

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
СТАВРОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ**

**УТВЕРЖДАЮ**

Директор/Декан  
факультета цифровых технологий  
Шлаев Дмитрий Валерьевич

\_\_\_\_\_  
\_\_\_\_\_  
«\_\_» \_\_\_\_\_ 20\_\_ г.

**Рабочая программа дисциплины**

**Б1.В.06 Разработка программных приложений**

09.03.02 Информационные системы и технологии

Инженерия информационных систем

бакалавр

очная

## 1. Цель дисциплины

Основные цели изучения дисциплины:

— Сформировать понимание жизненного цикла ПО и роли требований, архитектуры, интерфейса и процессов поставки в качестве конечного продукта.

— Освоить практики подготовки спецификаций, выбора архитектур, интеграции компонентов и управления совместимостью.

— Научиться организовывать инженерный процесс: контроль версий, стандарты кода, сборку, CI/CD, тестирование, релизы и мониторинг.

— Овладеть принципами проектирования доступных и эффективных интерфейсов и методами их экспертной оценки.

## 2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций ОП ВО и овладение следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.1 Способен создавать программный код, настраивать и конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС	<b>знает</b> как создавать программный код, настраивать и конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС <b>умеет</b> создавать программный код, настраивать и конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС <b>владеет навыками</b> навыками создания программного кода, настройки и конфигурирования ИС в рамках выполнения работ по созданию и сопровождению ИС
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-1.2 Способен развертывать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС	<b>знает</b> как развертывать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС <b>умеет</b> развертывать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС <b>владеет навыками</b> навыками развертывания серверной и клиентской части ИС у заказчика в рамках выполнения работ по созданию и сопровождению ИС
ПК-1 Выполнение работ по созданию (модификации) и сопровождению ИС,	ПК-1.3 Выполняет интеграцию ИС с существующими ИС заказчика в рамках	<b>знает</b> как выполнять интеграцию ИС с существующими ИС заказчика <b>умеет</b>

автоматизирующих задачи организационного управления и бизнес-процессы	выполнения работ по созданию (модификации) и сопровождению ИС	интегрировать ИС с существующими ИС заказчика в рамках выполнения работ по созданию и сопровождению ИС <b>владеет навыками</b> навыками интеграции ИС с существующими ИС заказчика в рамках выполнения работ по созданию и сопровождению ИС
---	---	--

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Разработка программных приложений» является дисциплиной части, формируемой участниками образовательных отношений программы.

Изучение дисциплины осуществляется в бсеместре(-ах).

Для освоения дисциплины «Разработка программных приложений» студенты используют знания, умения и навыки, сформированные в процессе изучения дисциплин:

Интернет-технологии

Конфигурирование в информационных системах

Базы данных Технологии контейнеризации и виртуализации

Интернет-технологии

Конфигурирование в информационных системах

Базы данных Автоматизация тестирования информационных систем

Интернет-технологии

Конфигурирование в информационных системах

Базы данных Конфигурирование в ИС

Интернет-технологии

Конфигурирование в информационных системах

Базы данных Интернет-технологии

Интернет-технологии

Конфигурирование в информационных системах

Базы данных Web-программирование

Освоение дисциплины «Разработка программных приложений» является необходимой основой для последующего изучения следующих дисциплин:

Преддипломная практика

Выполнение и защита выпускной квалификационной работы

Разработка серверных веб-приложений

Тестирование и надёжность информационных систем

Анализ и визуализация данных

Технологии разработки и интеграции

Программная инженерия

### 4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины «Разработка программных приложений» в соответствии с рабочим учебным планом и ее распределение по видам работ представлены ниже.

Семестр	Трудоемкость час/з.е.	Контактная работа с преподавателем, час			Самостоятельная работа, час	Контроль, час	Форма промежуточной аттестации (форма контроля)
		лекции	практические занятия	лабораторные занятия			
6	144/4	18		36	54	36	Эк
в т.ч. часов: в интерактивной форме		4		6			

практической подготовки	18		36	54		
-------------------------	----	--	----	----	--	--

Семестр	Трудоемкость час/з.е.	Внеаудиторная контактная работа с преподавателем, час/чел					
		Курсовая работа	Курсовой проект	Зачет	Дифференцированный зачет	Консультации перед экзаменом	Экзамен
6	144/4						0.25

**5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

№	Наименование раздела/темы	Семестр	Количество часов					Формы текущего контроля успеваемости и промежуточной аттестации	Оценочное средство проверки результатов достижения индикаторов компетенций	Код индикаторов достижения компетенций
			всего	Лекции	Семинарские занятия		Самостоятельная работа			
					Практические	Лабораторные				
1.	1 раздел. 1									
1.1.	ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ	6	14	6		8	16	КТ 1	Тест	ПК-1.1, ПК-1.2, ПК-1.3
1.2.	ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6	12	4		8	14		Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
1.3.	ОРГАНИЗАЦИОННОЕ ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	6	12	4		8	12	КТ 2	Тест	ПК-1.1, ПК-1.2, ПК-1.3
1.4.	ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ	6	16	4		12	12		Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
	Промежуточная аттестация							Эк		
	Итого		144	18		36	54			
	Итого		144	18		36	54			

