

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики

**УТВЕРЖДАЮ**  
декан факультета вычислительной  
математики и кибернетики

**/И.А. Соколов /**  
**«27» сентября 2022г.**

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Практикум на ЭВМ (Программирование)**

---

**Уровень высшего образования:**

**бакалавриат**

**Направление подготовки / специальность:**

**01.03.02 "Прикладная математика и информатика" (3++)**

**Направленность (профиль) ОПОП:**

**Искусственный интеллект и анализ данных**

**Форма обучения:**

**очная**

Рабочая программа рассмотрена и утверждена  
на заседании Ученого совета факультета ВМК  
(протокол № 7 от 27 сентября 2022 года)

Москва 2022

Рабочая программа дисциплины (модуля) разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.04.02 "Прикладная математика и информатика" утвержденного Приказом Министерства образования и науки РФ от 10 января 2018 г. N 13.

## 1. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОПОП ВО:

Настоящая дисциплина включена в учебный план по направлению 01.03.02 Прикладная математика и информатика, профиль Искусственный интеллект и анализ данных и входит в Обязательная часть Блока 1.

Дисциплина изучается на 1,2 курсе в 1,2,3,4 семестре.

## 2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ:

Цель изучения дисциплины: обучение студентов методам проектирования, описания на языке высокого уровня и тестирования программных реализаций алгоритмов различных математических задач.

## 3. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ):

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1. Разрабатывает программу для решения задачи с использованием языка высокого уровня. ОПК-5.2. Умение создавать, тестировать и отлаживать программы на языках программирования высокого уровня на компьютере. ОПК-5.3. Навыки написания качественного и хорошо документированного программного кода	Знать: – основные свойства алгоритмов; – основные формализации алгоритмов – машину Тьюринга и Нормальные алгоритмы Маркова; – о сложности стандартных алгоритмов; – способы формального описания языков ; – синтаксис и семантику языка высокого уровня; – способы разработки программного обеспечения; – типовые динамические структуры данных ; – знать об основных управляющих структурах Паскаля. принципы организации и функционирования компьютеров; особенности архитектуры процессора персонального компьютера (ПК); базовые концепции языка ассемблера ПК; технологии выполнения программ на ЭВМ. основные понятия, алгоритмы и методы организации управления процессами в операционной системе UNIX; основные понятия, алгоритмы и методы организации взаимодействия процессов в операционной системе UNIX; основные понятия, алгоритмы и методы организации работы с файлами; основные понятия языка C++; понятие перегрузки функций и операций; основные понятия наследования, единичное и множественное наследование; понятие механизма виртуальных функций; понятие об обработке исключений в C++; основы параметрического полиморфизма, шаблоны функций и классов; основы теории формальных языков и грамматик;

		<p>классификацию формальных грамматик и языков по Хомскому;</p> <p>понятие приведенных грамматик, алгоритм получения приведенной КС-грамматики;</p> <p>основы теории трансляции;</p> <p>алгоритм преобразования недетерминированного конечного автомата в детерминированный;</p> <p>метод рекурсивного спуска;</p> <p>понятие синтаксически управляемого перевода;</p> <p>ПОЛИЗ;</p> <p>задачи и схемы работы лексического и синтаксического анализаторов.</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>– применять и адаптировать для решения задач стандартные алгоритмы;</li> <li>– составлять машины Тьюринга и нормальные алгоритмы Маркова для решения задач;</li> <li>– строить металингвистические формулы и синтаксические диаграммы для модельных языков;</li> <li>– понимать и объяснять тексты программ на Паскале;</li> <li>– создавать, использовать, модифицировать программы, работающие со стандартными структурами данных;</li> <li>– пользоваться основными управляющими структурами Паскаля;</li> <li>– создавать и применять процедуры и функции, в том числе и рекурсивные, в программах на Паскале;</li> <li>– пользоваться процедурами и функциями, в том числе и рекурсивными, в программах на Паскале;</li> <li>– создавать, использовать, модифицировать программы, работающие с динамическими структурами данных</li> </ul> <p>составлять программы на машинном языке для модельных ЭВМ различных архитектур;</p> <p>разрабатывать программы на языке Ассемблере ПК;</p> <p>организовывать связь программы и процедур на Ассемблере в соответствии со стандартными соглашениями о связях;</p> <p>организовывать связь программ, написанных на языке высокого уровня (Free Pascal) и на машинно-зависимом языке (Ассемблер).</p> <p>разрабатывать алгоритмы для решения типовых задач, оценивать сложность полученных алгоритмов,</p> <p>реализовывать программы на языке Си с использованием системных вызовов ОС UNIX,</p>
--	--	---

