

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова»

«Утверждаю»

Декан факультета ВМК МГУ
имени М.В. Ломоносова

академик



Е. И. Моисеев

2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Основы программной инженерии»

Уровень высшего образования – подготовка научно-педагогических кадров в аспирантуре

Направление подготовки–09.06.01 «Информатика и вычислительная техника»

Направленность (профиль) – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» (05.13.11)

2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

1. НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ

Основы программной инженерии

2. УРОВЕНЬ ВЫСШЕГО ОБРАЗОВАНИЯ

Подготовка научно-педагогических кадров в аспирантуре.

3. НАПРАВЛЕНИЕ ПОДГОТОВКИ, НАПРАВЛЕННОСТЬ (ПРОФИЛЬ) ПОДГОТОВКИ

Направление 09.06.01 «Информатика и вычислительная техника». Направленность (профиль) «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» (05.13.11).

4. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к специальным дисциплинам вариативной части образовательной программы.

5. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Дисциплина участвует в формировании следующих компетенций образовательной программы:

Формируемые компетенции	Планируемые результаты обучения
Владением методологией теоретических и экспериментальных исследований в области профессиональной деятельности (ОПК-1)	З1 (ОПК-1) ЗНАТЬ: современные математические методы, применяющиеся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий У1 (ОПК-1) УМЕТЬ: применять современные методы постановки и анализа задач в области математики и информатики В1 (ОПК-1) ВЛАДЕТЬ: навыками оптимального выбора современных методов и средств постановки и анализа задач в области математики и информатики

<p>Способность разрабатывать и реализовывать алгоритмы организации работы современных вычислительных комплексов и компьютерных сетей</p> <p>(ПК-2)</p>	<p>З1 (ПК-2) ЗНАТЬ: современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения</p> <p>У1(ПК-2) УМЕТЬ: применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения</p> <p>В1 (ПК-2) ВЛАДЕТЬ: навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения</p>
<p>Способность к реализации различных математических алгоритмов в виде программных комплексов, ориентированных на современную вычислительную технику</p> <p>(ПК-4)</p>	<p>З1 (ПК-4)ЗНАТЬ: современные методы реализации различных математических алгоритмов в виде программных комплексов, особенности современных вычислительных комплексов</p> <p>У1(ПК-4) УМЕТЬ: применять современные методы реализации различных математических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов</p> <p>В1 (ПК-4) ВЛАДЕТЬ: навыками оптимального выбора и создания новых современных методов-реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов</p>

<p>Владение современными методами разработки архитектуры компьютерных сетей (ПК-5)</p>	<p>ЗНАТЬ: современные методы разработки архитектуры компьютерных сетей Код З1 (ПК-5) УМЕТЬ: применять современные методы разработки архитектуры компьютерных сетей Код У1 (ПК-5) ВЛАДЕТЬ: базовыми навыками выбора методов разработки архитектуры компьютерных сетей Код В1 (ПК-5)</p>

Оценочные средства для промежуточной аттестации приведены в Приложении.

6. ОБЪЕМ ДИСЦИПЛИНЫ

Объем дисциплины составляет 3 зачетных единицы, всего 108 часов.

42 часа составляет контактная работа с преподавателем – 36 часов занятий лекционного типа, 0 часов занятий семинарского типа (семинары, научно-практические занятия, лабораторные работы и т.п.), 0 часов индивидуальных консультаций, 2 часа мероприятий текущего контроля успеваемости, 2 часа групповых консультаций, 2 часа мероприятий промежуточной аттестации.

66 часов составляет самостоятельная работа аспиранта.

7. ВХОДНЫЕ ТРЕБОВАНИЯ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Учащиеся должны владеть знаниями по алгоритмам, алгоритмическим языкам и программированию в объеме, соответствующем основным образовательным программам бакалавриата и магистратуры по укрупненным группам направлений и специальностей 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки».

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе обучения на отдельных лекционных занятиях проводятся письменные опросы. В рамках опросов аспиранты составляют ответы на текстовые вопросы и/или решают задачи по материалам предыдущих лекций. При изложении материала лекций предполагается диалог со слушателями. На каждой лекции выделяется время для ответов на вопросы по текущему материалу и его обсуждения. Материалы лекций демонстрируются аспирантам в виде презентаций, сопровождаемых комментариями лектора. По ходу чтения конспекты прочитанных лекций выкладываются на специально созданном учебном сайте в сети Интернет. Дополнительно каждый аспирант может дистанционно получить разъяснения преподавателя по электронной почте.

9. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Освоение дисциплины «Основы программной инженерии» обеспечивает слушателей необходимыми знаниями и навыками в области современных технологий создания программного обеспечения. В рамках курса закрепляется основа, обязательная для профессиональной деятельности системного программиста, системного аналитика, архитектора и разработчика программного обеспечения.

