

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ

Руководитель направления

доц., к.т.н., доц.

(должность, уч. степень, звание)

В.А. Матьяш

(инициалы, фамилия)



(подпись)

«13» мая 2019 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Проектирование программных систем»  
(Наименование дисциплины)

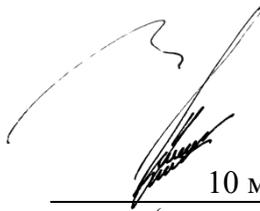
Код направления подготовки/ специальности	02.03.03
Наименование направления подготовки/ специальности	Математическое обеспечение и администрирование информационных систем
Наименование направленности	Системный анализ в информационных технологиях
Форма обучения	очная

Лист согласования рабочей программы дисциплины

Программу составил

старший преподаватель

(должность, уч. степень, звание)



10 мая 2019 г.

(подпись, дата)

Е. В. Павлов

(инициалы, фамилия)

Программа одобрена на заседании кафедры № 43

«13» мая 2019 г, протокол № 08-2018/19

Заведующий кафедрой № 43

д.т.н., проф.

(уч. степень, звание)



13 мая 2019 г.

(подпись, дата)

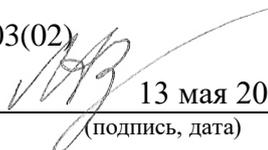
М. Ю. Охтилев

(инициалы, фамилия)

Ответственный за ОП ВО 02.03.03(02)

доц., к.т.н., доц.

(должность, уч. степень, звание)



13 мая 2019 г.

(подпись, дата)

В. А. Матьяш

(инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н., доц.

(должность, уч. степень, звание)



13 мая 2019 г.

(подпись, дата)

А. А. Ключарев

(инициалы, фамилия)

## Аннотация

Дисциплина «Проектирование программных систем» входит в образовательную программу высшего образования по направлению подготовки/ специальности 02.03.03 «Математическое обеспечение и администрирование информационных систем» направленности «Системный анализ в информационных технологиях». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-3 «Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения»

ОПК-4 «Способен участвовать в разработке технической документации программных продуктов и программных комплексов»

Содержание дисциплины охватывает круг вопросов, связанных с процессами разработки ПО в части сбора и анализа требований, проектирования архитектуры и детального проектирования программного продукта, выбора методологии и подходов при проектировании и разработке, в частности предметно-ориентированное проектирование (Domain-Driven Design), тестированием программного продукта на этапах разработки и управления требованиями, составления проектной документации.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося, консультации, курсовое проектирование.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 10 зачетных единиц, 360 часов.

Язык обучения по дисциплине «русский»

## 1. Перечень планируемых результатов обучения по дисциплине

### 1.1. Цели преподавания дисциплины

Целью преподавания данной дисциплины является получение студентами необходимых знаний и навыков в части проектирования программных систем с использованием различных методологий и подходов, инструментальных средств и артефактов проектирования, необходимых для решения задач проектирования и разработки программного продукта.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-3 Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.3.1 знает основные положения и концепции в прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. ОПК-3.У.1 умеет использовать современные информационные технологии в профессиональной деятельности при создании программных продуктов и программных комплексов различного назначения. ОПК-3.Д.1 имеет практические навыки разработки программного обеспечения.
Общепрофессиональные компетенции	ОПК-4 Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ОПК-4.Д.1 имеет практические навыки подготовки технической документации

## 2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Объектно-ориентированное программирование»,
- «Метрология программного обеспечения»,

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

- «Проектирование баз данных»
- «Web-технологии».

### 3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам		
		№4	№5	№6
1	2	3	4	5
<b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>	10/ 360	4/ 144	5/ 180	1/ 36
<b>Аудиторные занятия, всего час.</b>	153	68	68	17
в том числе:				
лекции (Л), (час)	68	34	34	
практические/семинарские занятия (ПЗ), (час)				
лабораторные работы (ЛР), (час)	68	34	34	
курсовой проект (работа) (КП, КР), (час)	17			17
экзамен, (час)	90	36	54	
<b>Самостоятельная работа, всего (час)</b>	117	40	58	19
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Экз., Экз.,	Экз.	Экз.	

Примечание: \*\* кандидатский экзамен

### 4. Содержание дисциплины

#### 4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 4					
Раздел 1. Организация процесса разработки Тема 1.1 Жизненный цикл ПО Тема 1.2 Модели разработки ПО	4	—	2	—	2
Раздел 2. Руководство программным проектом Тема 2.1 Планирование программного проекта Тема 2.2 Управление рисками Тема 2.3 Оценка планирования	2	—	—	—	2
Раздел 3. Формирование и анализ требований Тема 3.1 Разработка требований к ПО Тема 3.2 Риски возникновения недостатков в требованиях Тема 3.3 Представление требований Тема 3.4 Концепция продукта и границы проекта	4	—	2	—	6

Раздел 4. Основы проектирования программных систем Тема 4.1 Особенности проектирования архитектуры систем Тема 4.2 Модульная структура систем Тема 4.3 Рефакторинг	4	—	2	—	4
Раздел 5. Методологии проектирования ПО Тема 5.1 Структурная методология (обзор) Тема 5.2 Объектно-ориентированная методология	16	—	20	—	22
Раздел 6. Проектирование интерфейса	2	—	8	—	2
Раздел 7. Гибкая разработка	2	—	—	—	2
Итого в семестре:	34		34		40
Семестр 5					
Раздел 8. Структурная методология	4	—	14	—	4
Раздел 9. Проектирование структуры данных Тема 9.1 Реляционная парадигма Тема 9.2 Семантическое моделирование Тема 9.3 Нормализация баз данных Тема 9.4 Нереляционные решения	6	—	8	—	8
Раздел 10. Спецификация требований к ПО	4	—	—	—	6
Раздел 11. Тестирование программных систем Тема 11.1 Методика тестирования Тема 11.2 Структурное тестирование Тема 11.3 Функциональное тестирование	4	—	2	—	8
Раздел 12. Архитектура программных систем	14	—	10	—	28
Раздел 13. Обеспечение качества программных систем	1	—	—	—	2
Раздел 14. Автоматизация разработки	1	—	—	—	2
Итого в семестре:	34	—	34	—	58
Семестр 6					
Выполнение курсового проекта	—	—	—	17	19
Итого в семестре:				17	19
Итого	68	0	68	17	117

#### 4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Организация процесса разработки Тема 1.1 Жизненный цикл ПО Тема 1.1.1 Этапы и основные действующие лица Тема 1.1.2 Основные процессы Тема 1.2 Модели разработки ПО Тема 1.2.1 Водопадная модель Тема 1.2.2 VEE-модель Тема 1.2.3 Итерационная модель Тема 1.2.4 Модель гибкой разработки
2	Раздел 2. Руководство программным проектом Тема 2.1 Планирование программного проекта