		<p>тестировать написанные самостоятельно программы на соответствие исходным требованиям;</p> <p>находить, анализировать и контекстно обрабатывать научно-техническую информацию;</p> <p>создавать алгоритмические модели типовых задач, проводить спецификацию задачи, реализовывать программы на языке C++, оценивать сложность полученных алгоритмов;</p> <p>разрабатывать алгоритмы для решения задач системного и прикладного программирования.</p> <p>применять алгоритм получения приведенной КС-грамматики;</p> <p>применять метод преобразования недетерминированного конечного автомата к детерминированному;</p> <p>применять метод рекурсивного спуска к КС-грамматикам;</p> <p>применять на практике основные методы теории трансляции.</p> <p>Владеть:</p> <p>навыками использования компонент системы программирования для подготовки и запуска на выполнение программы на Ассемблере.</p> <p>основами алгоритмизации, пониманием методов построения алгоритмов на основе разбиения задачи на подзадачи;</p> <p>навыками программирования на языке Си с использованием функций стандартной библиотеки языка Си, а также с использованием библиотеки системных вызовов ОС UNIX;</p> <p>навыками написания программ для работы с текстовыми и бинарными файлами;</p> <p>базовыми навыками разработки и реализации параллельных программ, организации взаимодействия процессов с использованием средств, предоставляемых ОС UNIX;</p> <p>навыками работы с пользовательским интерфейсом ОС UNIX.</p> <p>навыками решения практических задач на языке C++;</p> <p>навыками решения практических задач по теории формальных грамматик.</p>
--	--	---

4. Объем дисциплины (модуля) составляет 12 з.е., в том числе 216 академических часов, отведенных на контактную работу обучающихся с преподавателем, 216 академических часов на самостоятельную работу обучающихся.

**5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ) С УКАЗАНИЕМ ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ И ВИДЫ УЧЕБНЫХ ЗАНЯТИЙ**

**5.1. Структура дисциплины (модуля) по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий (в строгом соответствии с учебным планом)**

1 семестр

Наименование разделов и тем дисциплины (модуля),  Форма промежуточной аттестации по дисциплине (модулю)	Номинальные трудозатраты обучающегося		Самостоятельная работа обучающегося, академические часы	Всего академических часов	Форма текущего контроля успеваемости* (наименование)
	Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, академические часы				
	Занятия лекционного типа	Занятия семинарского типа			
Системы счисления		2	0	2	Практическая работа
Машины Тьюринга		8	8	16	Практическая работа
Нормальные алгоритмы Маркова		8	8	16	Практическая работа
Металингвистические формулы. Синтаксические диаграммы.		8	8	16	Практическая работа
Язык программирования Pascal		4	2	6	Практическая работа
Ссылки, списки		8	8	16	Практическая работа
Стеки, очереди		8	8	16	Практическая работа
Деревья. Двоичные деревья поиска. AVL-деревья.		8	8	16	Практическая работа
Промежуточная аттестация (зачет)				4	зачет
<b>Итого</b>	0	54	50	<b>108</b>	—

2 семестр

Наименование разделов и тем дисциплины (модуля),  Форма промежуточной аттестации по дисциплине (модулю)	Номинальные трудозатраты обучающегося		Всего академических часов	Форма текущего контроля успеваемости* (наименование)
	Контактная работа (работа во взаимодействии с преподавателем) Виды контактной	Самостоятельная работа обучающегося,		