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе								
		Контактная работа (работа во взаимодействии с преподавателем), часы					Самостоятельная работа обучающегося, часы			
		из них					из них			
Занятия лекционного типа	Занятия семинарского типа	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др)*	Всего	Выполнение домашних заданий	Подготовка рефератов и т. п.	Всего		
1. Тема I. <i>Предмет и основные принципы программной инженерии</i>	4	2	0	0	0	0	2	2	0	2

<i>рии</i> (Проблемы разработки сложного программного обеспечения. Предмет программной инженерии. Общие принципы программной инженерии: абстракция и уточнение, модульность, повторное использование.)										
2. Тема II. Жизненный цикл программного обеспечения и процессы его разработки (Основные модели жизненного цикла: водопадная, итеративная, спиральная. Международные стандарты, описывающие требования к процессам жизненного цикла: ISO 12207, ISO 15288, ISO 15504, IEEE 1074. Модели процессов разработки CMM и CMMI.Примеры процессов разработки. Унифицированный процесс Rational. Гибкие подходы к разработке программного обеспечения. Экстремальное программирование.)	6	4	0	0	0	0	4	2	0	2
3. Тема III. Управление разработкой программного обеспечения (Виды деятельности, входящие в управление проектами. Управление содержанием проекта и качеством. Планирование и управление ресурсами. Особенности управления разработкой программного обеспечения. Метрики программного обеспечения и их использование.)	6	4	0	0	0	0	4	2	0	2

Управление рисками. Управление персоналом. Структура и культура организаций. Мотивация работников. Построение сплоченной команды. Управление конфликтами. Ведение переговоров и информационное обеспечение проектов.)										
4. Текущий контроль успеваемости: письменный опрос №1	5	0	0	0	0	1	1	4		
5. Тема IV. Анализ требований к программному обеспечению (Анализ предметной области. Требования. Стандарты, определяющие характеристики требований: IEEE 830, IEEE 1233. Методы выделения требований. Методы описания и систематизации требований. Диаграммы потоков данных и диаграммы вариантов использования. Описание требований в виде вариантов использования.)	8	5	0	0	0	0	5	3	0	3
6. Тема V. Качество программного обеспечения и методы его контроля (Понятие качества программного обеспечения и его основные характеристики. Стандарты ISO 9000, ISO 9126, ISO 25010. Методы контроля качества программного обеспечения. Верификация и валидация. Проверка моделей. Формальная верификация.	8	5	0	0	0	0	5	3	0	3

Тестирование. Основные виды тестирования. Полнота тестирования. Основные методы построения тестов.)										
7. Текущий контроль успеваемости: письменный опрос №2	5	0	0	0	0	1	1	4		
8. Тема VI. Архитектура программного обеспечения и ее проектирование (Понятие архитектуры программного обеспечения. Методы разработки и анализа архитектуры программного обеспечения на основе сценариев его работы и модификации. Описание архитектуры. Универсальный язык моделирования UML. Основные виды диаграмм UML и их элементы. Образцы анализа, архитектурные стили, образцы проектирования, идиомы и образцы организации работ. Описание образцов. Системы образцов. Примеры образцов и их использования.)	10	6	0	0	0	0	6	4	0	4
9. Тема VII. Удобство использования программного обеспечения (Психологические и физиологические факторы работы человека с программным обеспечением. Принципы и методы разработки удобного программного обеспечения. Методы контроля удобства использования.)	6	4	0	0	0	0	4	2	0	2

10. Тема VIII. Компонентные технологии разработки распределенных программных систем (Общие принципы компонентных технологий и разработки распределенных программных систем. Основные понятия компонентных технологий: компоненты, компонентная модель, базовые службы. Общие принципы построения распределенных систем: прозрачность, открытость, масштабируемость, безопасность. Образцы организации взаимодействия в распределенных системах: синхронное и асинхронное взаимодействие, транзакции. Организация распределенных транзакций. Современные технологии разработки распределенных компонентных приложений. Инфраструктура Интернет и Web-приложения. Технологии J2EE и .NET. Web-службы.)	10	6	0	0	0	0	6	4	0	4
11. Промежуточная аттестация: устный экзамен	40	0	0	2	0	2	4	36		
Итого	108	36	0	2	0	4	42	66		

10. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ УЧАЩИХСЯ

Для обеспечения самостоятельной работы слушателей можно использовать литературу к курсу и материалы практикума по разработке Web-приложений (страница <http://sites.google.com/site/cmcwtpractice>).

Список тем для заданий по разработке Web-приложений

1. Система информации об автобусных рейсах и билетах
2. Театральная касса
3. Система информации о спортивных соревнованиях
4. Интернет-магазин бытовой техники
5. Книжный Интернет-магазин
6. Информационная система автосалона
7. Складской учет
8. Учебное расписание
9. Учебный центр
10. Система информации о персонале компании
11. Кадровое агентство
12. Зарплатная ведомость
13. Клиентская база юридической фирмы
14. Биллинговая база оператора связи
15. Система информации о счетах клиентов банка
16. Библиотека
17. Web-форум
18. Видеопрокат
19. Система информации об авиарейсах и билетах
20. Система генеалогической информации
21. Система информации о структуре собственности
22. Астрономический каталог
23. Коллекция минералов
24. Информационная система заповедника
25. Агентство недвижимости

Пример формулировки задания по разработке Web-приложения

Система информации об автобусных рейсах и билетах

Система должна поддерживать управление информацией об автобусных рейсах, наличии билетов и ценах на них, а также продажи билетов.