**5.1. Лекционный курс с указанием видов интерактивной формы проведения занятий**

Тема лекции (и/или наименование раздел) (вид интерактивной формы проведения занятий)/ (практическая подготовка)	Содержание темы (и/или раздела)	Всего, часов / часов интерактивных занятий/ практическая подготовка
ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ	Методы сбора требований к программному обеспечению	4/-

ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ		
ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ	Средства формализации требований к программному обеспечению	2/-
ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Создание технической документации	2/-
ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Архитектурные решения программных приложений	2/2
ОРГАНИЗАЦИОННОЕ И ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	Стандарты кодирования и стиль	2/2
ОРГАНИЗАЦИОННОЕ И ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	Инструменты сопровождения	2/2
ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ	Основы визуального дизайна графического интерфейса	2/2
ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ	Анализ данных о действиях пользователей	2/2
Итого		18

#### 5.2.2. Лабораторные занятия с указанием видов проведения занятий в интерактивной форме

Наименование раздела дисциплины	Формы проведения и темы занятий (вид интерактивной формы проведения занятий)/(практическая подготовка)	Всего, часов / часов интерактивных занятий/ практическая подготовка	
		вид	часы
ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	Функциональные и нефункциональные требования к программному обеспечению	лаб.	4

И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ			
ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ	Проверка и валидация требований к программному обеспечению	лаб.	4
ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Использование шаблонов проектирования программных приложений	лаб.	4
ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Интеграция готовых решений в программные приложения	лаб.	4
ОРГАНИЗАЦИОН НОЕ И ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	Организация процесса кодирования	лаб.	4
ОРГАНИЗАЦИОН НОЕ И ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ	Поддержка и обновление приложений	лаб.	4
ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬС КИЙ ОПЫТ	Реализация интерфейсов	лаб.	6
ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬС КИЙ ОПЫТ	Эвристическая оценка графического интерфейса	лаб.	6

**5.3. Курсовой проект (работа) учебным планом не предусмотрен**



## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Учебно-методическое обеспечение для самостоятельной работы обучающегося по дисциплине «Разработка программных приложений» размещено в электронной информационно-образовательной среде Университета и доступно для обучающегося через его личный кабинет на сайте Университета. Учебно-методическое обеспечение включает:

1. Рабочую программу дисциплины «Разработка программных приложений».
2. Методические рекомендации для организации самостоятельной работы обучающегося по дисциплине «Разработка программных приложений».
3. Методические рекомендации по выполнению письменных работ () (при наличии).
4. Методические рекомендации по выполнению контрольной работы студентами заочной формы обучения (при наличии)
5. Методические указания по выполнению курсовой работы (проекта) (при наличии).

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации:

№ п/п	Темы для самостоятельного изучения	Рекомендуемые источники информации (№ источника)		
		основная (из п.8 РПД)	дополнительная (из п.8 РПД)	метод. лит. (из п.8 РПД)
1	ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
2	ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ И АНАЛИЗ ИСПОЛНЕНИЯ ТРЕБОВАНИЙ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
3	ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
4	ТЕХНИЧЕСКИЕ СПЕЦИФИКАЦИИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
5	ОРГАНИЗАЦИОННОЕ И ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
6	ОРГАНИЗАЦИОННОЕ И	Л1.1	Л2.1	Л3.1, Л3.2

	ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОДИРОВАНИЯ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. Изучение пройденного материала и подготовка к лабораторной работе			
7	ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2
8	ИНТЕРФЕЙС ПРОГРАММНЫХ ПРИЛОЖЕНИЙ И ПОЛЬЗОВАТЕЛЬСКИЙ ОПЫТ. Изучение пройденного материала и подготовка к лабораторной работе	Л1.1	Л2.1	Л3.1, Л3.2

## 7. Фонд оценочных средств (оценочных материалов) для проведения промежуточной аттестации обучающихся по дисциплине «Разработка программных приложений»

### 7.1. Перечень индикаторов компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Индикатор компетенции (код и содержание)	Дисциплины/элементы программы (практики, ГИА), участвующие в формировании индикатора компетенции	

### 7.2. Критерии и шкалы оценивания уровня усвоения индикатора компетенций, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Оценка знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций по дисциплине «Разработка программных приложений» проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль проводится в течение семестра с целью определения уровня усвоения обучающимися знаний, формирования умений и навыков, своевременного выявления преподавателем недостатков в подготовке обучающихся и принятия необходимых мер по её корректировке, а также для совершенствования методики обучения, организации учебной работы и оказания индивидуальной помощи обучающемуся.

Промежуточная аттестация по дисциплине «Разработка программных приложений» проводится в виде Экзамен.

За знания, умения и навыки, приобретенные студентами в период их обучения, выставляются оценки «ЗАЧТЕНО», «НЕ ЗАЧТЕНО». (или «ОТЛИЧНО», «ХОРОШО», «УДОВЛЕТВОРИТЕЛЬНО», «НЕУДОВЛЕТВОРИТЕЛЬНО» для дифференцированного зачета/экзамена)

Для оценивания знаний, умений, навыков и (или) опыта деятельности в университете применяется балльно-рейтинговая система оценки качества освоения образовательной программы. Оценка проводится при проведении текущего контроля успеваемости и промежуточных аттестаций обучающихся. Рейтинговая оценка знаний является интегрированным показателем качества теоретических и практических знаний и навыков студентов по дисциплине.