	работы, академические часы		академические часы		
	Занятия лекционного типа	Занятия семинарского типа			
Выдача заданий практикума		4	0	4	Контрольная работа
Представление чисел в ЭВМ		2	4	6	Контрольная работа
Модельные ЭВМ		6	4	10	Контрольная работа
Директивы, арифметика, переходы		6	4	10	Контрольная работа
Полные программы		2	4	6	Контрольная работа
Массивы, структуры, битовые команды		6	4	10	Контрольная работа
Стек и процедуры		6	4	10	Контрольная работа
Строки		2	4	6	Контрольная работа
Макросы, условное ассемблирование		4	4	8	Контрольная работа
Практическая работа на ЭВМ		16	18	34	Контрольная работа
Промежуточная аттестация (зачет)				4	зачет
<b>Итого</b>	0	54	50	<b>108</b>	—

### 3 семестр

Наименование разделов и тем дисциплины (модуля),  Форма промежуточной аттестации по дисциплине (модулю)	Номинальные трудозатраты обучающегося		Всего академических часов	Форма текущего контроля успеваемости* (наименование)
	Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, академические часы	Самостоятельная работа обучающегося, академические часы		

	Занятия лекционного типа	Занятия семинарского типа			
Язык программирования Си: Повторение.		4	6	10	Самостоятельная работа, индивидуальный опрос, прием практического домашнего задания
Язык программирования Си: Указатели на функции. Работа с файлами (стандартная библиотека языка Си).		8	6	14	Индивидуальный опрос, прием практического задания, прием практического домашнего задания
Интерфейс с системой UNIX: Низкоуровневая работа с файлами.		10	10	20	Индивидуальный опрос, прием практического задания, прием практического домашнего задания
Интерфейс с системой UNIX: Организация работы с процессами.		6	4	10	Индивидуальный опрос,



					прием практического домашнего задания
Интерфейс с системой UNIX: Взаимодействие процессов.		16	16	32	Индивидуальный опрос, прием практического задания, прием практического домашнего задания. Контрольная работа
Интерфейс с системой UNIX: Межпроцессное взаимодействие средствами IPC.		10	8	18	Индивидуальный опрос, прием практического домашнего задания. Практическая контрольная работа
Промежуточная аттестация (зачет)				4	зачет
<b>Итого</b>	0	54	50	<b>108</b>	—

4 семестр

Наименование разделов и тем дисциплины (модуля),  Форма промежуточной аттестации по дисциплине (модулю)	Номинальные трудозатраты обучающегося		Всего академических	Форма текущего контроля успеваемости* (наименование)
	Контактная работа (работа во взаимодействии с преподавателем) Виды контактной работы, академические	Самостоятельная работа обучающегося, академическая		

	часы		не часы		
	Занятия лекционного типа	Занятия семинарского типа			
Объектно-ориентированное программирование		2	4	6	Контрольная работа
Статический полиморфизм в С++.		6	6	12	Контрольная работа
Понятие наследования.		4	6	10	Контрольная работа
Множественное наследование в С++.		4	6	10	Контрольная работа
Исключения в С++.		6	6	12	Контрольная работа
Динамическая идентификация		4	6	10	Контрольная работа
Параметрический полиморфизм.		4	6	10	Контрольная работа
Основные понятия теории формальных грамматик.		6	2	8	Контрольная работа
Регулярные и автоматные грамматики.		6	2	8	Контрольная работа
Метод рекурсивного спуска.		4	2	6	Контрольная работа
Грамматики с действиями.		4	2	6	Контрольная работа
ПОЛИЗ		4	2	6	Контрольная работа
Промежуточная аттестация (зачет)				4	зачет
<b>Итого</b>	0	54	50	<b>108</b>	—

## 5.2. Содержание разделов (тем) дисциплины

1 семестр

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин

1.	Системы счисления	Представление числовых данных, системы счисления, логические данные и операции.
2.	Машины Тьюринга	Абстрактные машины. Система команд. Примеры схем машины Тьюринга. Вычислимые по Тьюрингу функции. Основная гипотеза теории алгоритмов. Машины Тьюринга и современные ЭВМ.
3.	Нормальные алгоритмы Маркова	Основные понятия ассоциативного исчисления. Способы композиции нормальных алгоритмов. Эквивалентность различных теорий алгоритмов. Основные понятия ассоциативного исчисления. Эквивалентность различных теорий алгоритмов.
4.	Металингвистические формулы.	Понятие формальной грамматики. Грамматики как способ определения синтаксиса языков программирования
5.	Синтаксические диаграммы.	Способы описания синтаксиса языков программирования: металингвистические формулы и диаграммы Вирта. Примеры.
6.	Язык программирования Pascal	Создание программы, компиляция, отладка, среда разработки. Константы, переменные. Основные типы данных и указатели. Операции и выражения. Ввод и вывод. Линейные программы. условный оператор, оператор выбора, операторы цикла. Целочисленные вычисления. Вычисления с плавающей точкой. синтаксис объявления и определения. Параметры, вызов функции, возвращаемое значение. Стек исполнения. Рекурсия. Заголовочные файлы, отдельная компиляция. Обработка массивов и строк символов с помощью индексов.
7.	Ссылки, списки	Ссылки, списки
8.	Стеки, очереди	Стеки, очереди
9.	Деревья. Двоичные деревья поиска. AVL-деревья.	Деревья. Двоичные деревья поиска. AVL-деревья.