	<p>Тема 2.1.1 Структура плана управления программным проектом</p> <p>Тема 2.1.2 Структура графика работ программного проекта</p> <p>Тема 2.2 Управление рисками</p> <p>Тема 2.2.1 Анализ рисков</p> <p>Тема 2.2.2 Разрешение рисков</p> <p>Тема 2.3 Оценка планирования</p> <p>Тема 2.3.1 Виды метрик для оценки планирования</p> <p>Тема 2.3.2 Выполнение оценки проекта на основе LOC- и FP метрик</p>
3	<p>Раздел 3. Формирование и анализ требований</p> <p>Тема 3.1 Разработка требований к ПО</p> <p>Тема 3.1.1 Виды требований</p> <p>Тема 3.1.2 Распределение работ с требованиями на протяжении жизненного цикла проекта в разных моделях разработки</p> <p>Тема 3.2 Риски возникновения недостатков в требованиях</p> <p>Тема 3.3 Представление требований</p> <p>Тема 3.3.1 Способы представления требований</p> <p>Тема 3.3.2 Выбор наиболее подходящих способов представления</p> <p>Тема 3.4 Концепция продукта и границы проекта</p> <p>Тема 3.4.1 Способы представления границ проекта</p> <p>Тема 3.4.2 Контекстная диаграмма</p> <p>Тема 3.4.3 Карта экосистемы</p> <p>Тема 3.4.4 Дерево функций</p> <p>Тема 3.4.5 Список событий</p>
4	<p>Раздел 4. Основы проектирования программных систем</p> <p>Тема 4.1 Особенности проектирования архитектуры систем</p> <p>Тема 4.1.1 Структурирование системы</p> <p>Тема 4.1.2 Виды архитектурных решений</p> <p>Тема 4.2 Модульная структура систем</p> <p>Тема 4.3 Рефакторинг</p>
5	<p>Раздел 5. Методологии проектирования ПО</p> <p>Тема 5.1 Структурная методология</p> <p>Тема 5.2 Объектно-ориентированная методология</p> <p>Тема 5.2.1 Метод проектирования предметных областей</p> <p>Тема 5.2.2 Метод наведения мостов</p> <p>Тема 5.2.3 Domain Driven Design</p> <p>Тема 5.2.4 Язык UML</p>
6	<p>Раздел 6. Проектирование интерфейса</p> <p>Тема 6.1 Построение навигационной карты приложения</p> <p>Тема 6.2 UX / UI</p>
7	<p>Раздел 7. Гибкая разработка</p> <p>Тема 7.1 Scrum</p> <p>Тема 7.2 XP</p>
8	<p>Раздел 8. Структурная методология</p> <p>Тема 8.1.1 Метод нисходящего проектирования</p> <p>Тема 8.1.2 Метод восходящего проектирования</p> <p>Тема 8.1.3 Метод расширения ядра</p> <p>Тема 8.1.4 Разработка диаграммы потоков данных</p>
9	<p>Раздел 9. Проектирование структуры данных</p> <p>Тема 9.1 Реляционная парадигма</p> <p>Тема 9.2 Семантическое моделирование</p> <p>Тема 9.3 Нормализация баз данных</p> <p>Тема 9.4 Нереляционные решения</p> <p>Тема 9.4.1 Key-value model</p> <p>Тема 9.4.2 Document Database</p> <p>Тема 9.4.3 Column Store</p> <p>Тема 9.6.4 Graph Database</p>

10	Раздел 10. Спецификация требований к ПО Тема 10.1 ISO/IEC/IEEE 29148:2018 Тема 10.2 ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011
11	Раздел 11. Тестирование программных систем Тема 11.1 Методика тестирования Тема 11.2 Структурное тестирование Тема 11.3 Функциональное тестирование
12	Раздел 12. Архитектура программных систем Тема 12.1 Архитектурные шаблоны и стили Тема 12.2 Проектирование архитектуры на основе предметной области Тема 12.3 Графическое представление архитектуры Тема 12.4 Архитектура клиент/сервер Тема 12.5 Компонентная архитектура Тема 12.6 Многослойная архитектура Тема 12.7 Шаблоны проектирования
13	Раздел 13. Обеспечение качества программных систем Тема 13.1 Факторы качества ПО Тема 13.2 Верификация и валидация
14	Раздел 14. Автоматизация разработки Тема 14.1 Генерация программного кода (прямое проектирование) Тема 14.2 Трансформация программного кода в модель UML (обратное проектирование)

#### 4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего				

#### 4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела дисциплины
Семестр 4			
1	Анализ предметной области. Моделирование функциональных требований к системе. Разработка диаграммы вариантов использования.	8	1, 3, 5
2	Моделирование статического представления системы. Разработка диаграммы классов.	8	4, 5
3	Моделирование потоков управления. Разработка диаграммы последовательности.	8	5
4	Проектирование пользовательского интерфейса. Составление диаграммы навигации по окнам.	10	5, 6

Семестр 5			
5	Функциональное моделирование системы. Разработка диаграммы потоков данных.	8	8
6	Детализация функциональных требований. Составление спецификации процессов.	6	8
7	Моделирование отношений данных. Составление словаря данных.	8	9
8	Проектирование архитектуры программной системы	12	11, 12
Всего		68	

#### 4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсового проекта: изучение способов описания законченного поведения проектируемой программной системы (стандарт ISO/IEC/IEEE 29148:2018)

Примерные темы заданий на курсовой проект приведены в разделе 10 РПД.

#### 4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час	Семестр 5, час	Семестр 6, час
1	2	3	4	5
Курсовое проектирование (КП, КР)	19	—	—	19
Подготовка к текущему контролю успеваемости (ТКУ)	98	40	58	—
Всего:	117	40	58	19

#### 5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

#### 6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8 – Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
ISBN: 978-5-16-011753-9	Назаров, С. В. Архитектура и проектирование программных систем : монография / С.В. Назаров. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2020. — 374 с. — (Научная мысль). — DOI 10.12737/18292. - ISBN 978-5-16-104150-5. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1093643">https://znanium.com/catalog/product/1093643</a> (дата обращения: 29.06.2020). – Режим доступа: по подписке	—
ISBN: 978-5-94774-817-8	Грекул, В. И. Проектирование информационных систем : учебное пособие / В. И. Грекул. — 2-е изд. — Москва	—

	: ИНТУИТ, 2016. — 570 с. — ISBN 978-5-94774-817-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/100391">https://e.lanbook.com/book/100391</a> (дата обращения: 29.06.2020). — Режим доступа: для авториз. пользователей.	
ISBN: 978-5-8199-0649-1	Гагарина, Л. Г. Введение в архитектуру программного обеспечения : учеб. пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. — Москва : ФОРУМ : ИНФРА-М, 2018. — 320 с. — (Высшее образование). - ISBN 978-5-16-104169-7. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/971770">https://znanium.com/catalog/product/971770</a> (дата обращения: 29.06.2020). – Режим доступа: по подписке	—
ISBN: 978-5-7638-2511-4	Антамошкин, О. А. Программная инженерия. Теория и практика [Электронный ресурс] : учебник / О. А. Антамошкин. - Красноярск: Сиб. Федер. ун-т, 2012. - 247 с. - ISBN 978-5-7638-2511-4. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/492527">https://znanium.com/catalog/product/492527</a> (дата обращения: 29.06.2020). – Режим доступа: по подписке	—
ISBN: 978-5-906923-22-6	Введение в программную инженерию : Учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. — Москва : КУРС: ИНФРА-М, 2019. — 336 с. - ISBN 978-5-16-103172-8. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1035160">https://znanium.com/catalog/product/1035160</a> (дата обращения: 29.06.2020). – Режим доступа: по подписке	—

#### 7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
ISBN: 5-94074-334-X	Буч, Г. Язык UML. Руководство пользователя : руководство / Г. Буч, Д. Рамбо, И. Якобсон. — Москва : ДМК Пресс, 2008. — 496 с. — ISBN 5-94074-334-X. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/1246">https://e.lanbook.com/book/1246</a> (дата обращения: 29.06.2020). — Режим доступа: для авториз. пользователей.
ISBN: 5-94074-050-2	Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов : руководство / Д. Розенберг, К. Скотт. — Москва : ДМК Пресс, 2007. — 160 с. — ISBN 5-94074-050-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/1226">https://e.lanbook.com/book/1226</a> (дата обращения: 29.06.2020). — Режим доступа: для авториз. пользователей.