## Состав балльно-рейтинговой оценки студентов очной формы обучения

Для студентов очной формы обучения знания по осваиваемым компетенциям формируются на лекционных и практических занятиях, а также в процессе самостоятельной подготовки.

В соответствии с балльно-рейтинговой системой оценки, принятой в Университете студентам начисляются баллы по следующим видам работ:

№ контрольной точки	Оценочное средство результатов индикаторов достижения компетенций	Максимальное количество баллов
6 семестр		
КТ 1	Тест	15
КТ 2	Тест	15
<b>Сумма баллов по итогам текущего контроля</b>		<b>30</b>
Посещение лекционных занятий		20
Посещение практических/лабораторных занятий		20
Результативность работы на практических/лабораторных занятиях		30
Итого		100

№ контрольной точки	Оценочное средство результатов индикаторов достижений компетенций	Максимальное количество баллов	Критерии оценки знаний студентов
6 семестр			
КТ 1	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.
КТ 2	Тест	15	11-15 баллов выставляется обучающемуся, если тестовые задания выполняются на 85% и выше; 8-10 баллов выставляется обучающемуся, если тестовые задания выполняются на 70 - 84%; 5-7 баллов выставляется обучающемуся, если тестовые задания выполняются на 55 – 69 %; 1-4 балла выставляется обучающемуся, если тестовые задания выполняются на 45 – 54%; 0 баллов выставляется обучающемуся, если тестовые задания выполняются на 44% и меньше.

## Критерии и шкалы оценивания результатов обучения на промежуточной аттестации

При проведении итоговой аттестации «зачет» («дифференцированный зачет», «экзамен») преподавателю с согласия студента разрешается выставлять оценки («отлично», «хорошо», «удовлетворительно», «зачет») по результатам набранных баллов в ходе текущего контроля успеваемости в семестре по выше приведенной шкале.

В случае отказа – студент сдает зачет (дифференцированный зачет, экзамен) по приведенным выше вопросам и заданиям. Итоговая успеваемость (зачет, дифференцированный зачет, экзамен) не может оцениваться ниже суммы баллов, которую студент набрал по итогам текущей и промежуточной успеваемости.

При сдаче (зачета, дифференцированного зачета, экзамена) к заработанным в течение семестра студентом баллам прибавляются баллы, полученные на (зачете, дифференцированном зачете, экзамене) и сумма баллов переводится в оценку.

## Критерии и шкалы оценивания ответа на экзамене

Сдача экзамена может добавить к текущей балльно-рейтинговой оценке студентов не более 20 баллов:

Содержание билета	Количество баллов
Теоретический вопрос №1	до 7
Теоретический вопрос №2	до 7
Задача (оценка умений и	до 6
Итого	20

## Критерии оценки ответа на экзамене

Теоретические вопросы (вопрос 1, вопрос 2)

7 баллов выставляется студенту, полностью освоившему материал дисциплины или курса в соответствии с учебной программой, включая вопросы рассматриваемые в рекомендованной программой дополнительной справочно-нормативной и научно-технической литературы, свободно владеющему основными понятиями дисциплины. Требуется полное понимание и четкость изложения ответов по экзаменационному заданию (билету) и дополнительным вопросам, заданных экзаменатором. Дополнительные вопросы, как правило, должны относиться к материалу дисциплины или курса, не отраженному в основном экзаменационном задании (билете) и выявляют полноту знаний студента по дисциплине.

5 балла заслуживает студент, ответивший полностью и без ошибок на вопросы экзаменационного задания и показавший знания основных понятий дисциплины в соответствии с обязательной программой курса и рекомендованной основной литературой.

3 балла дан недостаточно полный и недостаточно развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Речевое оформление требует поправок, коррекции.

2 балла дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

1 балл дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствуют фрагментарность, нелогичность изложения. Студент не осознает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

0 баллов - при полном отсутствии ответа, имеющего отношение к вопросу.

#### Оценивание задачи

6 баллов Задачи решены в полном объеме с соблюдением необходимой последовательности.

5 баллов Задачи решены с небольшими недочетами.

4 балла Задачи решены с небольшими недочетами.

3 балла Задачи решены не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы.

2 балла Задачи решены не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы.

1 баллов Задачи решены частично, с большим количеством вычислительных ошибок, объем выполненной части работы не позволяет сделать правильных выводов.

0 баллов Задачи не решены или работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов.