## 2 семестр

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин
1.	Выдача заданий практикума	Выдача заданий практикума
2.	Представление чисел в ЭВМ	Представление чисел в ЭВМ
3.	Модельные ЭВМ	Модельные ЭВМ
4.	Директивы, арифметика, переходы	Директивы, арифметика, переходы
5.	Полные программы	Полные программы
6.	Массивы, структуры, битовые команды	Массивы, структуры, битовые команды
7.	Стек и процедуры	Стек и процедуры
8.	Строки	Строки
9.	Макросы, условное ассемблирование	Макросы, условное ассемблирование
10.	Практическая работа на ЭВМ	Практическая работа на ЭВМ

## 3 семестр

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин
1.	Язык программирования Си: Повторение.	Аргументы командной строки, работа со строками, динамическое выделение памяти, динамические

		структуры данных.
2.	Язык программирования Си: Указатели на функции. Работа с файлами (стандартная библиотека языка Си).	Указатели на функции. Чтение сложных деклараций. Работа с файлами с использованием функций стандартной библиотеки языка Си: открытие и закрытие файлов, позиционирование в файле, чтение/запись данных. Функции fopen(), fclose(), fseek(), ftell(), fgets(), fputs(), fread(), fwrite() и т.д.
3.	Интерфейс с системой UNIX: Низкоуровневая работа с файлами.	Низкоуровневая работа с файлами. Работа с файлами с использованием системных вызовов ОС UNIX. Таблицы, используемые ОС UNIX для работы с файлами. Понятия индексного дескриптора и файлового дескриптора. Открытие и закрытие, создание и удаление файлов, позиционирование в файле, чтение/запись данных, права доступа к файлу, получение информации о файле. Функции open(), close(), lseek(), read(), write(), stat() и т.д.
4.	Интерфейс с системой UNIX: Организация работы с процессами.	Понятие процесса. Идентификация процессов. Создание и завершение процессов. Ожидание завершения процесса-потомка. Замена тела процесса. Системные вызовы: fork(), exit(), getpid(), getppid(), wait(), waitpid(), вызовы семейства exec().
5.	Интерфейс с системой UNIX: Взаимодействие процессов.	Взаимодействие процессов. Каналы. Перенаправление ввода-вывода. Организация конвейера. Системные вызовы: dup(), dup2(), pipe(). Сигналы. Функции: signal(), kill(), alarm(), pause().
6.	Интерфейс с системой UNIX: Межпроцессное взаимодействие средствами IPC.	Межпроцессное взаимодействие средствами IPC. Именованные разделяемые объекты. Очереди сообщений, разделяемая память, семафоры.

#### 4 семестр

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин
1.	Объектно-ориентированное программирование	Обзор основных свойств и возможностей объектно-ориентированного программирования на языке Си++. Работа с динамической памятью.
2.	Статический полиморфизм в С++.	Перегрузка функций и операторов. Перегрузка унарных операций с помощью функции-члена класса и с помощью функции-друга класса. Правила перегрузки бинарных и унарных операций в С++. Особенности перегрузки префиксных и постфиксных операций инкремента и декремента, операции ->, операции индексирования, операции присваивания, операции приведения к типу, операции ввода – вывода << и >>. Решение типовых задач.

