МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО Руководитель ОПОП	УТВЕРЖДАЮ Зав. кафедрой ПМИ
_____М.В. Коломина	_____М.В. Коломина
«5» апреля 2024 г.	«5» апреля 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Составители	**Корнеев Г.А., к.т.н., доцент ФИТиП, ИТМО** **Ульянцев В.И., к.т.н, доцент ФИТиП, ИТМО** **Нариманян Э.В., асситент каф. ПМИ, АГУ**
Согласовано с работодателями	**Белов С.В., директор ООО «ТРАСТ ПОИНТ»** **Измайлов Г.А., генеральный директор ООО «Агент Плюс»**
Направление подготовки / специальность	**01.03.02 Прикладная математика и информатика**
Направленность (профиль) ОПОП	**Программирование и искусственный интеллект**
Квалификация (степень)	**бакалавр**
Форма обучения	**очная**
Год приёма	**2024**
Курс	**4**
Семестр(ы)	**7, 8**

Астрахань – 2024

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целями освоения дисциплины «Проектирование программного обеспечения» является формирование у студентов представлений о современных процессах проектирования, обеспечивающих разработку качественного программного обеспечения, удовлетворяющего предъявляемым требованиям.

1.2. Задачи освоения дисциплины:

- изучение стилей программирования, классических паттернов проектирования, гибких технологий;
- формирование навыков кодирования, управления исходным кодом, планирования и управления проектом.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина «Проектирование программного обеспечения» относится к обязательной части и осваивается в 7,8 семестрах.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения, навыки, формируемые предшествующими учебными дисциплинами (модулями):

Технологии программирования,

Базы данных,

Основы проектной деятельности,

Параллельное программирование.

Знания: принципов объектно-ориентированного программирования, структур данных, принципы разработки ПО, архитектуры ПО, баз данных

Умения: решать типовые задачи по предшествующим дисциплинам, тестировать программное обеспечение, собирать и анализировать требования к программному обеспечению.

Навыки: создания программных решений, анализировать сложные проблемы проектирования и находить оптимальные решения, быстро адаптироваться к изменениям в требованиях или технологиях в процессе разработки.

2.3. Последующие учебные дисциплины и (или) практики, для которых необходимы знания, умения, навыки, формируемые данной учебной дисциплиной:

- Написание выпускной квалификационной работы,
- Производственная практика.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс освоения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки:

а) общепрофессиональных (ОПК)

- ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
- ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

б) профессиональных (ПК)

- ПК-2. Способен осуществлять интеграцию программных модулей и компонент и проверку работоспособности кода программного обеспечения.
- ПК-6. Способен выполнять работы по созданию и сопровождению информационных систем.
- ПК-7. Способен создавать и оценивать варианты архитектуры программного средства и осуществлять выбор среди них.

- ПК-9. Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.
- ПК-10. Разработка и реализация архитектуры программного обеспечения.
- ПК-11. Разработка и сопровождение программных проектов

Таблица 1. Декомпозиция результатов обучения

Код компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине (модулю)		
		Знать (1)	Уметь (2)	Владеть (3)
ОПК-4	ОПК-4.1. Использует современные информационные технологии и программные средства для решения профессиональных задач ОПК-4.2. Использует принципы информационной безопасности при работе с информацией в процессе решения задач профессиональной деятельности	Современные информационные-коммуникационные технологии необходимые для решения задач профессиональной деятельности, основные требования информационной безопасности.	Решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.	Навыками применения существующих информационно-коммуникационных технологий для решения задач в области профессиональной деятельности с учетом требований информационной безопасности.
ОПК-5	ОПК-5.1. Знает методы и алгоритмы разработки компьютерных программ для решения прикладных задач. ОПК-5.2. Умеет разрабатывать алгоритмы и компьютерные программы, реализовывать алгоритмические и программные решения в прикладных областях. ОПК-5.3. Имеет практический опыт разработки и реализации алгоритмических и программных решений, пригодных для практического применения.	Сложность реализации шаблонов проектирования по.	Оценивать необходимость рефакторинга и преимущества его проведения в существующем программном продукте.	Навыками работы с системами контроля версий.
ПК-2	ПК-2.1. Способен разрабатывать тестовые наборы данных. ПК-2.2. Способен проверять работоспособность программного обеспечения ПК-2.3. Способен осуществлять интеграцию программных модулей и компонентов и верификацию выпусков программного продукта	Программные продукты, программные модули и компоненты и верификацию выпусков программного продукта	Проверять работоспособность программного обеспечения	Осуществлять интеграции программных модулей
ПК-6	ПК-6.1. Способен осуществлять кодирование на языках программирования. ПК-6.2. Способен осуществлять установку и настройку системного и прикладного ПО, необходимого для функционирования ИС ПК-6.3. Способен осуществлять настройку оборудования, необходимого для работы ИС	Языки программирования	Кодировать на языках программирования, осуществлять установку и настройку системного и прикладного по	Навыками создания и сопровождения информационных систем

ПК-7	ПК-7.1. Способен определять перечни возможных типов, слоев и шаблонов проектирования программных компонентов ПК-7.2. Способен определять качественные характеристик и осуществлять выбор типа и слоев программных компонентов	Типы, слои и шаблоны проектирования программных компонентов, архитектуры программного средства	Определять качественные характеристики и осуществлять выбор типа и слоев программных компонентов	Навыками создания и оценивания вариантов архитектуры программного средства
ПК-9	ПК-9.1. Владение базовыми навыками теории графов и алгоритмами на них ПК-9.2. Владение основами теории вычислимости и оценки сложности алгоритмов	Базовые навыки теории графов и алгоритмами на них, основ теории вычислимости и оценки сложности алгоритмов	Владеть основами теории вычислимости и оценки сложности алгоритмов	Навыками разработки и применения алгоритмических и программных решений в области системного и прикладного программного обеспечения
ПК-10	ПК-10.1. Описание алгоритмов компонентов, включая методы и схемы ПК-10.2. Описание технологии обработки данных для возможности их использования в программном средстве, включая вопросы параллельной обработки	Алгоритмы компоненты, включая методы и схемы	Описать технологии обработки данных для возможности их использования в программном средстве	Навыками разработки и реализации архитектуры программного обеспечения
ПК-11	ПК-11.1. Формализация и алгоритмизация поставленных задач ПК-11.2. Проектирование программного обеспечения	Формализацию и алгоритмизацию поставленных задач	Проектировать программное обеспечение	Навыками разработки и сопровождения программных проектов

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины в соответствии с учебным планом составляет 8 зачетных единиц (288 часов).

Трудоемкость отдельных видов учебной работы студентов очной формы обучения приведена в таблице 2.1.

Таблица 2.1. Трудоемкость отдельных видов учебной работы по формам обучения

Вид учебной и внеучебной работы	для очной формы обучения
Объем дисциплины в зачетных единицах	8
Объем дисциплины в академических часах	288
Контактная работа обучающихся с преподавателем (всего), в том числе (час.):	131
- занятия лекционного типа, в том числе:	52
- практическая подготовка (если предусмотрена)	0
- занятия семинарского типа (семинары, практические, лабораторные), в том числе:	78
- практическая подготовка (если предусмотрена)	0
- в ходе подготовки и защиты курсовой работы	0
- консультация (предэкзаменационная)	1
- промежуточная аттестация по дисциплине	0
Самостоятельная работа обучающихся (час.)	157
Форма промежуточной аттестации обучающегося (зачет/экзамен), семестр (ы)	Диф. Зачет – 7 семестр Экзамен – 8 семестр

Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий и самостоятельной работы представлены в таблице 2.2.