Поддерживаемые данные

- Рейсы
 - o Компания, номер
 - o Пункты убытия и прибытия, промежуточные остановки
 - o Время и даты всех остановок
 - o Стоимость билетов для всех пар остановок
 - o Количество мест и наличие свободных мест с учетом промежуточных остановок
- Клиенты
 - o ФИО
 - o Контактная информация: адрес, телефон, e-mail
 - o Заказанные билеты

Поддерживаемые операции

- Получение списка рейсов по датам, направлениям и промежуточным остановкам, информации о ценах билетов и наличии свободных мест
- Получение списка клиентов, в т.ч. ехавших определенным рейсом, любыми рейсами компании, заказавших билеты
- Получение истории заказов клиента
- Заказ билетов на выбранный рейс между выбранными пунктами
- Добавление и удаление рейса, чтение и редактирование данных о нем
- Добавление и удаление клиента, чтение и редактирование данных о нем

Требования к выполнению задания по разработке Web-приложения (общие для всех вариантов)

Процесс создания приложения состоит из 3-х этапов, по окончании каждого из которых производится промежуточная сдача задания:

- 1) Детальное определение требований. Проектирование базы данных. Проектирование интерфейса системы.
- 2) Разработка кода промежуточного уровня, связывающего базу данных и обработку запросов пользователей.
- 3) Разработка Web-интерфейса и реализация обработки запросов.

Процесс создания приложения проходит по следующему плану.

- I. Первый этап
 1. Создание концептуальной модели пользовательского интерфейса
 - Определение сценариев использования
 - Определение набора страниц, навигации между ними и их содержимого
 - Результаты
 - o Перечень сценариев использования

Описание каждого сценария с указанием, на каких страницах какие действия выполняются

- о Перечень страниц приложения

Для каждой страницы: модель содержимого - какие данные присутствуют на странице и какие действия можно делать

- о Схема навигации между страницами

2. Создание базы данных приложения

- Определение набора таблиц, полей и их типов, связей между таблицами
- Определение отображения данных БД на страницы пользовательского интерфейса
- Разработка скриптов создания БД и ее инициализации

- Результаты

- о Схема БД

- о SQL-скрипт создания БД

- о SQL-скрипт примерного заполнения БД

- о Файл сборки Ant или Maven с задачами создания и инициализации БД, выдачи ее содержимого и очистки

3. Составление документации по первому этапу.

Она должна включать текстовый отчет, а также 2 SQL-скрипта (для создания и инициализации БД) и файл сборки Ant или Maven. При инициализации в каждой таблице должно быть не менее 3-5 записей (если таблиц мало - 2-3, то около 10). Если есть поля перечислимых типов, должны быть записи с каждым возможным значением такого поля.

Текст документации должен содержать

- Перечень и описания сценариев использования, по шагам.
- Перечень и описание страниц, какая информация на странице показывается и какие операции можно выполнить.
- Схему навигации - между какими страницами есть переходы.
- Схему БД - какие таблицы, какие у них поля и связи. Для полей с неясными названиями нужны пояснения, что они обозначают.

II. Второй этап

1. Разработка связывающего кода с помощью Hibernate

- Разработка классов хранимых объектов
- Разработка отображения БД на объекты в Hibernate
- Разработка служебных классов (DAO-классов), методы которых реализуют типовые запросы приложения к БД
- Разработка модульных тестов для методов служебных классов
- Результаты
- о Код классов хранимых объектов и служебных классов

- o Тесты в TestNG для всех методов созданных классов
 - o Конфигурационный файл Hibernate, определяющий отображение БД на объекты
 - o Дополненный файл сборки Ant с задачами сборки проекта, сборки тестов, выполнения тестов
2. Составление документации по второму этапу.

Результаты работы должны быть помещены в SVN-репозиторий студента.

Тесты должны проверять все методы служебных и хранимых классов с нетривиальной логикой, т.е. могут не проверяться только простейшие read-write методы. Методы хранимых классов могут проверяться не прямо, а только через обращающиеся к ним служебные методы.

Кроме того, если метод имеет несколько вариантов поведения или может возвращать разные результаты (нашел что-то нужное в БД или ничего не нашел), должны быть тесты, проверяющие все такие варианты.

В тестах должны проверяться (с помощью методов assert) все свойства полученных результатов, которые можно проверить в соответствующей ситуации.

III. Третий этап

1. Разработка Web-интерфейса и реализация обработки запросов
 - Создание JSP-страниц для всех определенных страниц Web-интерфейса
 - Создание классов-контроллеров с помощью Spring
 - Разработка системных тестов
 - Результаты
 - o Код классов-контроллеров
 - o Код JSP-страниц
 - o Конфигурационный файл Spring
 - o Конфигурация основного сервлета приложения
 - o Дополненный файл сборки Ant с задачей полной сборки и развертывания приложения
 - o Описание сценариев системного тестирования, системные тесты, созданные с помощью Selenium (или HTTPUnit).

Системные тесты должны проверять выполнение всех вариантов использования со всеми существенно различными результатами их работы (например, удачное добавление какого-либо объекта - один тест, неудачное добавление - по одному тесту для каждого вида сообщений, которые при этом можно получить).