3.	Понятие наследования.	Единичное наследование. Наследование и замещение, правила видимости. Конструкторы и деструкторы производных классов: порядок вызова, передача параметров. Указатели на базовый и производный классы, преобразование указателей. Виртуальные функции. Абстрактные классы. Использование виртуальных деструкторов. Решение типовых задач.
4.	Множественное наследование в C++.	Основные проблемы и способы их решения. Преобразование указателей. Виртуальные базовые классы. Неоднозначность из-за совпадающих имён в различных базовых классах.
5.	Исключения в C++.	Общая схема обработки исключений: try - throw – catch. Правило выбора обработчика исключения. Решение типовых задач.
6.	Динамическая идентификация	Динамическая идентификация типа, преобразования типов. Средства RTTI.
7.	Параметрический полиморфизм.	Программирование с помощью шаблонов. Шаблоны функций и классов .
8.	Основные понятия теории формальных грамматик.	Эквивалентность грамматик. Типы грамматик и языков по Хомскому. Дерево вывода. Неоднозначность грамматики. Неоднозначность языка. Приведенные грамматики. Преобразования грамматик. Решение типовых задач.
9.	Регулярные и автоматные грамматики.	Левосторонние и правосторонние регулярные грамматики. Алгоритм получения левосторонней грамматики по правосторонней. Построение диаграммы состояний. Построение детерминированного конечного автомата по недетерминированному конечному автомату. Решение типовых задач.
10.	Метод рекурсивного спуска.	Подкласс грамматик, к которому применим метод. Критерий применимости метода рекурсивного спуска. Решение типовых задач.
11.	Граматики с действиями.	Синтаксически управляемый перевод. Решение типовых задач.
12.	ПОЛИЗ	ПОЛИЗ выражений, алгоритм интерпретации ПОЛИЗ. Алгоритм Дейкстры. ПОЛИЗ операторов языков программирования. Решение типовых задач.

## **6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ФОС, ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ) ДЛЯ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ).**

Фонд оценочных средств приведен в отдельном документе

## **7. РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ:**

### **7.1. Перечень основной и дополнительной литературы**

Основная литература

1. Васильев Ф.П. Численные методы решения экстремальных задач. М.: Наука, 1988.
2. КORTE Б., Фиген Й. Комбинаторная оптимизация. Теория и алгоритмы. М: МЦНМО, 2015.
3. Пильщиков В.Н. Программирование на языке ассемблера IBM PC. М., Диалог-МИФИ, 2003.
4. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. М.:Наука, 1988
5. Пильщиков В.Н. Упражнения по языку ассемблера MASM. М., МГУ, 2007
6. Баула В.Г., Томилин А.Н., Волканов Д.Ю. Архитектура ЭВМ и операционные среды. – М.: Академия, 2012.
7. Баула В.Г. Введение в архитектуру ЭВМ и системы программирования. Учебное пособие. Электронный ресурс. <http://arch32.cs.msu.su/>
8. Бордаченкова Е. А. Модельные ЭВМ. М., МАКС Пресс, 2012
9. Бордаченкова Е. А., Панфёров А. А. Задания практикума. 1 курс. М., МАКС Пресс, 2016
10. Н.В.Вдовикина, И.В.Машечкин, А.Н.Терехин, А.Н.Томилин. Операционные системы: взаимодействие процессов. Москва, Макс-Пресс, 2008.
11. Н.В.Вдовикина, И.В.Машечкин, А.Н.Терехин, В.В.Тюляева. Программирование в ОС UNIX на языке Си. Москва, Макс-Пресс, 2009.
12. Л.Н.Кузина. Сборник задач по практикуму: язык Си, ОС UNIX. Учебно-методическое пособие для студентов бакалавриата 2 курса, обучающихся по направлению «Фундаментальные информатика и информационные технологии», 2014.
13. И. А. Волкова, А. В. Иванов, Л. Е. Карпов. Основы объектно-ориентированного программирования. Язык программирования C++. Учебное пособие для студентов 2 курса. — М.: Издательский отдел факультета ВМК МГУ, 2011.
14. И. А. Волкова, А. А. Вылиток, Т. В. Руденко.Формальные грамматики и языки. Элементы теории трансляции (3-е издание). — М.: Изд-во МГУ, 2009.
15. А.В.Столяров. Введение в язык C++. Учебное пособие.- 4 изд. – М.МАКС Пресс, 2018.
16. И. А. Волкова, А. А. Вылиток, Л.Е. Карпов. Сборник задач и упражнений по языку C++. Учебное пособие для студентов II курса.— М.: Издательский отдел факультета ВМК МГУ, 2013.
17. РуденкоТ.В. Интерпретатор модельного языка программирования.
18. Ю.С. Корухова. Сборник задач и упражнений по языку C++. Учебное пособие для студентов-бакалавров II курса, обучающихся по направлению «Информационные технологии». — М.: Издательский отдел факультета ВМК МГУ, 2009.