Таблица 2.2. Структура и содержание дисциплины (модуля)

для очной формы обучения

Раздел, тема дисциплины (модуля)	Контактная работа, час.							СР, час.	Итого часов	Форма текущего контроля успеваемости, форма промежуточной аттестации *[по семестрам]*
	Л		ПЗ		ЛР		КР / КП			
	Л	в т.ч. ПП	ПЗ	в т.ч. ПП	ЛР	в т.ч. ПП				
Семестр 7										
Стиль программирования	10				15			23	**48**	Лабораторная работа №1, 2
Проектирование программы	10				15			23	**48**	Лабораторная работа №3
Классические паттерны проектирования	10				15			23	**48**	Лабораторная работа №4, 5
Консультации										
Контроль промежуточной аттестации										**Диф. зачет (зачёт с оценкой)**
ИТОГО за семестр:	**30**				**45**			**69**	**144**	
Семестр 8										
Гибкие технологии	7				11			28	**46**	Лабораторная работа №6, 7
Кодирование, рефакторинг и управление исходным кодом	7				11			30	**48**	Лабораторная работа №1
Планирование и управление проектом	8				11			30	**49**	Лабораторная работа №2, 3, 4
Консультации									***1***	
Контроль промежуточной аттестации										**Экзамен**
ИТОГО за семестр:	**22**				**33**			**88**	**144**	
Итого за весь период	**52**				**78**			**157**	**288**	

Примечание: Л – лекция; ПЗ – практическое занятие, семинар; ЛР – лабораторная работа; ПП – практическая подготовка; КР / КП – курсовая работа / курсовой проект; КПА – контроль промежуточной аттестации; КС – консультации; СР – самостоятельная работа

Таблица 3. Матрица соотнесения разделов, тем учебной дисциплины (модуля) и формируемых компетенций

Раздел, тема дисциплины (модуля)	Кол-во часов	Код компетенции								Общее количество компетенций
		ОПК-4	ОПК-5	ПК-2	ПК-6	ПК-7	ПК-9	ПК-10	ПК-11	
Стиль программирования	48	+	+	+	+	+	+	+	+	8
Проектирование программы	48	+	+	+	+	+	+	+	+	8
Классические паттерны проектирования	48	+	+	+	+	+	+	+	+	8
Гибкие технологии	46	+	+	+	+	+	+	+	+	8
Кодирование, рефакторинг и управление исходным кодом	48	+	+	+	+	+	+	+	+	8
Планирование и управление проектом	49	+	+	+	+	+	+	+	+	8
Итого	287									

Краткое содержание каждой темы дисциплины (модуля)

№ раздела	Наименование раздела дисциплины	Содержание
1	Стиль программирования	Стиль кода, Задача стыковки интерфейсов, Интерфейсы, Поиск ошибок и хороший код, Основные проблемы при проектировании
2	Проектирование программы	Проектирование интерфейсов, Структуры данных
3	Классические паттерны проектирования	Model-View-Controller, Шаблоны, Паттерны проектирования
4	Гибкие технологии	Технологии и процесс разработки ПО, Моделирование и узкие места традиционного Waterfall процесса, TDD и тестирование, TDD

5	Кодирование, рефакторинг и управление исходным кодом	Рефакторинг и стандарты, Стандарты кода
6	Планирование и управление проектом	Экстремальное программирование в действии, Игра в планирование, Моделирование методологий, Методологии, Методики экстремального программирования, Построение команд и процессов

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю)

Лекционные занятия

Основной формой реализации теоретического обучения является лекция, которая представляет собой систематическое, последовательное изложение преподавателем-лектором учебного материала теоретического характера. Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Порядок подготовки лекционного занятия включает в себя выполнение следующих этапов:

- изучение требований программы дисциплины;
- определение целей и задач лекции;
- разработка плана проведения лекции;
- подбор литературы (ознакомление с методической литературой, публикациями периодической печати по теме лекционного занятия);
- отбор необходимого и достаточного по содержанию учебного материала;
- определение методов, приемов и средств поддержания интереса, внимания, стимулирования творческого мышления студентов;
- написание конспекта лекции.

Лекция должна включать следующие разделы:

- формулировку темы лекции;
- указание основных изучаемых разделов или вопросов и предполагаемых затрат времени на их изложение;
- изложение вводной части;
- изложение основной части лекции;
- краткие выводы по каждому из вопросов;
- заключение;
- рекомендации литературных источников по излагаемым вопросам.

Лабораторные занятия

Лабораторное занятие − целенаправленная форма организации педагогического процесса, направленная на углубление научно-теоретических знаний и овладение определенными методами работы, в процессе которых вырабатываются умения и навыки выполнения тех или иных учебных действий в данной сфере науки. Они развивают научное мышление и речь, позволяют проверить знания студентов и выступают как средства оперативной обратной связи.

Правильно организованные лабораторные занятия ориентированы на решение следующих задач:

- обобщение, систематизация, углубление, закрепление полученных на лекциях и в процессе самостоятельной работы теоретических знаний по дисциплине (предмету);
- формирование практических умений и навыков, необходимых в будущей профессиональной деятельности, реализация единства интеллектуальной и практической деятельности;
- выработка при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Состав заданий для лабораторного занятия должен быть спланирован с расчетом, чтобы за отведенное время они могли быть качественно выполнены большинством учащихся.

Лабораторные занятия должны так быть организованы, чтобы студенты ощущали нарастание сложности выполнения заданий, испытывали бы положительные эмоции от переживания собственного успеха в учении, поисками правильных и точных решений.

Самостоятельная работа

Самостоятельная работа – это вид учебной деятельности, которую студент совершает в установленное время и в установленном объеме индивидуально или в группе, без непосредственной помощи преподавателя (но при его контроле), руководствуясь сформированными ранее представлениями о порядке и правильности выполнения действий.

В учебном процессе образовательного учреждения выделяются два вида самостоятельной работы:

- аудиторная – выполняется на учебных занятиях, под непосредственным руководством преподавателя и по его заданию (выполнение самостоятельных работ; выполнение контрольных и практических работ; решение задач);
- внеаудиторная – выполняется по заданию преподавателя, но без его непосредственного участия (подготовка к аудиторным занятиям; изучение учебного материала, вынесенного на самостоятельную проработку; выполнение домашних заданий разнообразного характера; выполнение индивидуальных заданий, направленных на развитие у студентов самостоятельности и инициативы; подготовка к контрольной работе). Внеаудиторные самостоятельные работы представляют собой логическое продолжение аудиторных занятий, проводятся по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Лекция

- Лекция – основной вид обучения в вузе.
- В лекции излагаются основные положения теории, ее понятия и законы, приводятся факты, показывающие связь теории с практикой.
- Накануне лекции необходимо повторить содержание предыдущей лекции (а также теорию по изучаемой теме в школьных учебниках геометрии, если эта тема была представлена в них), а затем посмотреть тему очередной лекции по программе (по плану лекций).
- Полезно вести записи (конспекты) лекций: для непонятных вопросов оставлять место при работе над темой лекции с учебными пособиями.
- Записи лекций следует вести в отдельной тетради, оставляя место для дополнений во время самостоятельной работы.
- При конспектировании лекций выделяйте главы и разделы, параграфы, подчеркивайте основное.

Лабораторное занятие

- Лабораторное занятие – наиболее активный вид учебных занятий в вузе. Он предполагает самостоятельную работу над лекциями и учебными пособиями.
- К каждому лабораторному занятию нужно готовиться. Подготовку следует начинать с повторения теории (по записям лекций или по учебному пособию). После этого нужно решать задачи из предложенного домашнего задания.

Организация самостоятельной работы

Самостоятельность в учебной работе способствует развитию заинтересованности студента в изучаемом материале, вырабатывает у него умение и потребность самостоятельно получать знания, что весьма важно для специалиста с высшим образованием. Самостоятельная работа студентов представлена в следующих формах:

- работа с учебной литературой и конспектом лекций с целью подготовки к лабораторным занятиям, составление конспектов тем, выносимых на самостоятельную проработку;
- систематическое выполнение домашних работ.