Перевод рейтинговых баллов в пятибалльную систему оценки знаний обучающихся:  
для экзамена:

- «отлично» – от 89 до 100 баллов – теоретическое содержание курса освоено полностью, без пробелов необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному;

- «хорошо» – от 77 до 88 баллов – теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками;

- «удовлетворительно» – от 65 до 76 баллов – теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки;

- «неудовлетворительно» – от 0 до 64 баллов - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к существенному повышению качества выполнения учебных заданий

### **7.3. Примерные оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Разработка программных приложений»**

#### Вопросы для экзамена

1. Технология программирования и основные этапы ее развития
2. Проблемы разработки сложных программных приложений
3. Блочный-иерархический подход к созданию сложных систем
4. Жизненный цикл
5. Ускорение разработки программного обеспечения. Технология RAD
6. Понятие технологичности программного обеспечения
7. Нисходящая и восходящая разработка программного обеспечения
8. Эффективность и технологичность
9. Программирование «С защитой от ошибок»
10. Сквозной структурный контроль
11. Классификация программных продуктов по функциональному назначению
12. Основные эксплуатационные требования к программным продуктам
13. Предпроектные исследования предметной области
14. Разработка технического задания
15. Принципиальные решения начальных этапов проектирования
16. Спецификации программного обеспечения при структурном подходе
17. Диаграммы переходов состояний
18. Функциональные диаграммы

19. Диаграммы потоков данных
  20. Модели
  21. Разработка структурной и функциональной схем
  22. Исследование метода пошаговой детализации для проектирования структуры программного обеспечения
  23. Проектирование структур данных
  24. Проектирование программного обеспечения, основанное на декомпозиции данных
  25. Case-технологии
  26. UML – стандартный язык описания разработки программных продуктов с использованием объектного подхода
  27. Определение «вариантов использования»
  28. Построение концептуальной модели предметной области
  29. Описание поведения. Системные события и операции
  30. Разработка структуры программного обеспечения при объектном подходе
  31. Определение отношений между объектами
  32. Проектирование классов
  33. Компоновка программных компонентов
  34. Типы пользовательских интерфейсов и этапы их разработки
  35. Психологические особенности человека, учитываемые при создании пользовательского интерфейса
  36. Модели пользовательского интерфейса
  37. Виды контроля качества
  38. Структурное тестирование
  39. Функциональное тестирование
  40. Комплексное тестирование
  41. Оценочное тестирование
  42. Классификация ошибок
  43. Методы отладки
  44. Виды программных документов
  45. Пояснительная записка
  46. Руководство пользователя
  47. Руководство системного программиста
- Примерные темы курсовых работ
1. Разработка приложения для торгового предприятия
  2. Разработка приложения для кредитного отдела банка
  3. Разработка приложения для гостиницы
  4. Разработка приложения для авторемонтной мастерской
  5. Разработка приложения для автосалона
  6. Разработка приложения для агентства недвижимости
  7. Разработка приложения для склада
  8. Разработка приложения для учета расходов семьи
  9. Разработка приложения для организации делопроизводства
  10. Разработка приложения для рекламного агентства
  11. Разработка приложения для службы поддержки
  12. Разработка приложения для кадровой службы организации
  13. Разработка приложения для туристического агентства
  14. Разработка приложения компьютерного магазина
  15. Разработка приложения для страховой компании
  16. Разработка приложения для кафедры вуза
  17. Разработка приложения для мебельного магазина
  18. Разработка приложения для книжного магазина
  19. Разработка приложения для учета договоров в организации
  20. Разработка приложения для строительной компании
  21. Разработка приложения по взаимодействию с клиентами организации
  22. Разработка приложения для взаимодействия с заказчиками в организации
  23. Разработка приложения для салона красоты

24. Разработка приложения сервисного центра по ремонту компьютерной техники
25. Разработка приложения для транспортной компании
26. Разработка приложения для менеджера по продажам
27. Разработка приложения для учета компьютерной техники и программного обеспечения в организации
28. Разработка приложения для организации документооборота
29. Разработка приложения для управляющей компании ЖКХ
30. Разработка приложения для салона сотовой связи

ПК-1.1 Способен создавать программный код, настраивать и конфигурировать ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС

#### ЗНАНИЯ

1. Какое ключевое преимущество клиент–серверной архитектуры подчёркивается в учебнике?

- A) Отсутствие задержек сети
- B) Централизация бизнес-логики и данных на сервере
- C) Полная независимость клиента от сервера
- D) Ненужность масштабирования

Ответ: B.

2. Что включает этап сопровождения ПО?

- A) Только добавление новых функций
- B) Только исправление ошибок
- C) Исправление ошибок, добавление функций, оптимизация и адаптация
- D) Только оптимизацию производительности

Ответ: C.

3. Выберите ДВЕ верные выгоды от использования систем управления версиями (VCS):

- A) Возможность отката к предыдущим состояниям проекта
- B) Автоматическое исправление конфликтов без участия разработчика
- C) Параллельная работа команды с объединением изменений после проверки
- D) Запрет на интеграцию с конвейерами CI/CD

Ответ: A, C.

4. Какие ДВА понятия относятся к базовым концепциям Git?

- A) Ветки (branches)
- B) Оптимизатор SQL-запросов
- C) Пулл-реквесты (Pull Requests)
- D) Сборщик мусора JVM

Ответ: A, C.

5. Что характерно для Waterfall (каскадной) модели? Выберите ДВА варианта.