2. Составление итогового отчета

Указания по составлению итогового отчёта по практическому заданию

Итоговый отчёт пишется на русском языке. Вёрстку можно осуществлять в любой подходящей для Вас системе. Текст отчёта должен быть разбит на следующие части:

- Титульный лист, с «шапкой» – «Московский государственный университет имени М. В. Ломоносова, факультет Вычислительной математики и кибернетики». Далее следует заголовок: «Отчёт по практическому заданию», номер и тема варианта задания, сведения об исполнителе (фамилия, имя и отчество полностью, номер группы). Внизу титульного листа указывается город и год. Точки после заголовков не ставятся.
- Содержание, которое состоит из перечня названий глав и подглав, сопровождаемых указанием номеров страниц, с которых они начинаются. Нумеруются все страницы, за исключением титульного листа. Номер страницы с содержанием: 2.
- Первая глава, названная «Постановка задачи», содержит формулировку варианта задания. Каждую главу следует начинать с новой страницы.
- Вторая глава, названная «Первый этап задания», содержит всю документацию, подготовленную по первому этапу. Полный состав этой документации дан в требованиях к выполнению задания.
- Третья глава, названная «Второй этап задания», содержит всю документацию, подготовленную по второму этапу. Полный состав этой документации дан в требованиях к выполнению задания.
- Четвёртая глава, названная «Третий этап задания», содержит всю документацию, подготовленную по третьему этапу. Полный состав этой документации дан в требованиях к выполнению задания.
- Заключение (которое не нумеруется, но номер на странице ставится), где подводятся общий итог работы, завершает отчёт. В заключении можно указать характеристики разработанного Web-приложения, оценивается трудоёмкость выполнения задания, делаются выводы по выполненному заданию.
- Список использованной литературы приводится, если в ходе работы над заданием были использованы статьи и/или книги. Библиографические записи в списке следует оформлять по рекомендациям ГОСТ. Сделать это можно при помощи Google.Scholar, который умеет импортировать по ГОСТ. На каждую запись списка в тексте отчёта должна быть ссылка.

11.РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ

Основная литература

1. И. Соммервилл. Инженерия программного обеспечения. М.: Вильямс, 2002.
2. Е. А. Жоголев. Лекции по технологии программирования: Учебное пособие. М.: Издательский отдел факультета ВМК МГУ, 2001.
3. Г. Буч, Дж. Рамбо, А. Джекобсон. Язык UML. Руководство пользователя. М.: ДМК, 2000.
4. А. Якобсон, Г. Буч, Дж. Рамбо. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002.
5. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влоссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер-ДМК, 2001.

6. В. В. Кулямин. Технологии программирования. Компонентный подход. М.: Интернет-университет информационных технологий, Бином, 2007.

Дополнительная литература

1. М. Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley, 1997.
2. Дж. Рамбо, А. Якобсон, Г. Буч. UML: Специальный справочник. СПб.: Питер, 2002.
3. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. Второе издание. М.: Бином, СПб.: Невский диалект, 2000.

Ресурсы информационно-телекоммуникационной сети «Интернет»

1. Software Engineering Body of Knowledge, 2005. http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf.
2. Документация по платформе J2EE, версии 5. <http://java.sun.com/j2ee/5.0/index.jsp>.
3. Документация по платформе .NET. <http://msdn.microsoft.com/library/default.asp>.
4. Веб-страница курса <http://se-course.narod.ru>.
5. Сайт с материалами по практическому заданию <http://sites.google.com/site/cmcwtpractice>

Информационные технологии, используемые в процессе обучения

1. Программное обеспечение для подготовки слайдов лекций OpenOfficeImpress.
2. Программное обеспечение для создания и просмотра pdf-документов AdobeReader.
3. Платформа Java 2 Enterprise Edition <https://www.oracle.com/java/technologies/java-ee-glance.html>
4. Веб-браузер для доступа к материалам, размещённым в WWW (MozillaFirefoxили аналогичный).
5. Программа-клиент электронной почты для онлайн-консультаций с лектором (MozillaThunderbirdили аналогичный).

Активные и интерактивные формы проведения занятия

№ п/п	Тип занятия или внеаудиторной работы	Вид и тематика (название) интерактивного занятия
1	Лекции по темам №№ I-VIII.	В рамках каждого лекционного занятия выделяется время (около 15 минут) для интерактивного обсуждения материала.

Материально-техническая база

Для преподавания дисциплины требуется класс, оборудованный маркерной или меловой доской и мультимедийным проектором. Для выполнения домашних заданий, а также для выполнения и сдачи практических заданий необходим персональный компьютер с доступом в Internet и установленной на нём платформой Java 2 Enterprise Edition.

12. ЯЗЫК ПРЕПОДАВАНИЯ

Русский

13. РАЗРАБОТЧИК ПРОГРАММЫ, ПРЕПОДАВАТЕЛИ

доцент, к.ф.-м.н.Кулямин Виктор Вячеславович

**ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ
«Основы программной инженерии»**

Средства для оценивания планируемых результатов обучения, критерии и показатели оценивания приведены ниже.