## 8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10 – Перечень программного обеспечения

№ п/п	Наименование
1	Diagrams.net (бесплатный сервис для создания диаграмм) <a href="https://www.diagrams.net/">https://www.diagrams.net/</a>
2	Software Ideas Modeler (бесплатное CASE-средство для разработки и анализа программного обеспечения) <a href="https://www.softwareideas.net/">https://www.softwareideas.net/</a>

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
1	RSDN [Электронный ресурс]: Проектирование ПО / RSDN team. 2001-2020. URL: <a href="http://rsdn.org/?summary/3387.xml">http://rsdn.org/?summary/3387.xml</a>

## 9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	БМ, ауд. 32-04
2	Вычислительная лаборатория	БМ, ауд. 23-10

## 10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Тесты.
Выполнение курсового проекта	Экспертная оценка на основе требований к содержанию курсового проекта.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся глубоко и всесторонне усвоил программный материал;</li> <li>– уверенно, логично, последовательно и грамотно его излагает;</li> <li>– опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления;</li> <li>– умело обосновывает и аргументирует выдвигаемые им идеи;</li> <li>– делает выводы и обобщения;</li> <li>– свободно владеет системой специализированных понятий.</li> </ul>
«хорошо» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы;</li> <li>– не допускает существенных неточностей;</li> <li>– увязывает усвоенные знания с практической деятельностью направления;</li> <li>– аргументирует научные положения;</li> <li>– делает выводы и обобщения;</li> <li>– владеет системой специализированных понятий.</li> </ul>
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы;</li> <li>– допускает несущественные ошибки и неточности;</li> <li>– испытывает затруднения в практическом применении знаний направления;</li> <li>– слабо аргументирует научные положения;</li> <li>– затрудняется в формулировании выводов и обобщений;</li> <li>– частично владеет системой специализированных понятий.</li> </ul>
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> <li>– обучающийся не усвоил значительной части программного материала;</li> <li>– допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении;</li> <li>– испытывает трудности в практическом применении знаний;</li> <li>– не может аргументировать научные положения;</li> <li>– не формулирует выводов и обобщений.</li> </ul>

## 10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена
1	Жизненный цикл ПО. Отличительные особенности ЖЦ при классической разработке и гибкой (agile)
2	Методологии разработки ПО
3	Гибкая модель разработки на примере любого из её фреймворков
4	Разработка и управление требованиями. Типы требований
5	Распределение работ с требованиями на протяжении ЖЦ проекта в разных моделях разработки. Риски возникновения недостатков в требованиях
6	Представление требований

7	Спецификация требований к ПО
8	Описание бизнес-процессов системы (DFD). Составление спецификации процессов
9	Спецификация требований к ПО
10	Определение требований к данным (ERD). Словарь данных
11	Особенности гибкой разработки (agile) в применении к требованиям
12	Инструментальные средства разработки требований
13	Понятие об архитектуре ПО
14	Примеры архитектурных решений
15	Проектирование архитектуры ПО
16	Документирование программной архитектуры
17	Методы анализа архитектуры
18	Архитектурные типовые решения
19	Типовые решения представления данных
20	Основные архитектурные стили
21	Проектирование на основе предметной области (DDD)
22	Применение аналитических шаблонов (analysis patterns)
23	Тестирование документации и требований
24	Составление вариантов тестирования (test case)
25	Применение унифицированного языка моделирования (UML) в проектировании
26	Структура плана управления программным проектом
27	Структура графика работ программного проекта
28	Управление рисками (анализ и разрешение рисков)
29	Оценка планирования
30	Модульная структура систем
31	Рефакторинг
32	Структурная методология
33	Объектно-ориентированная методология
34	Метод проектирования предметных областей
35	Метод наведения мостов
36	Основные виды диаграмм языка UML
37	Построение навигационной карты приложения

38	Основные принципы UX / UI дизайна
39	Реляционная парадигма
40	Методы семантического моделирования
41	Нормализация баз данных (приведение к 1-3 нормальным формам)
42	Нереляционные решения (отличие от реляционной парадигмы на примере Key-value model)
43	Нереляционные решения (Key-value model, Document Database, Column Store, Graph Database)
44	Методика тестирования программных систем
45	Структурное и функциональное тестирование
46	Архитектурные шаблоны и стили
47	Архитектура клиент/сервер
48	Компонентная архитектура
49	Многослойная архитектура
50	Шаблоны проектирования (разобрать любые четыре из представленных в конспекте лекций)
51	Факторы качества ПО
52	Верификация и валидация
53	Автоматизация разработки (прямое и обратное проектирование)

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета
	Учебным планом не предусмотрено

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Перечень вопросов (задач) для экзамена
1	Онлайн-бронирование билетов (железнодорожных, авиа, автобусных etc.)
2	Онлайн-бронирование номеров в гостиницах
3	Онлайн-бронирование театральных билетов (театральная касса)
4	Онлайн-бронирование столов в ресторанах и ночных клубах
5	Онлайн-бронирование туров
6	Онлайн аренда автомобилей
7	Онлайн-путеводитель по городам и странам
8	Планировщик маршрутов общественного транспорта
9	Управление аэропортом

10	Управление железнодорожным вокзалом
11	Управление грузовыми перевозками
12	Географическая информационная система
13	Управление больницей
14	Электронная регистрация
15	Медицинский информационно-аналитический портал
16	Сервис для изучения иностранных языков
17	Биржа труда (Интернет-рекрутмент)
18	Система федеральной налоговой службы
19	Система миграционного учета (включая сервисы по вопросам миграции)
20	Система дактилоскопического учёта и хранения криминальных досье
21	Система поиска пропавших без вести людей
22	Мониторинг и учет преступлений в сфере информационных технологий
23	Центральная избирательная комиссия
24	Оплата государственных услуг
25	Веб-сервис мониторинга информации о дорожных пробках
26	Умное здание (Smart Building)
27	Умный дом (Smart Home)
28	Автоматизация процессов на предприятиях общественного питания
29	Система контроля кассовых операций для магазина
30	Управление образовательным процессом в школе
31	Управление образовательным процессом в высшем учебном заведении (электронный деканат)
32	Дистанционное обучение (виртуальная школа)
33	Обучающий портал (курсы и вебинары)
34	Онлайн курсы подготовки к ЕГЭ и ОГЭ
35	Музыкальная школа онлайн
36	Управление студенческим общежитием
37	Управление музеем
38	Виртуальные экскурсии по музеям
39	Художественная онлайн-галерея
40	Система контроля и управления доступом (СКУД)
41	Управление процессами охранного предприятия
42	Веб-портал для отдела кадров
43	Онлайн-страхование (прямое страхование)
44	Юридические консультации онлайн
45	Платформа для организации работы компании (виртуальный офис)
46	Управление взаимоотношениями с клиентами (клиентская база и call-центр)
47	Сервис для сбора клиентских отзывов
48	Родительский контроль
49	Агентство недвижимости