Дополнительная литература

1. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. М.: Мир, 1982.

2. Измаилов А.Ф., Солодов М.В. Численные методы оптимизации. 2-е изд., перераб. и доп. М.: Физматлит, 2008.
3. Иоффе А.Д., Тихомиров В.М. Теория экстремальных задач. М.: Наука, 1974.
4. Карманов В.Г. Математическое программирование. М.: Физматлит, 2000.
5. Вирт Н. Алгоритмы и структуры данных. СПб.: Невский диалект, 2001
6. Pascal ISO 7185:1990 – <http://www.moorecad.com/standardpascal/iso7185.pdf>
7. Баула В. Г., Бордаченкова Е. А. Задачи письменного экзамена по курсу "Архитектура ЭВМ ..." М., МАКС Пресс, 2019.
8. Д. Кнут. Искусство программирования. Том 1 – Основные алгоритмы. – М., изд-во Вильямс, 2005.
9. Д. Кнут. Искусство программирования. Том 3 – Сортировка и поиск. – М., изд-во Вильямс, 2005.
10. Б.Керниган, Д.Ритчи. Язык программирования Си. Второе издание, переработанное и дополненное, ИД Вильямс, 2019
11. А.Робачевский. Операционные системы UNIX. 2-е изд., перераб. и доп. Санкт-Петербург, ВНУ-Санкт-Петербург, 2010.
12. У.Стивенс. UNIX: взаимодействие процессов. Санкт-Петербург, Питер, 2002.
13. Страуструп Б. Язык программирования C++. Специальное изд./Пер. с англ. - М.: "Бином", 2015.
14. Шилдт Г.. Самоучитель C++. – СПб, ВНУ, 2006.
15. Д. Грис. Конструирование компиляторов для цифровых вычислительных машин. — М.: Мир, 1975.
16. Мейерс С. Эффективное использование C++. 50 рекомендаций по улучшению ваших программ и проектов. Питер, ДМК Пресс, Москва, 2006
17. Страуструп Б. Программирование: принципы и практика использования C++.: Пер. с англ. – М. ООО «И.Д.Вильямс», 2016.

## **7.2.Перечень лицензионного программного обеспечения, в том числе отечественного производства**

При реализации дисциплины может быть использовано следующее программное обеспечение:

- Операционная система Windows
- Операционная система Debian Linux
- Программное обеспечение для подготовки слайдов лекций MS PowerPoint, MS Word
- Программное обеспечение для создания и просмотра pdf-документов Adobe Reader
- Издательская система LaTeX
- Язык программирования Python и среда разработки Jupiter Notebook (вместе с библиотеками numpy, scikit-learn, pandas)
- Язык программирования R и среда разработки R Studio
- Файловый архиватор 7z. Свободно-распространяемое ПО
- Браузеры Google Chrome, Mozilla Firefox. Свободно-распространяемое ПО
- Офисный пакет LibreOffice. Свободно-распространяемое ПО
- Visual Studio Community Интегрированная среда разработки ПО. Свободно-распространяемое ПО
- PyCharm Community Интегрированная среда разработки ПО. Свободно-распространяемое ПО
- Anaconda Интегрированная среда разработки ПО. Свободно-распространяемое ПО

## **7.3.Перечень профессиональных баз данных и информационных справочных систем**

1. <http://www.edu.ru> – портал Министерства образования и науки РФ
2. <http://www.ict.edu.ru> – система федеральных образовательных порталов «ИКТ в образовании»

3. <http://www.openet.ru> - Российский портал открытого образования
4. <http://www.mon.gov.ru> - Министерство образования и науки Российской Федерации
5. <http://www.fasi.gov.ru> - Федеральное агентство по науке и инновациям

#### **7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Math-Net.Ru [Электронный ресурс] : общероссийский математический портал / Математический институт им. В. А. Стеклова РАН ; Российская академия наук, Отделение математических наук. - М. : [б. и.], 2010. - Загл. с титул. экрана. - Б. ц.