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	КРИТЕРИИ и ПОКАЗАТЕЛИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) <i>(критерии и показатели берутся из соответствующих карт компетенций, при этом пользуются либо традиционной системой оценивания, либо БРС)</i>					ОЦЕНОЧНЫЕ СРЕДСТВА
	1	2	3	4	5	
	Неудовлетворительно	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично	
ЗНАТЬ: современные математические методы, применяющиеся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий Код 31 (ОПК-1)	Отсутствие знаний	Фрагментарные представления о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	В целом сформированные, но неполные знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Сформированные, но содержащие отдельные пробелы знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Сформированные систематические знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Устный экзамен
УМЕТЬ: применять современные методы постановки анализа задач в	Отсутствие умений	Фрагментарные умения применять современные методы постановки анализа задач в	В целом успешное, но не систематическое умение применять современные	Успешное, но содержащее отдельные пробелы умение применять со-	Сформированное умение применять современные методы постановки анализа задач в	Устный экзамен

области математики и информатики Код У1 (ОПК-1)		области математики и информатики	методы постановки анализа задач в области математики и информатики	временные методы постановки анализа задач в области математики и информатики	области математики и информатики	
ВЛАДЕТЬ: навыками оптимального выбора современных методов и средств постановки анализа задач в области математики и информатики Код В1 (ОПК-1)	Отсутствие навыков	Фрагментарное владение навыками оптимального выбора современных методов и средств постановки анализа задач в области математики и информатики	В целом успешное, но не полное владение навыками оптимального выбора современных методов и средств постановки анализа задач в области математики и информатики	Успешное, но содержащее отдельные пробелы владения навыками оптимального выбора современных методов и средств постановки анализа задач в области математики и информатики	Сформированное владение навыками оптимального выбора современных методов и средств постановки анализа задач в области математики и информатики	Устный экзамен
ЗНАТЬ: современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения Код З1 (ПК-2)	Отсутствие знаний	Фрагментарные представления о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом сформированные, но неполные знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированные, но содержащие отдельные пробелы знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированные систематические знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Устный экзамен
УМЕТЬ: применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Отсутствие умений	Фрагментарные умения применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом успешное, но не систематическое умение применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Успешное, но содержащее отдельные пробелы умения применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированное умение применять современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Отчет

Код У1 (ПК-2)			плексов и компьютерных сетей последнего поколения	комплексов и компьютерных сетей последнего поколения		
ВЛАДЕТЬ: навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения Код В1 (ПК-2)	Отсутствие навыков	Фрагментарное владение навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом успешное, но не полное владение навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Успешное, но содержащее отдельные пробелы владение навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированное владение навыками оптимального выбора современных методов разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	отчет
ЗНАТЬ: современные методы реализации различных математических алгоритмов в виде программных комплексов, особенности современных вычислительных комплексов Код З1 (ПК-4)	Отсутствие знаний	Фрагментарные представления о современных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	В целом сформированные, но неполные знания о современных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Сформированные, но содержащие отдельные пробелы знания о современных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Сформированные систематические знания о современных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Устный экзамен
УМЕТЬ: применять современные методы реализации различных математических алгоритмов в виде программ-	Отсутствие умений	Фрагментарные умения применять современные методы реализации различных математических алгоритмов в виде программ-	В целом успешное, но не систематическое умение применять современные методы реализации различных матема-	Успешное, но содержащее отдельные пробелы умение применять современные методы реализации различ-	Сформированное умение применять современные методы реализации различных математических алгоритмов в виде программ-	отчет

ных комплексов с учетом особенностей современных вычислительных комплексов Код У1 (ПК-4)		ных комплексов с учетом особенностей современных вычислительных комплексов	тических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов	ных математических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов	ных комплексов с учетом особенностей современных вычислительных комплексов	
ВЛАДЕТЬ: навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов Код В1 (ПК-4)	Отсутствие навыков	Фрагментарное владение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	В целом успешное, но не полное владение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	Успешное, но содержащее отдельные пробелы владение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	Сформированное владение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	отчет
ЗНАТЬ: современные методы разработки архитектуры компьютерных сетей Код 31 (ПК-5)	Отсутствие знаний	Фрагментарные представления о современных методах разработки архитектуры компьютерных сетей	В целом сформированные, но неполные знания о современных методах разработки архитектуры компьютерных сетей	Сформированные, но содержащие отдельные пробелы знания о современных методах разработки архитектуры компьютерных сетей	Сформированные систематические знания о современных методах разработки архитектуры компьютерных сетей	Устный экзамен
УМЕТЬ: применять современ-	Отсутствие умений	Фрагментарные умения применять современ-	В целом успешное, но не систематиче-	Успешное, но содержащее отдельные	Сформированное умение применять современ-	отчет

ные методы разработки архитектуры компьютерных сетей Код У1 (ПК-5)		менные методы разработки архитектуры компьютерных сетей	ское умение применять современные методы разработки архитектуры компьютерных сетей	ные пробелы умение применять современные методы разработки архитектуры компьютерных сетей	менные методы разработки архитектуры компьютерных сетей	
ВЛАДЕТЬ: базовыми навыками выбора методов разработки архитектуры компьютерных сетей Код В1 (ПК-5)	Отсутствие навыков	Фрагментарное владение базовыми навыками выбора методов разработки архитектуры компьютерных сетей	В целом успешное, но не полное владение базовыми навыками выбора методов разработки архитектуры компьютерных сетей	Успешное, но содержащее отдельные пробелы владение базовыми навыками выбора методов разработки архитектуры компьютерных сетей	Сформированное владение навыками выбора методов разработки архитектуры компьютерных сетей	отчет

Фонды оценочных средств, необходимые для оценки результатов обучения

Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости

Варианты вопросов для письменных опросов

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 С
2. Необходимость разработки архитектуры до перехода к кодированию
3. Много людей вовлечено в создание системы
4. Отсутствие поддержки пользователей
5. Высокая стоимость разработки

2) Перечислите в столбцах таблицы номера интерфейсов из приведенного ниже списка, обладающих соответствующими свойствами.