50	Благотворительный фонд
51	Краудфандинговая платформа
52	Букмекерская контора
53	Торговля акциями на бирже
54	Сервис для проведения Интернет-аукционов
55	Сервис для проведения конкурса красоты
56	Виртуальная служба знакомств
57	Интернет-ателье
58	Научно-популярный портал
59	Проведение научных конференций
60	Астрономическая Интернет-обсерватория
61	Астрофизическая интерактивная база данных
62	Видеостриминговый сервис
63	Киберспортивный портал
64	Игровой портал
65	Клиент-серверное приложение для MMORPG
66	База знаний для MMORPG
67	Сервис покупки и продажи игровой валюты (биржа игровых ценностей)
68	Клиент-серверное приложение для фарма игровой валюты (MMORPG)
69	Клиент-серверное приложение для MOBA
70	Проведение чемпионата по MOBA
71	Клиент-серверное приложение для MMO-Action
72	Игровой сервис для совместной игры через Интернет
73	Игровая социальная сеть с обеспечением многопользовательской игры и услуг связи
74	Онлайн-кинотеатр
75	База данных о фильмах
76	Корпоративная социальная сеть
77	Веб-сервис социальных закладок
78	Сервис вопросов и ответов с элементами социальной сети
79	Сервис для обмена фотографиями и видеозаписями с элементами социальной сети
80	Сервис для создания онлайн дневников (блогов)
81	Сервис для мгновенного обмена сообщениями
82	Веб-сервис для тайм-менеджмента
83	Сервис поиска и прослушивания музыки онлайн (служба потокового аудио)
84	Сервис мониторинга музыкальных новинок
85	Онлайн платформа для продвижения музыкантов и исполнителей
86	Платформа для онлайн-трансляции концертов
87	Творческий портал
88	Социальная сеть для дизайнеров
89	Торговая площадка для покупки и продажи авторских хендмейд работ и

	дизайнерских вещей
90	Аниме портал
91	Система управления массиванным ответным ядерным ударом (Dead Hand)
92	Система контроля и управления атомной станцией
93	Интернет-казино
94	Видеохостинг
95	Облачное хранилище данных
96	Веб-сервис для хостинга IT-проектов и их совместной разработки
97	Управление марсоходом (планетоходом)
98	Фильтрация содержимого Интернета
99	Углубленное наблюдение за Интернет-трафиком
100	Веб-сервис по организации платного троллинга
101	Сервис для владельцев домашних питомцев (выгул, передержка, доставка корма, etc.)
102	Сервис для продвижения в социальных сетях (SMM)
103	Площадка для обеспечения прямых поставок (Drop Shipping System)

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
1	В рамках задач жизненного цикла ПО укажите самый распространённый механизм аккумуляции требований (в основном в гибких моделях разработки). Приведите шаблон данного инструмента описания требований.
2	Какой из документов позволяет оценить риски, стоимость, временные затраты, структурировать проблемы и оптимизировать процесс разработки? На каком этапе жизненного цикла ПО данный документ составляется и что является его аналогом?
3	В чем заключается обеспечение качества кода и на каком этапе жизненного цикла ПО оно осуществляется? Относится ли к данной функции задача фиксации изменений в проекте при его модификации (если нет, то частью какой функции является указанная задача)?
4	В чем отличие сопровождения от поддержки ПО? В каком случае заканчивается жизненный цикл ПО: когда вышла новая версия или когда разработчик перестал поддерживать приложение (поясните ответ)?
5	В какой из моделей разработки ПО после перехода на следующий этап жизненного цикла, нельзя вернуться к предыдущему? Затягиваются ли сроки сдачи проекта в данной модели в результате частых изменений требований, например, на этапе разработки (поясните ответ)?
6	В какой из моделей разработки ПО после каждого этапа жизненного цикла осуществляется контроль качества? Учитываются ли в данной модели риски, связанные с частыми изменениями требований, например, на этапе разработки (поясните ответ)?
7	Что понимается под прототипированием и основным понятием какой из моделей разработки оно является? Укажите второе основное понятие данной модели и приведите для него примеры.

8	В рамках какой модели разработки ПО могут применяться разные методологии процесса разработки ПО (поясните, какие именно)? Какова основная задача данной модели?
9	Для какой из моделей разработки ПО характерна задача сокращения цикла обратной связи ( <i>feedback loop</i> )? При реализации данной модели на крупных проектах, в чем выражается основной риск (проблема)?
10	Что понимают под фреймворком модели разработки ПО? Приведите примеры фреймворков для определенной модели разработки.
11	В рамках задач жизненного цикла ПО укажите самый распространённый механизм аккумуляции требований (в основном в гибких моделях разработки)? Затягиваются ли сроки сдачи проекта в каскадной модели в результате частых изменений требований, например, на этапе разработки (поясните ответ)?
12	В каком случае заканчивается жизненный цикл ПО: когда вышла новая версия или когда разработчик перестал поддерживать приложение (поясните ответ)? Учитываются ли в VEE-модели риски, связанные с частыми изменениями требований, например, на этапе разработки (поясните ответ)?
13	Какой из документов позволяет оценить риски, стоимость, временные затраты, структурировать проблемы и оптимизировать процесс разработки? Что понимают под фреймворком модели разработки ПО? Приведите примеры фреймворков для гибкой модели разработки.
14	В чем заключается обеспечение качества кода и на каком этапе жизненного цикла ПО оно осуществляется? Для какой из моделей разработки ПО характерна задача сокращения цикла обратной связи ( <i>feedback loop</i> )? При реализации данной модели на крупных проектах, в чем выражается основной риск (проблема)?
15	В каком случае заканчивается жизненный цикл ПО: когда вышла новая версия или когда разработчик перестал поддерживать приложение (поясните ответ)? В рамках какой модели разработки ПО могут применяться разные методологии процесса разработки ПО (поясните, какие именно)? Какова основная задача данной модели?
16	В какой из моделей разработки ПО после перехода на следующий этап жизненного цикла, нельзя вернуться к предыдущему? Затягиваются ли сроки сдачи проекта в данной модели в результате частых изменений требований, например, на этапе разработки (поясните ответ)? В какой из моделей разработки ПО после каждого этапа жизненного цикла осуществляется контроль качества? Учитываются ли в данной модели риски, связанные с частыми изменениями требований, например, на этапе разработки (поясните ответ)?
17	Что понимают под бизнес-требованиями и как связаны между собой бизнес-требования и бизнес-правила? Что включают в себя бизнес-правила?
18	Какой тип требований, как правило, представляется в виде вариантов использования, пользовательских историй или таблиц по типу «событие-отклик»? Включают ли данный вид требований в виде пользовательских историй в спецификацию требований к ПО (поясните ответ)?
19	Какой тип требований описывает характер поведения системы? Перечислите типы требований, которые выступают подмножествами данного типа требований.
20	Какой тип требований представляет собой правила, которые накладывают границы на возможности выбора разработчика при проектировании и разработке продукта? Приведите конкретный пример такого требования.
21	Чем отличается распределение работ с требованиями на протяжении жизненного цикла ПО в проектах с каскадной ( <i>waterfall model</i> ) и спиральной моделях