URL: <http://www.mathnet.ru>

2. Университетская библиотека Online [Электронный ресурс] : электронная библиотечная система / ООО "Директ-Медиа" . - М. : [б. и.], 2001. - Загл. с титул. экрана. - Б. ц. URL: [www.biblioclub.ru](http://www.biblioclub.ru)

3. Универсальные базы данных EastView [Электронный ресурс] : информационный ресурс / EastViewInformationServices. - М. : [б. и.], 2012. - Загл. с титул. экрана. - Б. ц.

URL: [www.ebiblioteka.ru](http://www.ebiblioteka.ru)

4. Научная электронная библиотека eLIBRARY.RU [Электронный ресурс] : информационный портал / ООО "РУНЭБ" ; Санкт-Петербургский государственный университет. - М. : [б. и.], 2005. - Загл. с титул. экрана. - Б. ц.

URL: [www.eLibrary.ru](http://www.eLibrary.ru)

#### **7.5. Описание материально-технического обеспечения.**

Образовательная организация, ответственная за реализацию данной Программы, располагает соответствующей материально-технической базой, включая современную вычислительную технику, объединенную в локальную вычислительную сеть, имеющую выход в Интернет. Используются специализированные компьютерные классы, оснащенные современным оборудованием. Материальная база соответствует действующим санитарно-техническим нормам и обеспечивает проведение всех видов занятий (лекционных, практических, семинарских, лабораторных, дисциплинарной и междисциплинарной подготовки) и научно-исследовательской работы обучающихся, предусмотренных учебным планом.

## **8. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ**

### **8.1. Формы и методы преподавания дисциплины**

Используемые формы и методы обучения: лекции и лабораторные работы, самостоятельная работа студентов.

В процессе преподавания дисциплины преподаватель использует как классические формы и методы обучения (лекции и практические занятия), так и активные методы обучения.

При проведении лекционных занятий преподаватель использует аудиовизуальные, компьютерные и мультимедийные средства обучения, а также демонстрационные и наглядно-иллюстрационные (в том числе раздаточные) материалы.

Семинарские (практические) занятия по данной дисциплине проводятся с использованием компьютерного и мультимедийного оборудования, при необходимости - с привлечением полезных Интернет-ресурсов и пакетов прикладных программ.

### **8.2. Методические рекомендации преподавателю**

Перед началом изучения дисциплины преподаватель должен ознакомить студентов с видами учебной и самостоятельной работы, перечнем литературы и интернет-ресурсов, формами текущей и промежуточной аттестации, с критериями оценки качества знаний для итоговой оценки по дисциплине.

При проведении лекций, преподаватель:

- 1) формулирует тему и цель занятия;
- 2) излагает основные теоретические положения;
- 3) с помощью мультимедийного оборудования и/или под запись дает определения основных понятий, расчетных формул;
- 4) проводит примеры из отечественного и зарубежного опыта, дает текущие статистические



данные для наглядного и образного представления изучаемого материала;

5) в конце занятия дает вопросы для самостоятельного изучения.

Во время выполнения заданий в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается невыполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж (консультацию) с определением цели задания, его содержания, сроков выполнения, основных требований к результатам работы, критериев оценки, форм контроля и перечня источников и литературы.

Для оценки полученных знаний и освоения учебного материала по каждому разделу и в целом по дисциплине преподаватель использует формы текущего, промежуточного и итогового контроля знаний обучающихся.

#### **Для семинарских занятий**

Подготовка к проведению занятий проводится регулярно. Организация преподавателем семинарских занятий должна удовлетворять следующим требованиям: количество занятий должно соответствовать учебному плану программы, содержание планов должно соответствовать программе, план занятий должен содержать перечень рассматриваемых вопросов.

Во время семинарских занятий используются словесные методы обучения, как беседа и дискуссия, что позволяет вовлекать в учебный процесс всех слушателей и стимулирует творческий потенциал обучающихся.