Адекватность	Полнота	Минимальность	Простота

Интерфейсы для работы с мультимножеством целых чисел (мультимножество реализует неупорядоченную коллекцию, в которую каждый элемент может входить несколько раз)

1.
 - a. `int add(int i)` — добавляет число i , возвращая количество его вхождений в мультимножество после этого
 - b. `int remove(int i)` — удаляет одно вхождение числа i , возвращая количество его вхождений в мультимножество после этого
 - c. `int size()` — возвращает общее количество элементов мультимножества (каждый элемент считается столько раз, какова кратность его вхождения)
2.
 - a. `int add(int i)` — добавляет число i , возвращая количество его вхождений в мультимножество после этого
 - b. `int remove(int i)` — удаляет одно вхождение числа i , возвращая общее количество элементов мультимножества (каждый элемент считается столько раз, какова кратность его вхождения) после этого
3.
 - a. `void add(int i, int k)` — добавляет число i в мультимножество k раз (кратность вхождения i увеличивается на k)
 - b. `void remove(int i, int k)` — удаляет k вхождений числа i или все его вхождения, если их число меньше k (кратность вхождения i уменьшается на k , если эта кратность больше k , иначе становится равной 0)
 - c. `int numberOf(int i)` — возвращает кратность вхождения i в мультимножество
 - d. `int size()` — возвращает общее количество элементов мультимножества (каждый элемент считается столько раз, какова кратность его вхождения)
4.
 - a. `void add(int i, int k)` — добавляет число i в мультимножество k раз (кратность вхождения i увеличивается на k)
 - b. `void remove(int i)` — удаляет все вхождения числа i в мультимножество
 - c. `int numberOf(int i)` — возвращает кратность вхождения i в мультимножество

3) Обозначим различные виды деятельности при разработке ПО следующим образом.

- A. Анализ требований
- B. Проектирование
- C. Кодирование
- D. Тестирование
- E. Развертывание
- F. 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.

<input type="checkbox"/>	ACBCBABCDEF
<input type="checkbox"/>	ABABABCDEF
<input type="checkbox"/>	ABCDEF CDEF

	BACDBCDEF
	ABABCABCDEF
	ABCBADEF
	ABCABABCDEF
	ABCDEABDF

4) Обозначим различные виды деятельности при разработке ПО следующим образом.

- A. Планирование итерации
- B. Определение задач очередной итерации
- C. Оценка рисков
- D. Сбор и анализ требований
- E. Проектирование и верификация проекта
- F. Кодирование и тестирование
- G. Развертывание и приемочное тестирование

Поставьте галочки в первом столбце приведенной ниже таблицы рядом с последовательностями выполнения этих видов деятельности, которые могут возникать при использовании спиральной модели жизненного цикла ПО.

	ABDEABCFAFCG
	ABCDEACBFABCG
	ABCDEABCFCG
	ABCDEABCFAFCFCG
	ABCDEABCDEFABCG
	ABCDFDABCG
	ABCDABCEFCG
	BACDEABCFCG

5) Заполните второй и третий столбцы таблицы, указав в каждой строке максимальный и минимальный уровни СММ, на которых может находиться организация с характеристиками, указанными в первом столбце.

Описание процессов разработки в организации	Максимальный уровень СММ	Минимальный уровень СММ
Используется процесс разработки по модели RUP. При этом руководство считает, что улучшать принятый процесс разработки не надо.		
Используется процесс разработки Team Software Process. Четко определена методика измерения и предварительной оценки трудоемкости проектов.		

Используются несколько методик планирования работ в проектах и оценки качества результатов. Снимаются основные показатели временных и трудовых затрат.		
Ход работ определяется указаниями старшего разработчика проекта, но иногда в него вмешивается вышестоящее руководство.		

б) Заполните следующую таблицу, поместив в первый столбец номера перечисленных ниже техник, используемых при разработке ПО в соответствии с Rational Unified Process (RUP), во второй столбец — номера техник, используемых в Extreme Programming (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
- 7) Отметьте те из указанных элементов, которые входят в технику «программирование парами».

<input type="checkbox"/>	Разработка кода выполняется парой программистов, один из которых пишет код, другой в это время отдыхает.
<input type="checkbox"/>	Разработка кода выполняется парой программистов, один из которых пишет код, другой в это время дает ему советы и думает над улучшением кода.
<input type="checkbox"/>	Пара программистов в течение всего проекта работает вместе.
<input type="checkbox"/>	Объединение программистов в пары меняется от задачи к задаче.
<input type="checkbox"/>	Разработка кода выполняется парой программистов, один из которых отвечает за все, сделанное ими, и руководит работой другого, а тот пишет код.
<input type="checkbox"/>	Разработка кода выполняется парой программистов попеременно, то одним, то другим, они меняются местами каждые 5-10 минут.