Таблица 4. Содержание самостоятельной работы обучающихся

для очной формы обучения

Вопросы, выносимые на самостоятельное изучение	Кол-во часов	Форма работы
Стиль программирования	23	Выполнение лабораторной работы № 1, 2
Проектирование программы	23	Выполнение лабораторной работы № 3
Классические паттерны проектирования	23	Выполнение лабораторной работы № 4, 5
Гибкие технологии	28	Выполнение лабораторной работы № 6, 7
Кодирование, рефакторинг и управление исходным кодом	30	Выполнение лабораторной работы № 1
Планирование и управление проектом	30	Выполнение лабораторной работы № 2, 3, 4

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины (модуля

Дисциплиной «Проектирование программного обеспечения» письменные работы не предусмотрены.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине «Проектирование программного обеспечения» могут использоваться электронное обучение и дистанционные образовательные технологии.

6.1. Образовательные технологии

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line или off-line в формах.

Таблица 5. Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Стиль программирования	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*
Проектирование программы	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*
Классические паттерны проектирования	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*
Гибкие технологии	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*
Кодирование, рефакторинг и управление исходным кодом	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*
Планирование и управление проектом	*Обзорная лекция*	*Не предусмотрено*	*Выполнение лабораторной работы*

6.2. Информационные технологии

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- система управления обучением LMS Moodle;
- использование возможностей Интернета в учебном процессе (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление обучающихся с оценками и т.д.);
- использование электронных учебников и различных сайтов (например, электронные библиотеки, журналы и т.д.) как источник информации;
- использование возможностей электронной почты;
- использование средств представления учебной информации (электронных учебных пособий, применение новых технологий для проведения занятий с использованием презентаций и т.д.);

- использование интерактивных средств взаимодействия участников образовательного процесса (технологии дистанционного или открытого обучения в глобальной сети);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т.е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс).

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

Наименование программного обеспечения	Назначение
Adobe Reader	Программа для просмотра электронных документов
Платформа дистанционного обучения LMS Moodle	Виртуальная обучающая среда
Microsoft Office 2013, Microsoft Office Project 2013, Microsoft Office Visio 2013	Пакет офисных программ
7-zip	Архиватор
Microsoft Windows 7 Professional	Операционная система
Kaspersky Endpoint Security	Средство антивирусной защиты
Google Chrome	Браузер
OpenOffice	Пакет офисных программ

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронная библиотека «Астраханский государственный университет» собственной генерации на платформе ЭБС «Электронный Читальный зал – БиблиоТех». https://biblio.asu.edu.ru
2. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». www.studentlibrary.ru.
3. Электронная библиотечная система издательства ЮРАЙТ, раздел «Легендарные книги». www.biblio-online.ru
4. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информсистем». https://library.asu.edu.ru

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине «Проектирование программного обеспечения» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин и прохождением практик, а в процессе освоения дисциплины– последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6. Соответствие разделов, тем дисциплины (модуля), результатов обучения по дисциплине (модулю) и оценочных средств

Контролируемый раздел, тема дисциплины (модуля)	Код контролируемой компетенции	Наименование оценочного средства
Стиль программирования	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 1, 2
Проектирование программы	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 3
Классические паттерны проектирования	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 4, 5
Гибкие технологии	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 6, 7

Контролируемый раздел, тема дисциплины (модуля)	Код контролируемой компетенции	Наименование оценочного средства
Кодирование, рефакторинг и управление исходным кодом	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 1
Планирование и управление проектом	ОПК-4, ОПК-5, ПК-2, ПК-6, ПК-7, ПК-9, ПК-10, ПК-11	Лабораторная работа № 2, 3, 4

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Таблица 7. Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 8. Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание по подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задания

7.3. Контрольные задания и иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Лабораторные работы 1

№ п/п	Номер раздела дисциплины	Наименование лабораторной работы	Трудоемкость, часов
1	5	Рефакторинг и стандарты	5
2	6	Экстремальное программирование в действии	5
3	6	Игра в планирование	5
4	6	Моделирование методологий	5

Примеры заданий к лабораторной работе 1 «Рефакторинг и стандарты»

1. Студенты должны внести изменения в программу, используя рефакториг
2. Студенты должны знать основные виды рефакторинга
3. Студенты должны уметь применять рефакторинг в программах
4. Студенты должны понимать, в каких ситуаций стоит применять рефакторинг

Примеры заданий к лабораторной работе 2 «Экстремальное программирование в действии»

1. Написать программу использую технику парного программирования
2. Знать основные методики управления требованиями
3. Уметь использовать технику парного программирования
4. Уметь осуществлять управление требованиями
5. Владеть базовыми методиками проектирования

Примеры заданий к лабораторной работе 3 «Игра в планирование»

1. Распределить роли в команде в различных ситуациях
2. Уметь планировать распределение ролей, версий и итераций
3. Знать основы организации рабочего времени, общения между членами команды, рабочего места разработчика
4. Уметь организовывать процесс разработки

Примеры заданий к лабораторной работе 4 «Моделирование методологий»

1. Описать все элементы экстремального программирования применительно к проекту
2. Знать принципы экстремального программирования и scrum
3. Уметь выявлять отсутствующие компоненты экстремального программирования

Порядок предоставления отчета по лабораторной работе

Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе. Время, отводимое на выполнение – 4 часа. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя.

Шаблон отчета по лабораторной работе

Отчет по лабораторной работе № ____

«Название лабораторной работы»

1. Цель и задачи лабораторной работы: ____
2. Методика проведения исследования: ____
3. Анализ погрешностей: ____
4. Результаты: ____
5. Выводы: ____

Требования к выполнению лабораторной работы

Отчеты по лабораторным работам должны быть отправлены на электронную почту преподавателя не позднее, чем через две недели после выдачи задания. Полученные выводы и графический материал должны быть информативными и корректными.

Лабораторные работы 2

№ п/п	Номер раздела дисциплины	Наименование лабораторной работы	Трудоемкость, часов
1	1	Поиск ошибок и хороший код	6
2	1	задача стыковки интерфейсов	5
3	2	проектирование интерфейсов	6
4	3	Model-View-Controller	4
5	3	Шаблоны	4
6	4	Моделирование и узкие места традиционного Waterfall процесса	4
7	4	TDD и тестирование	3

Лабораторная работа выполняется в рамках каждого раздела курса с целью усвоения прослушанного студентом теоретического материала.
Лабораторные работы должны быть сданы в период прочтения курса.
Сдача работы представляет собой предоставление отчета в свободной форме в письменном или электронном виде и, в случае необходимости, устные ответы на уточняющие вопросы по отдельным задачам.
Примеры заданий к лабораторной работе 1 «Поиск ошибок и хороший код»

1. Найти стилистические ошибки в заданной программе
2. Предложить вариант программы, лишенный указанных проблем
3. Написать программу, в которой будут остутствовать основные признаки плохо спроектированной программы для решения заданной задачи

Примеры заданий к лабораторной работе 2 «Задача стыковки интерфейсов»

1. Построить интерфейс небольшой библиотеки
2. Уметь писать программы, состоящие из нескольких модулей, связанных интерфейсами

Примеры заданий к лабораторной работе 3 «Проектирование интерфейсов»

1. Студенты должны спроектировать программу для решения заданной задачи
2. Студенты должны понимать принципы проектирования программы
3. Студенты должны уметь спроектировать структуру данных
4. Студенты должны уметь проектировать и применять на практике абстрактные типы данных

Примеры заданий к лабораторной работе 4 «Model-View-Controller»

1. Студенты должны спроектировать программу в соответствии с заданием
2. Студенты должны понимать принцип разделения программы на части: модель, вид, контроллер
3. Студенты должны уметь анализировать правильность проектирования программы по зависимости ее частей друг от друга
4. Студенты должны уметь делить программу на классы, а классы на пакеты в соответствии с их предназначением

Примеры заданий к лабораторной работе 5 «Шаблоны»

1. Разработать программу с использованием шаблона Factory method
2. Разработать программу с использованием шаблона Builder
3. Разработать программу с использованием шаблона Singleton
4. Разработать программу с использованием шаблона Listener
5. Разработать программу с использованием шаблона Visitor
6. Разработать программу с использованием шаблона Adapter
7. Разработать программу с использованием шаблона Strategy

Примеры заданий к лабораторной работе 6 «Моделирование и узкие места традиционного Waterfall процесса»

1. Описать плюсы и минусы традиционного подхода к разрабоке ПО и гибких технологий в различных ситуациях
2. Понимать основы гибких технологий
3. Уметь оценивать правильность построения процесса разработки в зависимости от требований и ресурсов

Примеры заданий к лабораторной работе 7 «TDD и тестирование»

1. Разработать программу, полностью покрытую тестами
2. Уметь строить тесты для программы
3. Знать основные виды тестирования
4. Уметь разрабатывать программу с использованием модульного и функционального тестирования
5. Уметь осуществлять разработку через тестирование

Порядок предоставления отчета по лабораторной работе

Отчет по лабораторной работе представляется в печатном виде в формате, предусмотренном шаблоном отчета по лабораторной работе. Время, отводимое на выполнение – 4 часа. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя.