	разработки? В каких проектах, как правило, используется каскадная модель (точнее вариации данной модели)?
22	Чем отличается распределение работ с требованиями на протяжении жизненного цикла ПО в проектах с каскадной ( <i>waterfall model</i> ) и гибкой ( <i>agile</i> ) моделях разработки? Почему <i>agile</i> менее всего подходит для больших команд и крупных проектов?
23	Чем отличается распределение работ с требованиями на протяжении жизненного цикла ПО в проектах с гибкой ( <i>agile</i> ) и спиральной моделях разработки? В каких проектах практически не используется спиральная модель (поясните ответ)?
24	Какие последствия могут иметь место в следующих ситуациях: 1) Добавление функций, которые не были прописаны в спецификации, но, по мнению разработчика, могли бы понравиться пользователям? 2) Недостаточное вовлечение пользователей в процесс разработки.
25	По какой причине формальные спецификации практически не используются как способ представления требований? Какие способы считаются наиболее подходящими для представлений внешних интерфейсов системы?
26	Какой из способов представления требований обладает наивысшей точностью? Какие способы считаются наиболее подходящими для определения данных и связей объектов данных?
27	Какой из способов документирования требований используется для большинства проектов по разработке ПО? Какие способы считаются наиболее подходящими для описания задач пользователей?
28	Дайте характеристику границ проекта и укажите, через что они определяются? Какой из способов представления границ проекта помимо разрабатываемой системы показывает другие системы, которые с ней связаны?
29	Что понимают под концепцией продукта и почему концепция продукта менее динамична чем границы проекта? Какой из способов представления границ проекта может включать в себя сигналы внешних компонентов, таких как аппаратные устройства?
30	Что понимают под бизнес-требованиями и как связаны между собой бизнес-требования и бизнес-правила? Какие способы считаются наиболее подходящими для представлений внешних интерфейсов системы?
31	Какой тип требований представляет собой правила, которые накладывают границы на возможности выбора разработчика при проектировании и разработке продукта (приведите конкретный пример такого требования)? Какие способы считаются наиболее подходящими для представлений нефункциональных требований?
32	Что понимают под бизнес-требованиями и как связаны между собой бизнес-требования и бизнес-правила? Что понимают под концепцией продукта и почему концепция продукта менее динамична чем границы проекта? Какой из способов представления границ проекта может включать в себя сигналы внешних компонентов, таких как аппаратные устройства?
33	Какой тип требований, как правило, представляется в виде вариантов использования, пользовательских историй или таблиц по типу «событие-отклик»? Включают ли данный вид требований в виде пользовательских историй в спецификацию требований к ПО (поясните ответ)? Дайте характеристику границ проекта и укажите, через что они определяются? Какой из способов представления границ проекта помимо разрабатываемой системы показывает другие системы, которые с ней связаны?
34	Какой тип требований описывает характер поведения системы? Перечислите типы требований, которые выступают подмножествами данного типа

	<p>требований.</p> <p>Какой из способов документирования требований используется для большинства проектов по разработке ПО? Какие способы считаются наиболее подходящими для описания задач пользователей?</p>
35	<p>Какой тип требований представляет собой правила, которые накладывают границы на возможности выбора разработчика при проектировании и разработке продукта? Приведите конкретный пример такого требования.</p> <p>Какой из способов представления требований обладает наивысшей точностью? Какие способы считаются наиболее подходящими для определения данных и связей объектов данных?</p>
36	<p>Чем отличается распределение работ с требованиями на протяжении жизненного цикла ПО в проектах с каскадной (<i>waterfall model</i>) и спиральной моделях разработки? В каких проектах, как правило, используется каскадная модель (точнее вариации данной модели)?</p> <p>По какой причине формальные спецификации практически не используются как способ представления требований? Какие способы считаются наиболее подходящими для представлений внешних интерфейсов системы?</p>
37	<p>Какими проблемам сопровождается организация долгосрочного хранения данных посредством файлов и что выступает их решением? Какой способ хранения данных противопоставляется долгосрочному хранению и по какой причине данные, хранение которых осуществляется указанным способом, носят временный характер?</p>
38	<p>Какой тип систем управления БД исключает наличие выделенного сервера и поставляется вместе с приложением, не требуя установки и профессионального администрирования? Описание чего включает в себя схема БД?</p>
39	<p>Какой тип систем управления БД предполагает работу системы на нескольких серверах одновременно? Какой язык и графическая модель данных, как правило, выступают средствами описания схемы реляционной БД?</p>
40	<p>Какой тип системы управления БД предполагает работу и размещение всех компонентов системы на одном сервере? Какие из нижеперечисленных положений для реляционной модели данных не являются верными (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. В каждой строке данные одного объекта;</li> <li>2. Столбцы описывают свойства этого объекта;</li> <li>3. Все строки таблицы имеют разную структуру;</li> <li>4. Для каждого поля строки нет заранее определенного типа данных.</li> </ol>
41	<p>Что понимают под масштабирование сервера? Какие из нижеперечисленных положений не являются верными (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Реляционные БД хорошо масштабируются только в том случае, если располагаются на единственном сервере;</li> <li>2. В отличие от реляционных БД, все нереляционные решения, включают в себя ряд общих признаков;</li> <li>3. Реляционные БД благодаря своей гибкой структуре хорошо подходят для задачи хранения разнородных данных;</li> <li>4. В нереляционных СУБД нет фиксированной схемы данных.</li> </ol>

42	Какой тип NoSQL БД осуществляет хранение данных в виде разреженной матрицы (приведите примеры СУБД для данного типа)? Какие из нижеперечисленных характеристик для SQL и NoSQL решений не являются верными (приведите корректные формулировки или пояснение):		
	<i>N</i>	<i>SQL</i>	<i>NoSQL</i>
	1	База состоит из таблиц, таблицы содержат строки и колонки, все строки одной таблицы имеют одинаковую структуру	Схема данных определена заранее, существуют отношения и ограничения для обеспечения целостности данных
	2	Схема данных заранее не определена и не имеет строгой типизации	Записи в БД могут иметь разную структуру
3	Используется нормализация для устранения избыточности	Данные разбиты по таблицам и требуют значительный объем кода для маппинга в структуры приложения	
43	Какой тип NoSQL БД осуществляет хранение данных в виде узлов и связей между ними (приведите примеры СУБД для данного типа)? Какие из нижеперечисленных характеристик для SQL и NoSQL решений не являются верными (приведите корректные формулировки или пояснение):		
	<i>N</i>	<i>SQL</i>	<i>NoSQL</i>
	1	Используется нормализация для устранения избыточности	Нет схемы данных, ограничений, отношений и соответственно контроля целостности
	2	Отношения между записями не заданы явно	Используется единый для всех БД язык запросов
3	Хранение данных осуществляется в том же виде, в котором они хранятся в приложении	У каждой записи в БД динамичный набор атрибутов	