При подготовке семинарскому занятию преподавателю необходимо знать план его проведения, продумать формулировки и содержание учебных вопросов, выносимых на обсуждение.

В начале занятия преподаватель должен раскрыть теоретическую и практическую значимость темы занятия, определить порядок его проведения, время на обсуждение каждого учебного вопроса. В ходе занятия следует дать возможность выступить всем желающим и предложить выступить тем слушателям, которые проявляют пассивность.

Целесообразно, в ходе обсуждения учебных вопросов, задавать выступающим и аудитории дополнительные и уточняющие вопросы с целью выяснения их позиций по существу обсуждаемых проблем, а также поощрять выступление с места в виде кратких дополнений. На занятиях проводится отработка практических умений под контролем преподавателя

#### **Для практических занятий**

Подготовка преподавателя к проведению практического занятия начинается с изучения исходной документации и заканчивается оформлением плана проведения занятия.

На основе изучения исходной документации у преподавателя должно сложиться представление о целях и задачах практического занятия и о том объеме работ, который должен выполнить каждый обучающийся. Далее можно приступить к разработке содержания практического занятия. Для этого преподавателю (даже если он сам читает лекции по этому курсу) целесообразно вновь просмотреть содержание лекции с точки зрения предстоящего практического занятия. Необходимо выделить понятия, положения, закономерности, которые следует еще раз проиллюстрировать на конкретных задачах и упражнениях. Таким образом, производится отбор содержания, подлежащего усвоению.

Важнейшим элементом практического занятия является учебная задача (проблема), предлагаемая для решения. Преподаватель, подбирая примеры (задачи и логические задания) для практического занятия, должен представлять дидактическую цель: привитие каких навыков и умений применительно к каждой задаче установить, каких усилий от обучающихся она потребует, в чем должно проявиться творчество студентов при решении данной задачи.

Преподаватель должен проводить занятие так, чтобы на всем его протяжении студенты были заняты напряженной творческой работой, поисками правильных и точных решений, чтобы каждый получил возможность раскрыться, проявить свои способности. Поэтому при планировании занятия и разработке индивидуальных заданий преподавателю важно учитывать

подготовку и интересы каждого студента. Педагог в этом случае выступает в роли консультанта, способного вовремя оказать необходимую помощь, не подавляя самостоятельности и инициативы студента.

### **8.3. Методические рекомендации студентам по организации самостоятельной работы.**

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке рекомендованные учебники и учебно-методические пособия, осуществить запись на соответствующий курс в среде электронного обучения университета.

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

### **Методические указания для обучающихся по подготовке к семинарским занятиям**

Для того чтобы семинарские занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на семинарских занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач.

При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

При подготовке к семинарским занятиям следует использовать основную литературу из представленного списка, а также руководствоваться приведенными указаниями и рекомендациями. Для наиболее глубокого освоения дисциплины рекомендуется изучать литературу, обозначенную как «дополнительная» в представленном списке.

### **Методические указания для обучающихся по подготовке к практическим занятиям**

Целью практических занятий по данной дисциплине является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.

### **Методические указания для самостоятельной работы обучающихся**

Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к практическим занятиям, к коллоквиуму, контрольным работам по отдельным темам дисциплины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для самостоятельной работы бюджетом времени.

Результатом самостоятельной работы является прочное усвоение материалов по предмету согласно программы дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления.

#### **Решение задач**

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи).

Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты.

Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

Задача — это цель, заданная в определенных условиях, решение задачи — процесс достижения поставленной цели, поиск необходимых для этого средств.

Алгоритм решения задач:

1. Внимательно прочитайте условие задания и уясните основной вопрос, представьте процессы и явления, описанные в условии.
2. Повторно прочтите условие для того, чтобы чётко представить основной вопрос, проблему, цель решения, заданные величины, опираясь на которые можно вести поиски решения.
3. Произведите краткую запись условия задания.
4. Если необходимо составьте таблицу, схему, рисунок или чертёж.
5. Определите метод решения задания, составьте план решения.
6. Запишите основные понятия, формулы, описывающие процессы, предложенные заданной системой.
7. Найдите решение в общем виде, выразив искомые величины через заданные.
9. Проверьте правильность решения задания.
10. Произведите оценку реальности полученного решения.
11. Запишите ответ.