Шаблон отчета по лабораторной работе

Отчет по лабораторной работе № ____

«Название лабораторной работы»

1. Цель и задачи лабораторной работы: ____
2. Методика проведения исследования: ____
3. Анализ погрешностей: ____
4. Результаты: ____
5. Выводы: ____

Требования к выполнению лабораторной работы

Отчеты по лабораторным работам должны быть отправлены на электронную почту преподавателя не позднее, чем через две недели после выдачи задания. Полученные выводы и графический материал должны быть информативными и корректными.

Перечень вопросов и заданий, выносимых на зачет

Зачет проводится в устной форме. Обучающемуся предлагается ответить на один вопрос из перечня, выбранный случайным образом. Преподаватель может задавать дополнительные вопросы в рамках лекционного материала.

Перечень вопросов:

1. Стиль кода: имена классов, переменных, методов.
2. Null и Optional. Недостатки и достоинства обоих подходов.
3. Корректность кода. Статические проверки кода.
4. Корректность кода. Continuous integration.
5. Корректность кода. Инварианты и динамические проверки, ассерты.
6. Обработка ошибок. Коды ошибок. Исключения. Checked, unchecked исключения в java.
7. ООП-дизайн. Single Responsibility Principle.
8. ООП-дизайн. Open Closed Principle.
9. ООП-дизайн. Liskov substitution Principle.
10. ООП-дизайн. Interface Segregation Principle.
11. ООП-дизайн. Dependency Inversion Principle.

Перечень вопросов и заданий, выносимых на экзамен

Порядок формирования билета: по одному вопросу из первой и второй части перечня вопросов к дифференцированному зачету, дополнительный вопрос из любой части списка

Требования к ответу: полный развернутый ответ на каждый из вопросов

Возможность дополнительных вопросов: возможны уточнения в рамках лекционного материала

1. Модульное тестирование.
2. Mock-объекты для тестирования кода.
3. Test driven development.
4. Тестирование и дизайн кода (как и почему тесты улучшают дизайн программы).
5. Рефакторинг. Определение, когда и как стоит применять рефакторинг.
6. Рефакторинг. Код с душком: нагромождения и ООП-нарушители.
7. Рефакторинг. Код с душком: противники изменений, мусорный код, чрезмерные связи между классами.
8. Рефакторинг. Примеры рефакторингов.
9. Model View Controller.
10. Порождающие паттерны. Абстрактная фабрика, фабричный метод.

11. Порождающие паттерны. Строитель, одиночка, пул объектов.
12. Структурные паттерны. Адаптер, мост.
13. Структурные паттерны. Прокси, декоратор, компоновщик.
14. Структурные паттерны. Фасад, приспособленец.
15. Паттерны поведения. Цепочка обязаностей, наблюдатель.
16. Паттерны поведения. Стратегия, шаблонный метод.
17. Паттерны поведения. Визитор, null-объект.
18. Паттерны поведения. Состояние, интерпретатор.

Билет № 1

Вопрос 1. Рефакторинг. Код с душком: нагромождения и ООП-нарушитили.

Вопрос 2. Паттерны поведения. Визитор, null-объект.