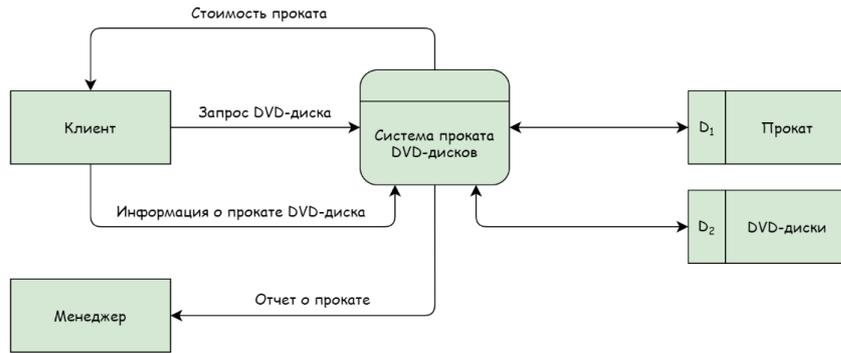
44	<p>Какой тип NoSQL БД осуществляет хранение данных в виде ассоциативного массива (приведите примеры СУБД для данного типа)? Какие из нижеперечисленных характеристик для SQL и NoSQL решений не являются верными (приведите корректные формулировки или пояснение):</p> <table border="1" data-bbox="347 333 1479 757"> <thead> <tr> <th data-bbox="347 333 405 389">N</th> <th data-bbox="405 333 938 389">SQL</th> <th data-bbox="938 333 1479 389">NoSQL</th> </tr> </thead> <tbody> <tr> <td data-bbox="347 389 405 488">1</td> <td data-bbox="405 389 938 488">Схема данных определена заранее и имеет строгую типизацию</td> <td data-bbox="938 389 1479 488">Заданы строгие ограничения для обеспечения целостности данных</td> </tr> <tr> <td data-bbox="347 488 405 616">2</td> <td data-bbox="405 488 938 616">Данные хранятся в разных таблицах и для связи таблиц используются отношения</td> <td data-bbox="938 488 1479 616">Для обеспечения выборки данных используются простые условия с применением оператора <i>join</i></td> </tr> <tr> <td data-bbox="347 616 405 757">3</td> <td data-bbox="405 616 938 757">У каждой записи в БД динамичный набор атрибутов</td> <td data-bbox="938 616 1479 757">Данные денормализованы и между записями отсутствуют отношения, либо не заданы явно</td> </tr> </tbody> </table>	N	SQL	NoSQL	1	Схема данных определена заранее и имеет строгую типизацию	Заданы строгие ограничения для обеспечения целостности данных	2	Данные хранятся в разных таблицах и для связи таблиц используются отношения	Для обеспечения выборки данных используются простые условия с применением оператора <i>join</i>	3	У каждой записи в БД динамичный набор атрибутов	Данные денормализованы и между записями отсутствуют отношения, либо не заданы явно
N	SQL	NoSQL											
1	Схема данных определена заранее и имеет строгую типизацию	Заданы строгие ограничения для обеспечения целостности данных											
2	Данные хранятся в разных таблицах и для связи таблиц используются отношения	Для обеспечения выборки данных используются простые условия с применением оператора <i>join</i>											
3	У каждой записи в БД динамичный набор атрибутов	Данные денормализованы и между записями отсутствуют отношения, либо не заданы явно											
45	<p>Какой тип NoSQL БД осуществляет хранение данных в виде иерархических древовидных структур (приведите примеры СУБД для данного типа)? Какие из нижеперечисленных характеристик для SQL и NoSQL решений не являются верными (приведите корректные формулировки или пояснение):</p> <table border="1" data-bbox="347 994 1479 1541"> <thead> <tr> <th data-bbox="347 994 405 1050">N</th> <th data-bbox="405 994 938 1050">SQL</th> <th data-bbox="938 994 1479 1050">NoSQL</th> </tr> </thead> <tbody> <tr> <td data-bbox="347 1050 405 1200">1</td> <td data-bbox="405 1050 938 1200">Все данные хранятся в одной таблице, поэтому необходима нормализация для устранения избыточности</td> <td data-bbox="938 1050 1479 1200">Отсутствует единый API для управления данными</td> </tr> <tr> <td data-bbox="347 1200 405 1328">2</td> <td data-bbox="405 1200 938 1328">Все строки одной таблицы имеют одинаковую структуру</td> <td data-bbox="938 1200 1479 1328">Нет схемы данных, ограничений, отношений и соответственно контроля целостности</td> </tr> <tr> <td data-bbox="347 1328 405 1541">3</td> <td data-bbox="405 1328 938 1541">Для нормализации данных используется оператор <i>join</i></td> <td data-bbox="938 1328 1479 1541">Данные могут храниться примерно в том же виде, в котором они хранятся в приложении, поэтому необходим маппинг реляционной структуры базы в структуру приложения</td> </tr> </tbody> </table>	N	SQL	NoSQL	1	Все данные хранятся в одной таблице, поэтому необходима нормализация для устранения избыточности	Отсутствует единый API для управления данными	2	Все строки одной таблицы имеют одинаковую структуру	Нет схемы данных, ограничений, отношений и соответственно контроля целостности	3	Для нормализации данных используется оператор <i>join</i>	Данные могут храниться примерно в том же виде, в котором они хранятся в приложении, поэтому необходим маппинг реляционной структуры базы в структуру приложения
N	SQL	NoSQL											
1	Все данные хранятся в одной таблице, поэтому необходима нормализация для устранения избыточности	Отсутствует единый API для управления данными											
2	Все строки одной таблицы имеют одинаковую структуру	Нет схемы данных, ограничений, отношений и соответственно контроля целостности											
3	Для нормализации данных используется оператор <i>join</i>	Данные могут храниться примерно в том же виде, в котором они хранятся в приложении, поэтому необходим маппинг реляционной структуры базы в структуру приложения											
46	<p>Какой тип NoSQL БД осуществляет хранение данных в виде узлов и связей между ними (приведите примеры СУБД для данного типа)? Какие из нижеперечисленных положений не являются верными (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Реляционные БД хорошо масштабируются только в том случае, если располагаются на единственном сервере;</li> <li>2. В отличие от реляционных БД, все нереляционные решения, включают в себя ряд общих признаков;</li> <li>3. Реляционные БД благодаря своей гибкой структуре хорошо подходят для задачи хранения разнородных данных;</li> <li>4. В нереляционных СУБД нет фиксированной схемы данных.</li> </ol>												
47	<p>В каких проектах можно отказаться от составления спецификации требований к ПО (SRS)? Какие из нижеперечисленных положений не являются верными</p>												

	<p>(поясните, почему данные положения неверны, и приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. SRS позволяет получить точную оценку стоимости и затрат времени, поскольку подвержена частому изменению в процессе пересмотра требований;</li> <li>2. SRS помогает сформировать одинаковое представление о продукте у заказчика и исполнителя;</li> <li>3. SRS позволяет структурировать проблемы и оптимизировать процесс разработки;</li> <li>4. SRS помогает оценить полный объем работ по тестированию, включая определение критериев начала и окончания тестирования.</li> </ol>
48	<p>В каком разделе или подразделе SRS приводят контекстную диаграмму или карту экосистемы? Какие из нижеперечисленных положений не являются верными по отношению к SRS (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Структура раздела функций системы зависит от выделенных классов пользователей;</li> <li>2. Словарь данных может храниться отдельно от SRS;</li> <li>3. Информация по обслуживанию данных выходит за рамки SRS;</li> <li>4. SRS может включать в себя в том числе юридические и законодательные требования.</li> </ol>
49	<p>Какой элемент в описании варианта использования (ВИ) представляет собой условие, которое инициирует выполнение ВИ? Какие из нижеперечисленных положений не являются верными по отношению к ВИ (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Список предварительных условий — обязательный элемент описания ВИ;</li> <li>2. Основное действующее лицо — реальный человек или система;</li> <li>3. Альтернативный поток — условия, которые препятствуют успешному выполнению ВИ;</li> <li>4. Выходные условия — описывают состояние системы после успешного выполнения ВИ.</li> </ol>
50	<p>Какой элемент в описании варианта использования (ВИ) представляет собой условия, которое препятствуют успешному выполнению ВИ? Какие из нижеперечисленных положений не являются верными по отношению к ВИ (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Основной поток описывает сценарий, который с точки зрения пользователя, должен происходить по умолчанию;</li> <li>2. Условие-триггер не является частью предварительных условий;</li> <li>3. Описание ВИ должно содержать как минимум одно предварительное условие;</li> <li>4. Выходные условия описывают только наблюдаемое поведение (например, информационное сообщение) или физические результаты (например, печать чека).</li> </ol>
51	<p>Какой элемент в описании варианта использования (ВИ) содержит условия, которые определяют, возможно ли выполнение ВИ? Какие из нижеперечисленных положений не являются верными по отношению к ВИ (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. В описании ВИ могут отсутствовать выходные условия;</li> <li>2. Основной поток описывает сценарий, который с точки зрения пользователя, должен происходить по умолчанию;</li> <li>3. Условие-триггер не является частью предварительных условий;</li> <li>4. Исключения представляют собой менее популярные или менее приоритетные вариации самой задачи или способа её выполнения.</li> </ol>
52	<p>В чем отличие логической DFD от физической? По какой причине DFD 3-го и последующих уровней встречается достаточно редко?</p>