- A) Проектирование предшествует реализации, ошибки на этом этапе критичны
- B) Лёгкая адаптация архитектуры в ходе разработки
- C) Разработка стартует до завершения проектирования
- D) Возврат на прошлые этапы возможен, но дорог

Ответ: A, D.

#### УМЕНИЯ

6. Расположите шаги типового рабочего процесса с Git для проекта на Rust:

- A) Создание новой ветки
- B) Внесение изменений и коммит
- C) Отправка изменений в удалённый репозиторий (push)
- D) Клонирование удалённого репозитория

Правильный порядок: D → A → B → C

7. Соотнесите термин (1–4) и определение (A–D):

1. Функциональные требования
2. Нефункциональные требования
3. «Хлебные крошки»
4. Эвристическая оценка

A) Навигационный паттерн, показывающий путь и уровень вложенности

В) Описывают, что должна делать система

С) Быстрый экспертный аудит интерфейса по набору правил

Д) Определяют требуемые качества: производительность, надёжность, безопасность, UX

Ответ: 1–В, 2–D, 3–А, 4–С.

8. Верно ли, что результатом выполнения программы на RUST будет 2?

```
fn main() {  
    let s = String::from("hi");  
    let len = s.len();  
    println!("{}", len);  
}
```

Ответ: Верно

9. Верно ли, что результатом выполнения программы на RUST будет 6?

```
fn main() {  
    let mut x = 5;  
    x = x + 1;  
    println!("{}", x);  
}
```

Ответ: Верно

10. Верно ли, что результатом выполнения программы на RUST будет 4?

```
fn main() {  
    let s = String::from("ok");  
    let r1 = &s;  
    let r2 = &s;  
    println!("{}", r1.len() + r2.len());  
}
```

Ответ: Верно

## НАВЫКИ

11. В ответе впишите целое число, которое будет получено в результате выполнения программы:

```
fn fact(n: u64) -> u64 {  
    (1..=n).product()  
}  
fn main() {  
    println!("{}", fact(5));  
}
```

Ответ: 120

12. В ответе впишите целое число, которое будет получено в результате выполнения программы:

```
fn main() {  
    let s = "abracadabra";  
    let vowels = ['a','e','i','o','u'];  
    let count = s.chars().filter(|c| vowels.contains(c)).count();  
    println!("{}", count);  
}
```

Ответ: 5

13. В ответе впишите целое число, которое будет получено в результате выполнения программы:

```
fn main() {  
    let sum: i32 = (1..=10).filter(|n| n % 3 == 0).sum();  
    println!("{}", sum);  
}
```

Ответ: 18

14. Верно ли, что результатом выполнения программы на RUST будет 1?

```
fn main() {  
    let s = String::from("a");  
    let t = s;  
    println!("{}", s.len());  
}
```

Ответ: Неверно (ошибка после перемещения владения)

15. Верно ли, что результатом выполнения программы на RUST будет 3?

```
fn main() {  
    let a = [1, 2, 3];  
    println!("{}", a[3]);  
}
```

Ответ: Неверно (паника из-за выхода за пределы массива)

ПК-1.2 Способен развертывать и администрировать серверную и клиентскую часть ИС у заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС

#### ЗНАНИЯ

1. Что является ключевым преимуществом клиент-серверной архитектуры при развёртывании и администрировании?

- A) Полная независимость клиента от сервера
- B) Централизация бизнес-логики и данных на сервере
- C) Отсутствие задержек сети
- D) Необходимость ручного обновления клиентов

Ответ: B.

2. Какая практика прямо предотвращает SQL-инъекции при работе серверной части с БД в Rust (diesel)?

- A) Подготовленные (параметризованные) запросы
- B) Конкатенация SQL-строк
- C) Отключение внешних ключей
- D) Логирование паролей для отладки

Ответ: A.

3. Выберите ДВА шага, которые учебник относит к типичному CI/CD-конвейеру для проектов на Rust:

- A) cargo fmt/cargo clippy как обязательные проверки
- B) cargo audit/cargo deny для контроля уязвимостей/лицензий
- C) Ручная правка README перед каждым деплоем
- D) Обязательная «ручная» миграция ОС на сервере перед сборкой

Ответ: A, B.

4. Какие ДВА утверждения отражают рекомендуемые практики наблюдаемости/логирования при эксплуатации ИС?

- A) Структурированные логи без секретов
- B) Корреляционные идентификаторы запросов сквозь все слои
- C) Хранение паролей в логах для быстрого доступа
- D) Отключение метрик в продакшене

Ответ: A, B.

5. Что характерно для микросервисной архитектуры в эксплуатации? (выберите ДВА)

- A) Требуется инструментов оркестрации (например, Kubernetes) и мониторинга
- B) Все сервисы деплоятся только единым артефактом
- C) Отказ одного сервиса валит всю систему
- D) Сервисы разворачиваются и масштабируются независимо

Ответ: A, D

#### УМЕНИЯ

6. Расположите шаги типового конвейера поставки сервиса на Rust:

- A) Сборка и публикация артефактов/образов
- B) Линт и сборка (fmt, clippy, build)
- C) Развёртывание в целевое окружение
- D) Тестирование (cargo test, доктесты)

Правильный порядок: B → D → A → C.