Таблица 9. Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
Код и наименование проверяемой компетенции ***ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности***				
1.	Задание закрытого типа	*Выберите верное утверждение*. Какие программы можно отнести к системному ПО? а. операционные системы б. прикладные программы в. игровые программы	а	1-3
2.		*Выберите верное утверждение*. Какие программы можно отнести к системному ПО? а. драйверы б. текстовые редакторы в. электронные таблицы г. графические редакторы	а	1-3
3.		*Выберите верное утверждение*. Какие программы можно отнести к системному ПО? а. программа расчета заработной платы б. электронные таблицы в. СУБД (системы управления базами данных)	в	1-3
4.		*Выберите верное утверждение*. Какие программы НЕЛЬЗЯ отнести к системному ПО? а. игровые программы б. компиляторы языков программирования в. операционные системы г. системы управления базами данных	а	1-3
5.		*Выберите верное утверждение*. Какие программы можно отнести к прикладному ПО?	б	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. таблицы решений б. электронные таблицы в. СУБД (системы управления базами данных)		
6.	Задание открытого типа	Назовите не менее трех специфических особенностей ПО как продукта.	1. Продажа по низким ценам ниже себестоимости (лицензирование). 2. Низкие материальные затраты при создании программ. 3. Возможность создания программ небольшим коллективом или даже одним человеком. 4. Разнообразие решаемых задач с помощью программных средств.	2-5
7.		Что такое Mock-объекты?	Mock-объект (от англ. mock object, дословно — «объект-пародия», «объект-имитация», а также «подставка») — в объектно-ориентированном программировании — тип объектов, реализующих заданные аспекты моделируемого программного окружения. Mock-объект представляет собой конкретную фиктивную реализацию интерфейса, предназначенную исключительно для тестирования взаимодействия и относительно которого высказывается утверждение. В процедурном программировании аналогичная конструкция называется «dummy» (с англ. — «заглушка»). Функция, выдающая константу, или случайную величину из допустимого диапазона значений. Mock-объекты активно используются в разработке через тестирование.	8-10
8.		Опишите основную идею разработки через тестирование.	Техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам. Кент Бек, считающийся изобретателем этой техники, утверждал в 2003 году, что разработка через тестирование поощряет простой дизайн и внушает уверенность. В 1999 году при своём появлении разработка через тестирование была тесно связана с концепцией «сначала тест» (англ. test-first), применяемой в экстремальном программировании, однако позже выделилась как независимая методология.	8-10
9.		Какие недостатки свойственны разработке через тестирование?	Существуют задачи, которые невозможно (по крайней мере, на текущий момент) решить только при помощи тестов. В частности, TDD не позволяет механически продемонстрировать адекватность разработанного кода в области безопасности данных и взаимодействия между процессами. Разработку через тестирование сложно применять в тех случаях, когда для тести-	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			рования необходимо прохождение функциональных тестов. Требуется больше времени на разработку и поддержку, а одобрение со стороны руководства очень важно. Модульные тесты, создаваемые при разработке через тестирование, обычно пишутся теми же, кто пишет тестируемый код. Если разработчик неправильно истолковал требования к приложению, и тест, и тестируемый модуль будут содержать ошибку. Большое количество используемых тестов может создать ложное ощущение надежности, приводящее к меньшему количеству действий по контролю качества.	
10.		Какой подход к видимости кода используется при разработке через тестирование?	Набор тестов должен иметь доступ к тестируемому коду. С другой стороны, принципы инкапсуляции и сокрытия данных не должны нарушаться. Поэтому модульные тесты обычно пишутся в том же модуле или проекте, что и тестируемый код. Из кода теста может не быть доступа к приватным (англ. private) полям и методам. Поэтому при модульном тестировании может потребоваться дополнительная работа. В Java разработчик может использовать отражение (англ. reflection), чтобы обращаться к полям, помеченными как приватные. Модульные тесты можно реализовать во внутренних классах, чтобы они имели доступ к членам внешнего класса. В .NET Framework могут применяться разделяемые классы (англ. partial classes) для доступа из теста к приватным полям и методам. Важно, чтобы фрагменты кода, предназначенные исключительно для тестирования, не оставались в выпущенном коде. В Си для этого могут быть использованы директивы условной компиляции. Однако это будет означать, что выпускаемый код не полностью совпадает с протестированным. Систематический запуск интеграционных тестов на выпускаемой сборке поможет удостовериться, что не осталось кода, скрыто полагающегося на различные аспекты модульных тестов. Не существует единого мнения среди программистов, применяющих разработку через тестирование, о том, насколько осмысленно тестировать приватные, защищённые(англ. protected) методы, а также данные. Одни убеждены, что достаточно протестировать любой класс только через его публичный интерфейс, поскольку приватные переменные — это всего лишь деталь реализации, которая может меняться, и её изменения не должны отражаться на	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			наборе тестов. Другие утверждают, что важные аспекты функциональности могут быть реализованы в приватных методах и тестирование их неявно через публичный интерфейс лишь усложнит ситуацию: модульное тестирование предполагает тестирование наименьших возможных модулей функциональности.	
11.	Задание комбинированного типа	Можно ли отнести операционную систему к программному обеспечению? Поясните ответ.	Да, поскольку операционная система - это программное обеспечение, управляющее компьютерами (включая микроконтроллеры) и позволяющее запускать на них прикладные программы.	2-5
Код и наименование проверяемой компетенции ***ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения***				
12.	Задание закрытого типа	*Выберите верное утверждение*. Какие программы можно отнести к системному ПО? а. утилиты б. экономические программы в. статистические программы г. мультимедийные программы	а	1-3
13.		*Выберите верное утверждение*. Этап, занимающий наибольшее время в жизненном цикле программы? а. проектирование б. сопровождение в. тестирование г. программирование	б	1-3
14.		*Выберите верное утверждение*. Этап, занимающий наибольшее время при разработке программы? а. проектирование б. сопровождение в. тестирование г. программирование	в	1-3
15.		*Выберите верное утверждение*. Первый этап в жизненном цикле программы – это... а. автономное тестирование б. проектирование в. анализ требований г. формулирование требований	г	1-3
16.		*Выберите верное утверждение*. Один из необязательных этапов жизненного цикла программы – это...	в	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. программирование б. тестирование в. оптимизация г. проектирование		
17.	Задание открытого типа	Как используются модульные тесты при тестировании кода?	Модульные тесты тестируют каждый модуль по отдельности. Неважно, содержит ли модуль сотни тестов или только пять. Тесты, используемые при разработке через тестирование, не должны пересекать границы процесса, использовать сетевые соединения. В противном случае прохождение тестов будет занимать большое время, и разработчики будут реже запускать набор тестов целиком. Введение зависимости от внешних модулей или данных также превращает модульные тесты в интеграционные. При этом если один модуль в цепочке ведет себя неправильно, может быть не сразу понятно какой именно.	8-10
18.		Когда разрабатываемый код использует базы данных, веб-сервисы или другие внешние процессы, имеет смысл выделить покрываемую тестированием часть. Какие шаги при этом должны быть реализованы?	Везде, где требуется доступ к внешним ресурсам, должен быть объявлен интерфейс, через который этот доступ будет осуществляться. См. принцип инверсии зависимостей для обсуждения преимуществ этого подхода независимо от TDD. Интерфейс должен иметь две реализации. Первая, собственно предоставляющая доступ к ресурсу, и вторая, являющаяся fake- или mock-объектом. Всё, что делают fake-объекты, это добавляют сообщения вида «Объект person сохранен» в лог, чтобы потом проверить правильность поведения. Mock-объекты отличаются от fake- тем, что сами содержат утверждения (англ. assertion), проверяющие поведение тестируемого кода. Методы fake- и mock-объектов, возвращающие данные, можно настроить так, чтобы они возвращали при тестировании одни и те же правдоподобные данные. Они могут эмулировать ошибки так, чтобы код обработки ошибок мог быть тщательно протестирован. Другими примерами fake-служб, полезными при разработке через тестирование, могут быть: служба кодирования, которая не кодирует данные, генератор случайных чисел, который всегда выдает единицу. Fake- или mock-реализации являются примерами внедрения зависимости.	8-10
19.		Интеграционные тесты, которые изменяют данные в базе данных, должны откатывать состоянии базы данных к тому, которое было до запуска теста, даже если тест не прошёл. Какие, наиболее часто, для этого применяются техники?	Метод TearDown, присутствующий в большинстве библиотек для тестирования. try...catch...finally структуры обработки исключений, там где они доступны. Транзакции баз данных. Создание снимка (англ. snapshot) базы данных перед запуском тестов и откат к нему после окончания тестирования. Сброс базы данных в чистое состояние перед тестом, а не после них. Это может	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			быть удобно, если интересно посмотреть состояние базы данных, оставшееся после не прошедшего теста. Существуют библиотеки Moq, jMock, NMock, EasyMock, Typemock, jMockit, Unitils, Mockito, Mockachino, PowerMock или Rhino Mocks, а также sinon для JavaScript предназначенные упростить процесс создания mock-объектов.	
20.		Что понимают под рефакторингом кода?	Рефакторинг — это переработка исходного кода программы, чтобы он стал более простым и понятным. Рефакторинг не меняет поведение программы, не исправляет ошибки и не добавляет новую функциональность. Он делает код более понятным и удобочитаемым.	5-8
21.		Для чего применяется рефакторинг кода?	Стройный, хорошо структурированный код легко читается и быстро дорабатывается. Но редко удаётся сразу сделать его таким. Разработчики спешат, в процессе могут меняться требования к задаче, тестировщики находят баги, которые нужно быстро исправить, или возникают срочные доработки, и их приходится делать второпях. В результате даже изначально хорошо структурированный исходник становится беспорядочным и непонятным. Программисты знают, как легко завязнуть в этом хаосе. Причём неважно, чужой это код или собственный. Чтобы решить все эти проблемы, делается рефакторинг программы. В новом проекте он нужен, чтобы: 1.сохранить архитектуру проекта, не допустить потери структурированности; 2.упростить будущую жизнь разработчиков, сделать код понятным и прозрачным для всех членов команды; 3.ускорить разработку и поиск ошибок. Но любое приложение со временем устаревает: язык программирования совершенствуется, появляются новые функции, библиотеки, операторы, делающие код проще и понятнее. То, что год назад требовало пятидесяти строк, сегодня может решаться всего одной.	8-10
Код и наименование проверяемой компетенции ***ПК-2. Способен осуществлять интеграцию программных модулей и компонент и проверку работоспособности кода программного обеспечения.***				
22.	Задание закрытого типа	*Выберите верное утверждение*. Самый большой этап жизненного цикла программы – это... а. эксплуатация б. изучение предметной области в. программирование г. тестирование	а	1-3
23.		*Выберите верное утверждение*.	в	1-3

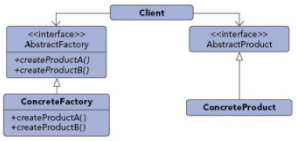
№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		Какой из этапов выполняется раньше? а. отладка б оптимизация в. программирование г. тестирование		
24.		*Выберите верное утверждение*. Что выполняется раньше? а. компиляция б. отладка в. компоновка г. тестирование	а	1-3
25.		*Выберите верное утверждение*. В стадии разработки программы НЕ входят: а. автоматизация программирования б. постановка задачи в. эскизный проект г. тестирование	а	1-3
26.		*Выберите верный ответ*. Способом оценки качества программы является... а. сравнение с аналогом б. наличие документации в. оптимизация программы г. структурирование алгоритма	а	1-3
27.	Задание открытого типа	Опишите схему разделения данных приложения Model View Controller.	Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо	5-8
28.		Опишите компоненты схемы Model View Controller.	Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние. Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели. Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений	5-8
29.		Какова концепция Model View Controller?	Концепция MVC позволяет разделить модель, представление и контроллер на три отдельных компонента: Модель Модель предоставляет данные и методы работы с ними: запросы в базу данных, проверка на корректность. Модель не зависит от представления (не знает как данные	8-10

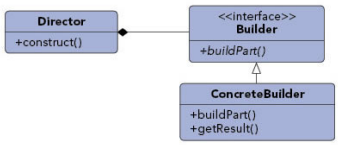
№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			визуализировать) и контроллера (не имеет точек взаимодействия с пользователем), просто предоставляя доступ к данным и управлению ими. Модель строится таким образом, чтобы отвечать на запросы, изменяя своё состояние, при этом может быть встроено уведомление «наблюдателей». Модель, за счёт независимости от визуального представления, может иметь несколько различных представлений для одной «модели» Представление Представление отвечает за получение необходимых данных из модели и отправляет их пользователю. Представление не обрабатывает введённые данные пользователя. Контроллер Контроллер обеспечивает «связь» между пользователем и системой. Контролирует и направляет данные от пользователя к системе и наоборот. Использует модель и представление для реализации необходимого действия.	
30.		Какова основная цель применения концепции Model View Controller?	Основная цель применения этой концепции состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида). За счёт такого разделения повышается возможность повторного использования кода. Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения. В частности, выполняются следующие задачи: К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели. Например, некоторые данные могут быть одновременно представлены в виде электронной таблицы, гистограммы и круговой диаграммы; Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных) — для этого достаточно использовать другой контроллер; Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (модели), вообще не будут осведомлены о том, какое представление будет использоваться.	8-10
31.		Опишите схему разделения данных приложения Model-View-ViewModel.	Model-View-ViewModel (MVVM) — шаблон проектирования архитектуры приложения. Представлен в 2005 году Джоном Госсманом (John Gossman) как модифика-	5-8

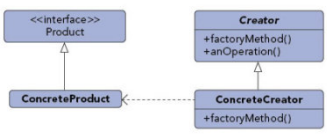
№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			ция шаблона Presentation Model. Ориентирован на соответствующие платформы разработки, такие как Windows Presentation Foundation, Silverlight от компании Microsoft, ZK framework.	
32.	Задание комбинированного типа	Существует ли связь между эффективностью и оптимизацией программы? Поясните ответ.	Да, связь существует. Оптимизация программы должна быть направлена на увеличение ее эффективности.	2-5
Код и наименование проверяемой компетенции ***ПК-6. Способен выполнять работы по созданию и сопровождению информационных систем.***				
33.	Задание закрытого типа	*Выберите верный ответ.* Что можно отнести к наиболее важному критерию качества? а. надежность б. быстродействие в. эффективность г. удобство в эксплуатации	а	1-3
34.		*Выберите верный ответ.* В каких единицах можно измерить надежность? а. км/час б. отказов/час в. Кбайт/сек г. операций/сек	б	1-3
35.		*Выберите верный ответ.* В каких единицах можно измерить быстродействие? а. км/час б. отказов/час в. Кбайт/сек г. операций/сек	г	1-3
36.		*Выберите верный ответ.* Что относится к этапу программирования? а. написание кода программы б. разработка интерфейса в. работоспособность г. анализ требований	а	1-3
37.		*Выберите верный ответ.* Выберите верную последовательность этапов программирования: а. компилирование, компоновка, отладка б. компоновка, отладка, компилирование в. отладка, компилирование, компоновка	а	1-3

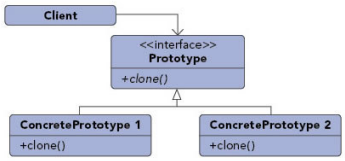
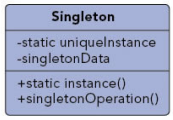
№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		г. компилирование, отладка, компоновка		
38.	Задание открытого типа	Опишите компоненты схемы Model-View-ViewModel.	Шаблон MVVM делится на три части: Модель (англ. Model) (так же, как в классической MVC) представляет собой логику работы с данными и описание фундаментальных данных, необходимых для работы приложения. Представление (англ. View) — графический интерфейс (окна, списки, кнопки и т. п.). Выступает подписчиком на событие изменения значений свойств или команд, предоставляемых Моделью Представления. В случае, если в Модели Представления изменилось какое-либо свойство, то она оповещает всех подписчиков об этом, и Представление, в свою очередь, запрашивает обновлённое значение свойства из Модели Представления. В случае, если пользователь воздействует на какой-либо элемент интерфейса, Представление вызывает соответствующую команду, предоставленную Моделью Представления. Модель Представления (англ. ViewModel) — с одной стороны, абстракция Представления, а с другой — обёртка данных из Модели, подлежащих связыванию. То есть, она содержит Модель, преобразованную к Представлению, а также команды, которыми может пользоваться Представление, чтобы влиять на Модель.	8-10
39.		Где обычно применяется схема Model-View-ViewModel?	MVVM удобно использовать вместо классического MVC и ему подобных в тех случаях, когда в платформе, на которой ведётся разработка, есть «связывание данных». В шаблонах проектирования MVC/MVP изменения в пользовательском интерфейсе не влияют непосредственно на Модель, а предварительно идут через Контроллер (англ. Controller) или Presenter. В таких технологиях как WPF и Silverlight есть концепция «связывания данных», позволяющая связывать данные с визуальными элементами в обе стороны. Следовательно, при использовании этого приёма применение модели MVC становится крайне неудобным из-за того, что привязка данных к представлению напрямую не укладывается в концепцию MVC/MVP.	8-10
40.		Опишите схему разделения данных приложения Model-View-Presenter.	Model-View-Presenter (MVP) — шаблон проектирования, производный от MVC, который используется в основном для построения пользовательского интерфейса. Элемент Presenter в данном шаблоне берёт на себя функциональность посредника (аналогично контроллеру в MVC) и отвечает за управление событиями пользовательского интерфейса (например, использование мыши) так же, как в других шаб-	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			лонах обычно отвечает представление.	
41.		Опишите компоненты шаблона Model View Controller	MVP — шаблон проектирования пользовательского интерфейса, который был разработан для облегчения автоматического модульного тестирования и улучшения разделения ответственности в презентационной логике (отделения логики от отображения): Модель (англ. Model) предоставляет данные для пользовательского интерфейса. Представление (англ. View) реализует отображение данных (Модели) и маршрутизацию пользовательских команд или событий Presenter'у. Presenter управляет Моделью и Представлением. Например извлекает данные из Модели и форматирует их для отображения в Представлении.	5-8
42.		Дайте определение понятию бизнес-логика в разработке информационных систем.	Бизнес-логика — в разработке информационных систем — совокупность правил, принципов, зависимостей поведения объектов предметной области (области человеческой деятельности, которую система поддерживает). Иначе можно сказать, что бизнес-логика — это реализация правил и ограничений автоматизируемых операций. Является синонимом термина «логика предметной области» (англ. domain logic). Бизнес-логика задает правила, которым подчиняются данные предметной области.	5-8
Код и наименование проверяемой компетенции ***ПК-7. Способен создавать и оценивать варианты архитектуры программного средства и осуществлять выбор среди них.***				
43.	Задание закрытого типа	*Выберите верный ответ.* Что относится к инструментальным средствам программирования? а. компиляторы, интерпретаторы б. СУБД в. BIOS г. ОС	а	1-3
44.		*Выберите верный ответ.* Как называется доступ, при котором записи файла читаются в физической последовательности? а. прямым б. простым в. последовательным г. основным	в	1-3
45.		*Выберите верный ответ.* Как называется доступ, при котором записи файла читаются в произвольной последовательности?	а	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. прямым б. простым в. последовательным г. основным		
46.		*Выберите верный ответ*. Что НЕ относится к методам программирования? а. логическое программирование б. структурное программирование в. модульное программирование	а	1-3
47.		*Выберите верный ответ*. Укажите НЕ правильное условие для создания имен. а. можно использовать большие буквы б. из имен лучше выбрасывать гласные в. длинное имя можно сократить г. имена могут содержать пробелы	г	1-3
48.	Задание открытого типа	Для чего используется порождающий шаблон «Абстрактная фабрика»?	Предоставляет интерфейс, который делегирует вызовы создания одному или нескольким конкретным классам для поставки определенных объектов. Можно использовать: ● Если создание объектов должно быть независимым от использующей их системы. ● Системы должны быть способны использовать несколько семейств объектов. ● Семейства объектов должны использоваться вместе. ● Библиотеки должны быть опубликованы без раскрытия деталей реализации. ● Конкретные классы должны быть отделены от клиентов.	5-8
49.		Какова структура порождающий шаблон «Абстрактная фабрика»?	 Абстрактная фабрика (Abstract Factory) объявляет методы создания различных абстрактных продуктов (Abstract Product). Конкретные фабрики (Concrete Factory) создают вариации продуктов. Клиент (Client) может работать с любыми вариациями продуктов через абстрактные интерфейсы.	8-10
50.		Для чего используется порождающий шаблон «Строитель»?	Позволяет динамически создавать сложные объекты на основе легко взаимозаменяемых алгоритмов.	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			Можно использовать: ● Если алгоритмы создания объектов должны быть отделены от системы. ● Требуется наличие нескольких представлений алгоритмов создания. ● Необходимо добавление новой функциональности создания без изменения основного кода. ● Нужен контроль над процессом создания во время выполнения.	
51.		Какова структура порождающий шаблон «Строитель»?	 Интерфейс строителя (Builder) объявляет шаги конструирования продуктов, общие для всех конкретных строителей. Директор(Director) определяет порядок вызова строительных шагов для производства определенного вида продуктов. Конкретные строители (Concrete Builder) реализуют строительные шаги, производя различные продукты.	5-8
52.		Для чего используется порождающий шаблон «Фабричный метод»?	Открывает метод для создания объектов, позволяя подклассам контролировать процесс создания. Можно использовать: ● Если класс не должен знать, какие подклассы он будет создавать. ● Нужно дать подклассам возможность определять, какие объекты будут создаваться. ● Родительским классам нужно отложить создание объектов для своих подклассов.	5-8
53.	Задание комбинированного типа	Можно ли внутри цикла поместить еще один цикл? Поясните ответ.	Да, в большинстве языков программирования допускается использование вложенных циклов	2-5
Код и наименование проверяемой компетенции ***ПК-9. Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.***				
54.	Задание закрытого типа	*Выберите верный ответ.* Какие символы НЕ допускаются в именах переменных? а. пробелы б. цифры в. подчеркивания	а	1-3
55.		*Выберите верный ответ.* Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной?	б	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. прямым указанием б. венгерской нотацией в. структурным программированием г. поляризацией		
56.		*Выберите верный ответ*. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки? а. для экономических задач б. для системных задач в. для инженерных задач	а	1-3
57.		*Выберите верный ответ*. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата? а. для экономических задач б. для системных задач в. для инженерных задач	в	1-3
58.		*Выберите верный ответ*. На каком этапе производится выбор языка программирования? а. отладка б. тестирование в. программирование г. проектирование	г	1-3
59.	Задание открытого типа	Какова структура порождающий шаблон «Фабричный метод»?	 Создатель (Creator) объявляет фабричный метод, который должен возвращать новые продукты. Продукт (Product) определяет общий интерфейс объектов, которые создают подклассы Creator. Конкретные создатели (Concrete Creator) по-разному реализуют фабричный метод, производя различные конкретные продукты (Concrete Products)	5-8
60.		Для чего используется порождающий шаблон «Прототип»?	Клонирование существующих объектов. ● Если композиция, создание и представление объектов должны быть отделены от системы. ● Создаваемые классы задаются во время выполнения. ● В объекте существует ограниченное число комбинаций состояний. ● Требуются объекты или структуры объектов, которые идентичны или	5-8

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			очень похожи на другие существующие объекты или структуры объектов. ● Первоначальное создание каждого объекта является ресурсоемкой операцией.	
61.		Какова структура порождающий шаблон «Прототип»?	 Клиент (Client) обращается к объекту через общий интерфейс прототипов для создания копии. Интерфейс прототипов (Prototype) описывает операции клонирования. Конкретный прототип (Concrete Prototype) реализует клонирование себя вместе со связанными объектами.	5-8
62.		Для чего используется порождающий шаблон «Одиночка»?	Гарантирует, что в системе находится только один экземпляр класса. Можно использовать: Если требуется ровно один экземпляр класса. Необходим контролируемый доступ к одному объекту.	2-5
63.		Какова структура порождающий шаблон «Одиночка»?	 Одиночка (Singleton)определяет статический метод instance(), который возвращает единственный экземпляр своего класса. Конструктор одиночки скрыт от клиентов. Вызов метода instance() должен быть единственным способом получить объект этого класса.	5-8
64.	Задание комбинированного типа	Можно ли использовать комбинацию языков программирования в рамках одного проекта? Поясните ответ.	Да, это допускается. Соглашение о вызове позволяют разным языкам вызывать функции из библиотек, написанных на других языках.	1-3
Код и наименование проверяемой компетенции ***ПК-10. Разработка и реализация архитектуры программного обеспечения.***				
65.	Задание закрытого типа	*Выберите верный ответ.* Для решения экономических задач характерно применение... а. СУБД б. языков высокого уровня в. языков низкого уровня г. применение сложных математических расчетов	а	1-3
66.		*Выберите верный ответ.* Для решения инженерных задач характерно применение...	г	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. СУБД б. языков высокого уровня в. ОС г. САПР		
67.		*Выберите верный ответ*. Что может служить причиной синтаксических ошибок? а. ошибки в исходных данных б. плохое знание языка программирования в. ошибки, допущенные на более ранних этапах г. неправильное применение процедуры тестирования	б	1-3
68.		*Выберите верный ответ*. Что такое автоматизация программирования? а. создание исходного кода при помощи компилятора б. создание исходного кода без разработки алгоритма в. создание исходного кода программными средствами	в	1-3
69.		*Выберите верный ответ*. В чем заключается сущность автоматизации программирования? а. создание программы без написания ее текста б. получение готовой программы без выполнения компоновки в. отсутствие компиляции	а	1-3
70.	Задание открытого типа	Что понимается под шаблонами проектирования?	Шаблоны проектирования — это руководства по решению повторяющихся проблем. Это не классы, пакеты или библиотеки, которые можно было бы подключить к вашему приложению и сидеть в ожидании чуда. Они скорее являются методиками решения определенных проблем в определенных ситуациях.	2-5
71.		Что понимают под адаптером?	Адаптер — структурный шаблон проектирования, предназначенный для организации использования функций объекта, недоступного для модификации, через специально созданный интерфейс.	2-5
72.		Что понимают под термином мост?	Мост — структурный шаблон проектирования, используемый в проектировании программного обеспечения чтобы разделять абстракцию и реализацию так, чтобы они могли изменяться независимо. Шаблон мост использует инкапсуляцию, агрегирование и может использовать наследование	2-5

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			для того, чтобы разделить ответственность между классами.	
73.		Что понимают под термином компоновщик?	Компоновщик — структурный шаблон проектирования, объединяющий объекты в древовидную структуру для представления иерархии от частного к целому. Компоновщик позволяет клиентам обращаться к отдельным объектам и к группам объектов одинаково. Паттерн определяет иерархию классов, которые одновременно могут состоять из примитивных и сложных объектов, упрощает архитектуру клиента, делает процесс добавления новых видов объекта более простым.	2-5
74.		Что понимают под термином декоратор?	Декоратор — структурный шаблон проектирования, предназначенный для динамического подключения дополнительного поведения к объекту. Шаблон декоратор предоставляет гибкую альтернативу практике создания подклассов с целью расширения функциональности.	2-5
Код и наименование проверяемой компетенции ***ПК-11. Разработка и сопровождение программных проектов***				
75.	Задание закрытого типа	*Выберите верный ответ.* Один из методов автоматизации программирования... а. структурное программирование б. модульное программирование в. визуальное программирование г. объектно-ориентированное программирование	в	1-3
76.		*Выберите верный ответ.* Что из перечисленного легко поддается автоматизации? а. интерфейс б. работа с файлами в. сложные логические задачи г. алгоритмизация	а	1-3
77.		*Выберите верный ответ.* Что можно отнести к критериям оптимизации? а. размер программы и ее эффективность б. время выполнения или размер требуемой памяти в. независимость модулей г. качество программы, ее надежность	б	1-3
78.		*Выберите верный ответ.* В чем сущность модульного программирования?	а	1-3