8) Отметьте те из перечисленных пунктов, которые правильно описывают тестовые варианты.

<input type="checkbox"/>	Запустить программу подготовки бухгалтерского отчета. Разобраться в том, как с ее помощью создать отчет за последний квартал.
<input type="checkbox"/>	Составить список проблем удобства использования.

	Установить в свойствах экрана Windows маленькие шрифты. Запустить программу и открыть диалог настройки цветов. Надписи на кнопках диалога должны помещаться на них и находиться в центре кнопок. Надписи около полей ввода должны находиться напротив соответствующих им полей, быть видны целиком и не должны налезать на соседнюю надпись сверху или снизу.
	Запустить текстовый редактор. В новом документе набрать фразу «Яр-Тур.~» (без кавычек). Сразу станет ясно, если это Microsoft Word 2000 без установленных обновлений.
	Составить список возможных ошибок в коде (деление на 0, разыменование нулевого указателя, обращение к адресу за пределами массива или буфера и пр.). Раздать его 3-м участникам. Определить каждому для просмотра кусок исходного кода размером до 10000 строк. Каждый в течение недели внимательно просматривает свой участок кода, пытаясь найти места возможных ошибки из списка и заносит их в список. После этого полученные списки объединяются.

9) 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.

10) Поставьте галочки в первом столбце приведенной ниже таблицы рядом с описаниями свойств, которыми должны обладать правильно составленные требования к ПО согласно IEEE 830.

	Возможность предсказания трудоемкости проекта
	Невозможность внесения изменений
	Упорядоченность по лицам, от которых требования поступили
	Отсутствие двусмысленностей
	Максимально возможная детальность
	Явное указание связей между требованиями
	Возможность контроля соблюдения

11) К какой из характеристик качества ПО по ISO 9126 относятся описанные требования? Перечислите номера соответствующих характеристик качества в первом столбце каждой строки с описанием требования.

1. Функциональность
2. Надежность
3. Эффективность, производительность
4. Удобство использования
5. Переносимость
6. Удобство сопровождения

	Обработка событий, связанных с изменением характеристик полета должна занимать не более 40 мс, обработка управляющих событий должна занимать не более 10 мс.
	Все модули системы должны сопровождаться модульными тестами, обеспечивающими покрытие не менее чем 90% ветвей в коде.
	Каждый класс в коде должен быть снабжен комментариями, описывающими все задачи, решаемые его объектами. Каждый метод должен быть снабжен описанием его поведения и всех исключительных ситуаций, возможных в нем, с условиями их возникновения.
	Графический интерфейс пользователя должен быть стандартным интерфейсом в оболочке KDE.
	Система должна быть способна обрабатывать до 3000 транзакций в секунду.
	При доступе к данным должны проверяться права пользователей на выполнение операций соответствующего типа (чтение и запись).
	Система не должна ломаться при вводе пользователем некорректных данных.

12) 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.

13) Индексатор — архитектура 1

Пояснения

Индексатор строит список слов, встречающихся в текстовом документе, упорядоченный по алфавиту и снабженный для каждого слова упорядоченным по возрастанию списком страниц, на которых оно встречается. Слова отождествляются по совпадению их букв с точностью до регистра (т.е. грамматические формы одного слова считаются разными словами).

Парсер — преобразовывает входной документ в список страниц, в котором каждая страница представлена множеством ее различных слов.

Анализатор — обрабатывает страницы друг за другом, собирает все встреченные слова в общую хэш-таблицу, привязывая к каждому слову список страниц, на которых оно встречается.

Сортировщик — сортирует таблицу слов по алфавиту.

Принтер — выдает готовый индекс в виде текста в исходном формате.