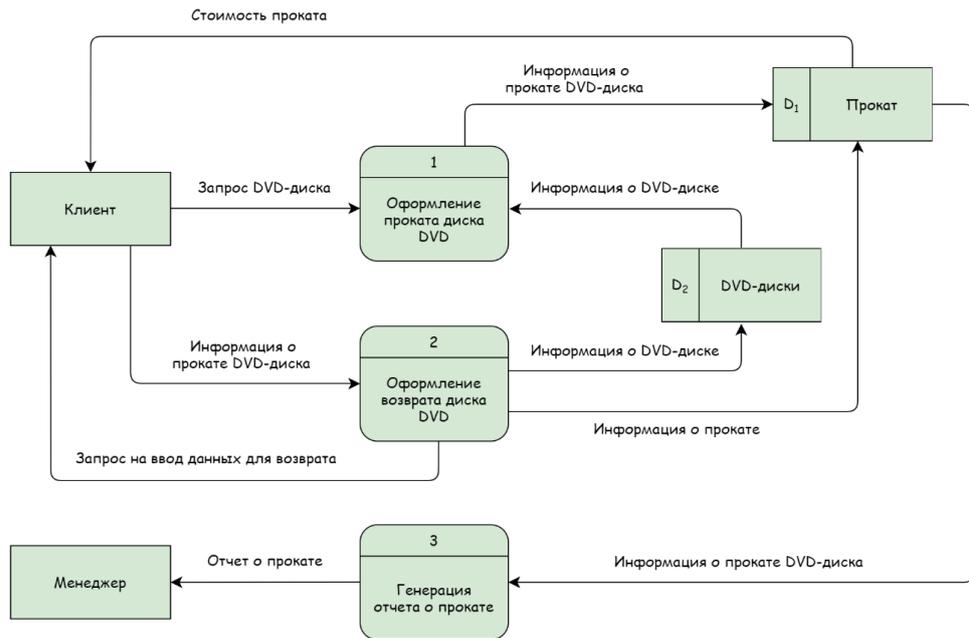
53	<p>В чём отличие между 1 и 2 уровнями DFD? Какие из нижеперечисленных положений не являются верными по отношению к спецификации процессов DFD (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Процессы нижнего уровня сопровождаются спецификацией, когда достигается требуемая глубина декомпозиции;</li> <li>2. Описание логики процесса осуществляется в виде DFD 2-го уровня;</li> <li>3. Способ представления спецификации процессов зависит от уровня DFD;</li> <li>4. Спецификация процессов (для некоторых DFD) может сопровождаться указанием связанных с процессами подпрограмм и функций.</li> </ol>
54	<p>Каким образом словарь данных повышает качество разработки ПО? Какие из нижеперечисленных положений не являются верными по отношению к словарю данных (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Словарь данных определяет имена переменных в программе;</li> <li>2. Необязательные элементы (в словаре данных), значение которых не должно предоставляться пользователем или системой, заключают в скобки;</li> <li>3. Словарь данных представляет собой централизованное хранилище подробной информации о внешних по отношению к системе источниках и адресах данных;</li> <li>4. Все элементы данных, которые представлены на ERD, подробно описывает словарь данных, предназначенный для сбора, документирования и систематизации функциональных требований к системе.</li> </ol>
55	<p>С помощью какой диаграммы UML можно представить навигационную карту пользовательского интерфейса? Какие из нижеперечисленных положений не являются верными по отношению к диаграмме навигации по окнам (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Стереотипы состояния могут обозначать различные типы окон и другие GUI-объекты, которые обладают определенным временем жизни;</li> <li>2. Графически стереотип показан как имя, заключённое в фигурные скобки;</li> <li>3. Диаграмма навигации по окнам для GUI (Graphical User Interface) в качестве основы использует диаграмму видов деятельности UML (activity diagram), а для WUI (Web User Interface) — диаграмму состояний UML (statechart diagram);</li> <li>4. Список стереотипов состояний и видов деятельности для проектирования диаграммы навигации по окнам строго закреплён и не зависит от выбранного типа пользовательского интерфейса.</li> </ol>
56	<p>Каким образом словарь данных повышает качество разработки ПО? В чём отличие логической DFD от физической?</p>
57	<p>Какой элемент в описании варианта использования (ВИ) содержит условия, которые определяют, возможно ли выполнение ВИ? В чём отличие между 1 и 2 уровнями DFD?</p>
58	<p>С помощью какой диаграммы UML можно представить навигационную карту пользовательского интерфейса? Какие из нижеперечисленных положений не являются верными (поясните, почему данные положения неверны, и приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. SRS помогает сформировать одинаковое представление о продукте у заказчика и исполнителя;</li> <li>2. SRS позволяет получить точную оценку стоимости и затрат времени, поскольку подвержена частому изменению в процессе пересмотра требований;</li> <li>3. SRS помогает оценить полный объём работ по тестированию, включая определение критериев начала и окончания тестирования.</li> <li>4. SRS позволяет структурировать проблемы и оптимизировать процесс разработки;</li> </ol>
59	<p>Каким образом словарь данных повышает качество разработки ПО?</p>

	<p>Какие из нижеперечисленных положений не являются верными по отношению к SRS (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Словарь данных может храниться отдельно от SRS;</li> <li>2. Структура раздела функций системы зависит от выделенных классов пользователей;</li> <li>3. SRS может включать в себя в том числе юридические и законодательные требования.</li> <li>4. Информация по обслуживанию данных выходит за рамки SRS;</li> </ol>
60	<p>Какой элемент в описании варианта использования (ВИ) представляет собой условие, которое инициирует выполнение ВИ?</p> <p>Какие из нижеперечисленных положений не являются верными по отношению к словарю данных (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Словарь данных представляет собой централизованное хранилище подробной информации о внешних по отношению к системе источниках и адресатах данных;</li> <li>2. Все элементы данных, которые представлены на ERD, подробно описывает словарь данных, предназначенный для сбора, документирования и систематизации функциональных требований к системе.</li> <li>3. Словарь данных определяет имена переменных в программе;</li> <li>4. Необязательные элементы (в словаре данных), значение которых не должно предоставляться пользователем или системой, заключают в скобки;</li> </ol>
61	<p>В чём отличие между 1 и 2 уровнями DFD?</p> <p>Какие из нижеперечисленных положений не являются верными по отношению к ВИ (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Основное действующее лицо — реальный человек или система;</li> <li>2. Список предварительных условий — обязательный элемент описания ВИ;</li> <li>3. Выходные условия — описывают состояние системы после успешного выполнения ВИ.</li> <li>4. Альтернативный поток — условия, которые препятствуют успешному выполнению ВИ;</li> </ol>
62	<p>Какой элемент в описании варианта использования (ВИ) содержит условия, которые определяют, возможно ли выполнение ВИ?</p> <p>Какие из нижеперечисленных положений не являются верными по отношению к спецификации процессов DFD (приведите корректные формулировки):</p> <ol style="list-style-type: none"> <li>1. Описание логики процесса осуществляется в виде DFD 2-го уровня;</li> <li>2. Процессы нижнего уровня сопровождаются спецификацией, когда достигается требуемая глубина декомпозиции;</li> <li>3. Спецификация процессов (для некоторых DFD) может сопровождаться указанием связанных с процессами подпрограмм и функций.</li> <li>4. Способ представления спецификации процессов зависит от уровня DFD;</li> </ol>
63	<p>Какие ошибки были допущены при составлении представленной DFD? Поясните, какой вариант диаграммы будет корректным (словами — чертить ничего не нужно). Примечание: отсутствие каких-либо потоков данных, хранилищ или процессов не является ошибкой — на DFD представлена только релевантная информация</p>

DFD Level 0 (контекстная диаграмма)



DFD Level 1 (декомпозиция контекстной диаграммы)



(данная задача имеет 18 вариаций)

Какие ошибки были допущены при составлении представленного фрагмента словаря данных? Примечание: в словаре некоторые данные денормализованы для упрощения маппинга и ускорения работы базы данных — в представленном примере это не является ошибкой; также обратите внимание, что это именно фрагмент словаря, а не весь словарь данных, соответственно некоторые таблицы здесь не показаны, что также не является ошибкой).