7. Соотнесите термин (1–4) и определение (A–D):

- 1. Контейнеризация
  - 2. Оркестрация
  - 3. Наблюдаемость
  - 4. План отката (rollback)
- A) Управление множеством контейнеров/сервисов в продакшене
  - B) Упаковка приложения и зависимостей в образ
  - C) Заранее подготовленная процедура возврата версии
  - D) Видимость состояния через логи, метрики и трассировки
- Ответ: 1–B, 2–A, 3–D, 4–C.

8. Верно ли, что если PORT не задана, программа выведет 8080?

```
use std::env;
fn main() {
    let port = env::var("PORT").unwrap_or("8080".to_string());
    println!("{}", port);
}
```

Ответ: Верно

9. Верно ли, что программа выведет true?

```
use std::net::IpAddr;
fn main() {
    let ip: IpAddr = "0.0.0.0".parse().unwrap();
    println!("{}", ip.is_ipv4());
}
```

Ответ: Верно

10. Верно ли, что программа выведет 2?

```
fn main() {
    let levels = vec!["INFO", "ERROR", "DEBUG", "ERROR"];
    let count = levels.iter().filter(|&s| s == "ERROR").count();
    println!("{}", count);
}
```

Ответ: Верно

## НАВЫКИ

11. Впишите число, которое напечатает программа:

```
fn main() {
    let cfg = "HOST=0.0.0.0;PORT=9090";
    let port: u16 = cfg
        .split(';')
        .find(|s| s.starts_with("PORT="))
        .and_then(|s| s[5..].parse().ok())
        .unwrap();
    println!("{}", port);
}
```

Ответ: 9090

12. Впишите число, которое напечатает программа:

```
fn main() {
  let s = "error,info,info,debug,error,trace";
  let out = s.split(',').filter(|&x| x == "error").count();
  println!("{}", out);
}
```

Ответ: 2

13. Впишите слово, которое напечатает программа:

```
fn main() {
  let url = "https://example.org/api/v1/health";
  let last = url.rsplit('/').next().unwrap();
  println!("{}", last);
}
```

Ответ: health

14. Верно ли, что программа выведет true?

```
fn main() {
  let line = "user=admin;password=secret";
  let has_secret = line.contains("password=");
  println!("{}", has_secret);
}
```

Ответ: Верно

15. Верно ли, что программа успешно скомпилируется и выведет error?

```
fn main() {
  let levels = ["info", "warn", "error"];
  let first = levels[3];
  println!("{}", first);
}
```

Ответ: Неверно (паника из-за выхода за границы массива во время выполнения)

ПК-1.3 Способен интегрировать ИС с существующими ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС

## ЗНАНИЯ

1. Что является базовым требованием безопасности при интеграции с внешним API?

- A) Отключить TLS ради скорости
- B) Использовать HTTPS и механизм аутентификации (ключ/токен)
- C) Передавать секреты в URL
- D) Хранить токены в логах

Ответ: B.

2. Что помогает избежать «ломающих» изменений при эволюции внешнего сервиса?

- A) Игнорирование изменений в ответах API
- B) Жёсткое кодирование структуры ответа
- C) Явное версионирование API и отслеживание обновлений
- D) Полный отказ от внешних API

Ответ: C.

3. Выберите ДВА подхода для повышения надёжности интеграции с внешним API:

- A) Повторные попытки с backoff
- B) Кэширование ответов
- C) Передача паролей в открытом виде
- D) Жёсткое падение при первом сетевом сбое

Ответ: A, B.

4. Какие два фактора часто вызывают несовместимость при интеграции?

- A) Разные форматы/версии API (JSON, XML, Protobuf)
- B) Единый формат по умолчанию
- C) Стабильные неизменяемые схемы

D) Изменение структуры ответов внешнего сервиса

Ответ: A, D

5. Какие ДВА преимущества событийной интеграции через брокер сообщений?

A) Слабая связность компонентов

B) Обязательная синхронность вызовов

C) Буферизация событий в очереди при временной недоступности потребителя

D) Жёсткая зависимость от порядка вызовов RPC

Ответ: A, C

## УМЕНИЯ

6. Расположите шаги типовой интеграции с внешним API:

A) Реализовать повторные попытки при сбоях

B) Настроить безопасность (HTTPS, токен/ключ)

C) Явно указать используемую версию API

D) Добавить кэширование ответов

Правильный порядок: B → C → A → D

7. Соотнесите термин (1–4) и определение (A–D):

1. API-токен/ключ

2. Версионирование API

3. Кэширование ответов

4. Мок-сервер для интеграционных тестов

A) Явное указание версии интерфейса для контролируемых переходов

B) Симуляция внешнего сервиса для тестов интеграции

C) Секрет для аутентификации запросов

D) Снижение числа запросов к внешнему API путём хранения результатов

Ответ: 1–C, 2–A, 3–D, 4–B

8. Верно ли, что после компиляции программа выведет `https://api.example.com/v2/users?`

```
fn main() {  
  let base = "https://api.example.com";  
  let v = "v2";  
  let path = "users";  
  let url = format!("{}/{}/", base, v, path);  
  println!("{}", url);  
}
```