Индексатор — архитектура 2

Пояснения

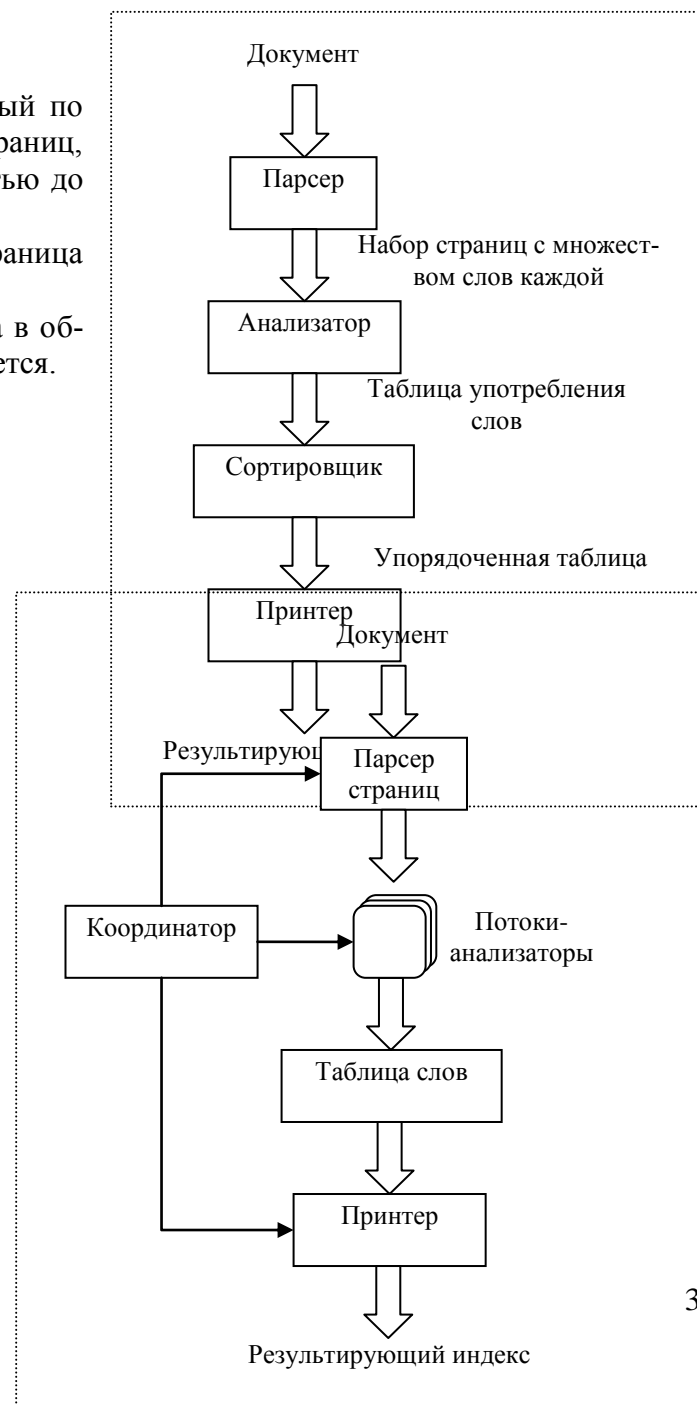
Парсер страниц — выделяет страницы исходного документа.

Координатор — запускает сначала парсер страниц, затем потоки-анализаторы, по одному для каждой страницы исходного документа, следя за тем, чтобы одновременно работало не более M потоков. После окончания работы последнего потока-анализатора, запускает принтер.

Поток-анализатор — анализирует одну страницу, добавляя в таблицу слов найденные слова и информацию об их использовании на данной странице.

Таблица слов — хранит текущую информацию о найденных словах и их использовании на страницах исходного документа. Слова с самого начала хранятся упорядоченными по алфавиту — таблица представляет собой сбалансированное бинарное дерево.

Принтер — выдает готовый индекс в виде текста в исходном формате.



Какая из представленных архитектур демонстрирует более высокие показатели по следующим метрикам качества (оцените их на основании соответствующих сценариев работы или изменения системы).

Первая лучше	Вторая лучше	Одинаковы	
			Производительность — скорость обработки больших документов
			Производительность — время обработки одной страницы
			Модифицируемость — трудоемкость добавления возможности обработки документов в новом формате
			Модифицируемость — трудоемкость добавления возможности обработки архивированных документов
			Модифицируемость — трудоемкость добавления распознавания грамматических форм одного слова
			Переносимость — трудоемкость переноса всего индекса на другую операционную систему

14) Поставьте галочки в первом и втором столбцах таблицы напротив названий архитектурных стилей, использованных в архитектурах, описанных в предыдущей задаче.

Первая архитектура	Вторая архитектура	
		Каналы и фильтры
		Многоуровневая система
		Клиент-сервер
		Данные-представление-обработка
		Представление-абстракция-управление
		Репозиторий
		Абстрактные типы данных
		Замкнутый цикл управления

Типовые контрольные задания или иные материалы для проведения промежуточного контроля успеваемости

Вопросы устного экзамена по курсу

1. Понятие программной системы.
2. 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. 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. Тестирование ПО. Понятие тестирования и характеристики тестов.
8. Характеристики и атрибуты качества ПО по стандарту ISO 9126. Ошибки в ПО.
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.

Методические материалы для проведения процедур оценивания результатов обучения

Система контроля и оценивания

Итоговая оценка по курсу выставляется в зависимости от суммы баллов, набранных слушателем в ходе семестра. Каждый вопрос в рамках письменных опросов оценивается от 1 до 3 баллов. Общая сумма по письменным опросам может составить от 0 до 56 технических баллов. Выполнение практического задания по разработке Web-приложения оценивается суммой от 0 до 44 технических баллов. Составление итогового отчета по практическому заданию по разработке Web-приложения оценивается суммой от 0 до 15 технических баллов. Ответ на устном экзамене оценивается суммой от 0 до 45 технических баллов. Таким образом, максимально возможная сумма набранных технических составляет до 160 технических баллов. Оценка «отлично» ставится студентам, набравшим от 120 баллов и выше. Оценка «хорошо» ставится студентам, набравшим от 90 до 119 технических баллов. Оценка «удовлетворительно» ставится студентам, набравшим от 70 до 89 технических баллов. Оценка «неудовлетворительно» ставится студентам, набравшим менее 70 технических баллов.

Структура и график контрольных мероприятий

Письменный опрос №1 после лекции по теме №III, письменный опрос №2 после лекции по теме №V, отчет по практическому заданию в конце семестра, устный экзамен в конце семестра.