64

Структура или элемент данных	Тип данных	Длина	Значение
<b>Заказ</b>			
ID заказа	TINYINT	10	Генерируемый системой порядковый номер, начиная с 1
Дата заказа	DATETIME	18	ГГГГ-ММ-ДД ЧЧ:ММ:СС
Статус заказа	VARCHAR	16	Незавершенный, принят, готов, ожидает доставки, доставлен, отменен
Заказанный товар	INT	10	—

ID клиента	INT	10	—
Email клиента	INT	40	Email должен соответствовать стандарту RFC 5322
Телефон клиента	VARCHAR	11	Номер хранится в международном формате, поддерживаются только номера российских операторов (поэтому цифр 11)
(Местоположение клиента)	VARCHAR	120	—
Имя клиента	VARCHAR	60	—
Сумма заказа	NUMERIC	(18, 0)	Сумма заказа указывается в рублях
Способ оплаты	VARCHAR	16	Наличные, кредитная карта, дебетовая карта
(Комментарий клиента)	VARCHAR	255	—
<b>Заказанный товар</b>			
ID	INT	10	Генерируемый системой порядковый номер, начиная с 1
(ID заказа)	INT	10	—
ID товара	INT	10	—
Количество позиций товара	TINYINT	3	По умолчанию — 1; максимум — 255 (не может превышать количество товаров, которые имеются в запасе)
(данная задача имеет 24 вариации)			

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

#### 11. Методические указания для обучающихся по освоению дисциплины

Целью дисциплины является – получение студентами необходимых знаний, умений и навыков в области проектирования программных систем и использовании соответствующих инструментальных средств.

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

1. Основные определения и понятия;
2. Раскрытие темы лекции;
3. Рассмотрение примеров;
4. Выводы по теме;
5. Уточнение заданий на лабораторные работы (если задания пересекаются с темой лекции).

11.2. Методические указания для обучающихся по выполнению лабораторных работ.

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

## Задание и требования к проведению лабораторных работ

- ЛР 1 — Анализ предметной области. Моделирование функциональных требований к системе. Разработка диаграммы вариантов использования.

На основе представленных пользовательских историй разработать диаграмму вариантов использования (use case diagram), с учётом следующих требований:

1. Диаграмма должна включать все основные функциональные возможности системы с точки зрения заданных действующих лиц (актёров);
2. На диаграмме должны быть представлены четыре вида отношений (ассоциация, обобщение, включение и расширение), а также заданы точки расширения.

- ЛР 2 — Моделирование статического представления системы. Разработка диаграммы классов.

Разработать диаграмму классов (class diagram) с точки зрения моделирования словаря системы или схемы базы данных с учётом следующих требований:

1. На диаграмме классов должны быть представлены минимум три вида отношений, а также в явном виде указаны кратности ассоциаций, уровни доступа к атрибутам и методам классов (public, private, protected);
2. Список атрибутов и методов должен описывать назначение каждого отдельного класса. В противном случае необходимо ввести в диаграмму комментарии для описания роли, которую конкретный класс выполняет в разрабатываемой системе.

- ЛР 3 — Моделирование потоков управления. Разработка диаграммы последовательности.

Разработать не менее 3-х диаграмм последовательности, описывающих взаимодействие объектов для прецедентов из ЛР №1 и осуществить ручное или автоматическое преобразование любой одной диаграммы последовательности в диаграмму коммуникации.

- ЛР 4 — Проектирование пользовательского интерфейса. Составление диаграммы навигации по окнам.

Разработать диаграмму навигации по окнам, используя нотацию и семантику UML-диаграммы видов деятельности.

На диаграмме должны быть представлены основные окна и объекты системы (приложения), включая наиболее значимые пути навигации между ними.

- ЛР 5 — Функциональное моделирование системы. Разработка диаграммы потоков данных.

Разработать структурную модель системы, включая следующие задачи:

1. Построить контекстную диаграмму (DFD 0-го уровня);
2. Осуществить декомпозицию контекстной диаграммы (DFD 1-го уровня);
3. Построить DFD 2-го уровня, выполнив декомпозицию любых 4-х процессов DFD 1-го уровня.

- ЛР 6 — Детализация функциональных требований. Составление спецификации процессов.

Составить спецификацию всех процессов DFD 1-го уровня (из ЛР 5) с учетом подпроцессов при декомпозиции на структурированном естественном языке или псевдокоде.

- ЛР 7 — Моделирование отношений данных. Составление словаря данных.

Построить диаграмму «сущность-связь» (ERD) и определить элементы данных (сущности) через словарь данных с учетом следующих требований:

1. Сущности на ERD и их количество должны соответствовать информации, представленной на DFD в ЛР 1 — то есть ERD должна отражать объекты, хранение которых предусмотрено в системе (атрибуты сущностей на ERD можно не указывать);
2. Информация в словаре данных должна согласовываться с потоками данных спецификации процессов в ЛР 6.

- ЛР 8 — Проектирование архитектуры программной системы

Описать архитектуру системы в соответствии со стандартом ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011.

#### Структура и форма отчета о лабораторной работе

1. Титульный лист;
2. Задание на лабораторную работу и вариант;
3. Результаты выполнения;
4. Выводы по работе;
5. Список использованных источников.
6. Приложение (если предусмотрено заданием)

#### Требования к оформлению отчета о лабораторной работе

1. Отчет о лабораторной работе может быть представлен как на бумажном носителе, так и в электронной форме;
2. Отчет о лабораторной работе должен быть оформлен в соответствии с правилами, приведенными в ГОСТ 7.32-2017.

11.3. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы.

Курсовой проект/ работа проводится с целью формирования у обучающихся опыта комплексного решения конкретных задач профессиональной деятельности.

Курсовой проект/ работа позволяет обучающемуся: изучить способы описания законченного поведения проектируемых программных систем (стандарт ISO/IEC/IEEE 29148:2018).

#### Структура пояснительной записки курсового проекта/ работы

1. Введение (назначение, обозначения и сокращения, границы проекта, ссылки)
2. Общее описание
  - 2.1 Общий взгляд на продукт (контекстная диаграмма)
  - 2.1 Классы и характеристики пользователей
  - 2.1 Операционная среда
  - 2.1 Ограничения дизайна и реализации
3. Системные функции
  - 3.1 Функциональные требования (категория 1)
  - 3.2 Функциональные требования (категория 2)
4. Варианты использования

- 5. Требования к данным
  - 5.1 Логическая модель данных (Entity-Relationship Diagram)
  - 5.2 Словарь данных
  - 5.3 Отчеты
  - 5.4 Целостность, сохранение и утилизация
- 6. Требования к внешним интерфейсам
  - 6.1 Пользовательские интерфейсы
  - 6.2 Навигационная карта пользовательского интерфейса
  - 6.3 Макеты экранных форм
  - 6.4 Коммуникационные интерфейсы
- 7. Атрибуты качества
  - 7.1 Требования к удобству использования
  - 7.2 Требования к производительности
  - 7.3 Требования к безопасности
  - 7.4 Требования к защите
  - 7.5 Требования к доступности
  - 7.6 Требования к надежности
- Заключение
- Список использованных источников

Требования к оформлению пояснительной записки курсового проекта/ работы

Пояснительная записка должна быть оформлена в соответствии с правилами, приведенными в ГОСТ 7.32-2017.

11.4. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине.

11.5. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Тестовый контроль проводится в электронной форме. Вопросы тестового контроля охватывают пройденный лекционный материал и теоретический материал лабораторных работ.

11.6. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой