl-schemas#DDL-SCHEMAS-PRIV

• Полное описание синтаксиса встретившихся команд https://postgrespro.ru/docs/postgrespro/11/sql-commands

*Примеры вопросов при сдаче задания:*

• Для чего нужны роли?

• Что такое схема?

• Рассказать про директивы GRANT и REVOKE.

• Для чего нужна роль PUBLIC?

• Как добавить нового пользователя в текущую базу данных?

• Как позволить пользователю заходить на сервер?

• Какие существуют права?

• Исправить ошибки в обязательной части.

• Сменить владельца базы данных.

• Сменить пароль для пользователя.

• Определить роль с заданными правами.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_