Ответ: Верно

9. Верно ли, что после компиляции программа выведет `true`?

```
fn main() {  
  let mut headers = String::from("Authorization: Bearer ");  
  headers.push_str("ABC123");  
  println!("{}", headers.contains("Bearer ABC123"));  
}
```

Ответ: Верно

10. Верно ли, что после компиляции программа выведет `xyz`?

```
fn main() {  
  let qs = "a=1&b=2&token=xyz";  
  let token = qs.split('&')  
    .find(|p| p.starts_with("token="))  
    .unwrap();  
  println!("{}", &token[6..]);  
}
```

Ответ: Верно

## НАВЫКИ

11. В ответе впишите совокупность текстовых символов и чисел, которые напечатает программа:

```
fn main() {
  let h = "Authorization: Bearer TOK123";
  let tok = h.split_whitespace().last().unwrap();
  println!("{}", tok);
}
```

Ответ: TOK123

12. В ответе впишите число, которое напечатает программа:

```
fn main() {
  let qs = "a=1&b=2&b=3";
  let count = qs.split('&').filter(|p| p.starts_with("b=")).count();
  println!("{}", count);
}
```

Ответ: 2

13. В ответе впишите число, которое напечатает программа:

```
fn main() {
  let base = "https://srv.local";
  let ver = "v1";
  let ep = "health";
  let url = format!("{}/{}{}", base, ver, ep);
  println!("{}", url.len());
}
```

Ответ: 27

14. Верно ли, что программа успешно скомпилируется и выведет XYZ?

```
fn main() {
  let token = String::from("XYZ");
  let header = token;
  println!("{}", token);
}
```

Ответ: Неверно (владение token перемещено в header, повторное использование недопустимо)

15. Верно ли, что программа успешно скомпилируется и выведет true?

```
fn main() {
  let payload = r#"{"ok":true}"#;
  let ok = payload.contains("\"ok\":true");
  println!("{}", ok);
}
```

Ответ: Верно

## 8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

### основная

Л1.1 Болотнов А. М. Разработка программных приложений в среде BlackBox [Электронный ресурс]: учеб. пособие ; ВО - Бакалавриат. - Санкт-Петербург: Лань, 2022. - 144 с. – Режим доступа: <https://e.lanbook.com/book/213140>

### дополнительная

Л2.1 сост. Н. И. Битюцкая Разработка программных приложений [Электронный ресурс]:лаборатор. практикум; ВО - Бакалавриат, Магистратура. - Ставрополь: СКФУ, 2015. - 140 с. – Режим доступа: <https://e.lanbook.com/book/155249>

б) Методические материалы, разработанные преподавателями кафедры по дисциплине, в соответствии с профилем ОП.

Л3.1 Букунов С. В., Букунова О. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат, Магистратура, Аспирантура. - Санкт-Петербург: Лань, 2023. - 88 с. – Режим доступа: <https://e.lanbook.com/book/292856>

Л3.2 Кузнецов И. В., Исаев М. С., Пономарчук Ю. В., Холодилов А. А. Разработка приложений для ОС Android [Электронный ресурс]:учеб. пособие; ВО - Бакалавриат, Магистратура. - Хабаровск: ДВГУПС, 2023. - 112 с. – Режим доступа: <https://e.lanbook.com/book/433625>

## 9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Наименование ресурса сети «Интернет»	Электронный адрес ресурса
1	Заяц А. М., Васильев Н. П. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js	<a href="https://lanbook.com/catalog/informatika/proektirovanie-i-razrabotka-web-prilozheniy-vvedenie-v-frontend-i-backend-razrabotku-na-javascript-i/">https://lanbook.com/catalog/informatika/proektirovanie-i-razrabotka-web-prilozheniy-vvedenie-v-frontend-i-backend-razrabotku-na-javascript-i/</a>

## 10. Методические указания для обучающихся по освоению дисциплины

При изучении дисциплины используются следующие виды учебных занятий:

1. Лекции, на которых рассматриваются основные теоретические вопросы данного междисциплинарного курса. Материалы лекций можно изучить, обратившись к списку основной и дополнительной литературы. Посещаемость лекций входит в балльную оценку по междисциплинарному курсу и контролируется преподавателем.

В ходе лекционных занятий обязательно конспектирование учебного материала. При этом стоит обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой - в ходе подготовки к семинарам изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т.д.

Конспекты лекций контролируется преподавателем

2. Лабораторные работы предполагают выполнение предложенных заданий письменно или в электронном виде, в зависимости от типа задания.

В каждом задании указывается форма его выполнения и способ предоставления на оценку.

При подготовке лабораторным занятиям следует:

- использовать рекомендованные преподавателями учебники и учебные пособия
- для закрепления теоретического материала;
- изучить лекционный материал по данной теме;
- разобрать, совместно с другими студентами, обсудить вопросы по теме занятия;
- уточнить особенности оформления заданий и предоставления их на оценку, если представленных на образовательном портале комментариев недостаточно.

Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры,

поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам.

В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Идя на консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения.

Лабораторные работы также входят в балльную оценку по междисциплинарному курсу и контролируются преподавателем.

3. Самостоятельная работа студента является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа студента над усвоением учебного материала может выполняться в библиотеке, учебных кабинетах, компьютерных классах, а также в домашних условиях.

Учебный материал самостоятельной работы выносится на итоговый контроль наряду с учебным материалом, который разрабатывался при проведении учебных занятий. Содержание самостоятельной работы студента определяется учебной программой, методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа студентов осуществляется в аудиторной и внеаудиторной формах.

Самостоятельная работа студентов в аудиторное время может включать:

- конспектирование (составление тезисов) лекций;
- выполнение контрольных работ, коллоквиума;
- решение задач;
- работу со справочной и методической литературой;
- защиту выполненных работ;
- участие в тестировании и др.

Самостоятельная работа студентов во внеаудиторное время может состоять из:

- повторение лекционного материала;
- подготовки к лабораторным занятиям;
- изучения учебной и научной литературы;
- решения задач, выданных на практических занятиях;
- подготовки к контрольным работам, тестированию и т.д.;
- подготовки рефератов, эссе и иных индивидуальных письменных работ по заданию преподавателя.

Самостоятельная работа студентов также входит в балльную оценку по междисциплинарному курсу и контролируется преподавателем

Рекомендации по подготовке к зачету

Формой итогового контроля знаний студентов по дисциплине является зачет с оценкой.

Зачет с оценкой, на который явка обязательна, проводится согласно расписанию учебных занятий. Зачет с оценкой является формой отчетности, фиксирующей, что студент выполнил необходимый минимум работы по освоению определенного раздела образовательной программы.

Подготовка к экзамену и успешное освоение материала дисциплины начинается с первого дня изучения дисциплины и требует от студента систематической работы:

- 1) не пропускать аудиторные занятия (лекции, лабораторные занятия);
- 2) активно участвовать в работе семинаров (выступать с сообщениями, проявляя себя в выполнении всех видов заданий – устном опросе, творческих заданиях, в решении и обсуждении контекстных задач, в деловой игре, выполнять все требования преподавателя по изучению курса, приходить подготовленными к занятию).

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства и информационных справочных систем (при необходимости).**

### *11.1 Перечень лицензионного программного обеспечения*

1. Kaspersky Endpoint Security 12.11 - Антивирус

2. Microsoft Windows Server STDCORE AllLngLicense/Software AssurancePack Academic OLV 16Licenses LevelE AdditionalProduct CoreLic 1Year - Серверная операционная система
3. OPERA - Система управления отелем

### 11.3 Перечень программного обеспечения отечественного производства

1. Kaspersky Endpoint Security 12.11 - Антивирус

При осуществлении образовательного процесса студентами и преподавателем используются следующие информационно справочные системы: СПС «Консультант плюс», СПС «Гарант».

## 12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование специальных помещений и помещений для самостоятельной работы	Номер аудитории	Оснащенность специальных помещений и помещений для самостоятельной работы
1	Учебная аудитория для проведения занятий всех типов (в т.ч. лекционного, семинарского, практической подготовки обучающихся), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации		
		Э-184	Оснащение: специализированная мебель на 25 посадочных мест, персональные компьютеры – 25 шт., мультимедийный проектор Epson EB-965H– 1 шт., интерактивная доска SMART Board 690 – 1 шт., сервер Hp, учебно-наглядные пособия в виде презентаций, информационные плакаты, подключение к сети «Интернет», доступ в электронную информационно-образовательную среду университета, выход в корпоративную сеть университета.
2	Помещение для самостоятельной работы обучающихся, подтверждающее наличие материально-технического обеспечения, с перечнем основного оборудования		

### 13. Особенности реализации дисциплины лиц с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- задания для выполнения, а также инструкция о порядке проведения промежуточной аттестации оформляются увеличенным шрифтом;

- задания для выполнения на промежуточной аттестации зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- промежуточная аттестация проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по желанию студента промежуточная аттестация может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента промежуточная аттестация проводится в устной форме.

Рабочая программа дисциплины «Разработка программных приложений» составлена на основе Федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии (приказ Минобрнауки России от 19.09.2017 г. № 926).

Автор (ы)

\_\_\_\_\_ доц. , кэн Ермакова А.Н.

Рецензенты

\_\_\_\_\_ проф. , дэн Шуваев А.В.

\_\_\_\_\_ доц. КИС, кпн Богданова Светлана Викторовна

Рабочая программа дисциплины «Разработка программных приложений» рассмотрена на заседании Кафедра информационных систем протокол № 9 от 07.04.2026 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии

Заведующий кафедрой \_\_\_\_\_ Березницкий А.С.

Рабочая программа дисциплины «Разработка программных приложений» рассмотрена на заседании учебно-методической комиссии Факультет цифровых технологий протокол № 2 от 08.04.2026 г. и признана соответствующей требованиям ФГОС ВО и учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии

Руководитель ОП \_\_\_\_\_