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		а. в разбиении программы на отдельные функционально независимые части б. в разбиении программы на отдельные процедуры и функции в. в разбиении программы на процедуры и функции		
79.		*Выберите верный ответ*. В чем заключается независимость модуля? а. в разработке и написании независимо от остальных модулей б. в разработке и написании независимо от других модулей в. в независимости от работы основной программы	а	1-3
80.	Задание открытого типа	Что понимают под термином фасад?	Фасад — структурный шаблон проектирования, позволяющий скрыть сложность системы путём сведения всех возможных внешних вызовов к одному объекту, делегирующему их соответствующим объектам системы.	2-5
81.		Что понимают под термином приспособленец?	Приспособленец — структурный шаблон проектирования, при котором объект, представляющий себя как уникальный экземпляр в разных местах программы, по факту не является таковым.	2-5
82.		Что понимают под термином заместитель?	Заместитель — структурный шаблон проектирования, который предоставляет объект, который контролирует доступ к другому объекту, перехватывая все вызовы (выполняет функцию контейнера).	2-5
83.		Опишите паттерн «Цепочка обязанностей».	Паттерн "Цепочка обязанностей" (далее также — "Цепочка") позволяет избежать тесной связи и взаимного влияния между отправителем и получателем запроса, предоставляя нескольким объектам возможность последовательно обрабатывать запрос. В рассматриваемом паттерне многочисленные объекты ссылаются друг на друга, формируя цепочку объектов. Запрос передаются по цепочке до тех пор, пока один из объектов не осуществит его окончательную обработку.	5-8
84.		Опишите паттерн «Наблюдатель».	Паттерн "Наблюдатель" широко используется в веб-приложениях — MutationObserver, IntersectionObserver, PerformanceObserver, ResizeObserver, ReportingObserver. Все эти API можно рассматривать как примеры применения "Наблюдателя". Кроме того, данный паттерн также используется для перманентного мониторинга событий и реагирования на модификацию данных. В "Наблюдателе" существует две основные	8-10

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			роли: Субъект/Subject и Наблюдатель/Observer. Паттерн "Наблюдатель" определяет отношение один ко многим (one-to-many), позволяя нескольким объектам-наблюдателям одновременно следить за наблюдаемым субъектом. При изменении состояния наблюдаемого субъекта об этом уведомляются все объекты-наблюдатели, чтобы они, в свою очередь, могли обновить собственное состояние.	
85.	Задание комбинированного типа	Верно ли утверждение: при модульном программирование программа создается по частям в определенной последовательности. Поясните ответ.	Утверждение неверно, поскольку при модульном программировании создание программы осуществляется по частям в произвольном порядке.	2-5

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Таблица 10. Технологическая карта рейтинговых баллов по дисциплине (модулю)

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
7 семестр				
Основной блок				
1.	*Выполнение лабораторных работ*	5/18	90	Сроки указаны в Moodle
Всего			**90**	-
Блок бонусов				
2.	*Посещение занятий*		10	
Всего			**10**	-
ИТОГО			**100**	-
8 семестр				
Основной блок				
3.	*Выполнение лабораторных работ*	6/15	90	Сроки указаны в Moodle
Всего			**90**	-
Блок бонусов				
4.	*Посещение занятий*		10	
Всего			**10**	-
ИТОГО			**100**	-

Таблица 11. Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине

Сумма баллов	Оценка по 4-балльной шкале
90–100	5 (отлично)
85–89	4 (хорошо)
75–84	
70–74	
65–69	3 (удовлетворительно)
60–64	
Ниже 60	2 (неудовлетворительно)

При реализации дисциплины в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература

1. Hopcroft J. E., Motwani R., Ullman J. D. Introduction to Automata Theory, Languages, and Computation (3rd Edition). — Addison-Wesley, Boston, MA, USA, 2006. — 750 с.
2. Шень А. Программирование: теоремы и задачи. — М.: МЦНМО, 2014. — 296 с.
3. Шень А., Верещагин Н. Языки и исчисления. — М.: МЦНМО, 2012. — 240 с.
4. Верещагин, Н. К. Колмогоровская сложность и алгоритмическая случайность [Электронный ресурс] / Н. К. Верещагин, В. А. Успенский, А. Шень. — Электрон. дан. — СПб: Лань, 2013. — 575 с. — Режим доступа: https://e.lanbook.com/book/56395 — Загл. с экрана.
5. Иванова, О. Г., Громов, Ю. Ю. Методы и средства проектирования информационных систем и технологий. Основы UML Тамбов: Тамбовский государственный технический университет, ЭБС АСВ 2020 https://www.iprbooks hop.ru/115768.html

8.2. Дополнительная литература:

1. Кривцова, И. Е. Основы дискретной математики. Часть 1. Учебное пособие [Электронный ресурс] / И. Е. Кривцова, И. С. Лебедев, А. В. Настека. — Электрон. дан. — СПб: ИТМО, 2016. — 92 с. — Режим доступа: http://books.ifmo.ru/book/1869/osnovy_diskretnoy_matematiki._chast_1._uchebnoe_posobie.htm — Загл. с экрана.
2. Теофили, Т. Глубокое обучение для поисковых систем : руководство / Т. Теофили ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 318 с. — ISBN 978-5-97060-776-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140574

8.3. Интернет-ресурсы, необходимые для освоения дисциплины

1. Вики-конспекты. — http://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница
2. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем»: https://library.asu.edu.ru
3. Корпоративный проект Ассоциации региональных библиотечных консорциумов (АРБИКОН) «Межрегиональная аналитическая роспись статей» (МАРС): http://mars.arbicon.ru
4. Единое окно доступа к образовательным ресурсам http://window.edu.ru

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для проведения **лекционных занятий**:

1. Используется аудитория, оборудованная необходимым количеством столов, стульев, доской маркерной и электронной.
2. Аудитория должна иметь следующие нормы освещенности
   - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.
   - СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
3. Электронная доска должна быть подключена к сети Интернет.

Для проведения **лабораторных занятий**:

1. Лабораторные занятия проводятся с группами или подгруппами не более 15 человек.
2. Аудитория должна быть оснащена необходимым количеством столов, стульев, доской маркерной и электронной.

4. Аудитория должна иметь следующие нормы освещенности
   - СНиП 23-05-95 «Естественное и искусственное освещение» норма освещенности аудиторий ВУЗов 400 Лк.
   - СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» пункт 3.3.3. «Общее освещение в помещениях общественных зданий должно быть равномерным».
5. В аудитории должно быть не менее 15 компьютеров, находящихся в исправном состоянии.
6. Расположение компьютеров в аудитории должно позволять преподавателю подойти к рабочему месту студента.
7. Компьютеры должны быть соединены локальной сетью со скоростью не менее 1 Гбит/с и подключены к сети Интернет.
8. Компьютеры должны обладать минимальными характеристиками:
   - Объем оперативной памяти 16 Гб
   - Накопитель SDD 500 Гб
   - Процессор 12th Gen Intel(R) Core(TM) i3-12100
   - Видеоадаптер Intel(R) UHD Graphics 730

10. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ (МОДУЛЯ) ПРИ ОБУЧЕНИИ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков. Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания. Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.). Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь. Текущий контроль успеваемости осуществляется в устной форме. При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую

техническую помощь (занять рабочее место, передвигаться по аудитории, прочитать задание, оформить ответ, общаться с преподавателем).