



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФГБОУ ВО «ИГУ»

Кафедра естественнонаучных дисциплин

УТВЕРЖДАЮ

декан факультета бизнес-коммуникаций и
информатики

М.Г. Синчурина

«19» марта 2025 г

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля)	Б1.В.05 Тестирование программного обеспечения <i>(индекс дисциплины по учебному плану, наименование дисциплины (модуля))</i>
Направление подготовки:	09.03.03 Прикладная информатика <i>(код, наименование направления подготовки)</i>
Направленность (профиль) подготовки:	Прикладная информатика (разработка программного обеспечения)
Квалификация выпускника: бакалавр	
Форма обучения: очная <i>(очная, заочная (с использованием электронного обучения и дистанционных образовательных технологий)*, очно-заочная (с использованием электронного обучения и дистанционных образовательных технологий)*)</i>	

Согласовано с УМК факультета бизнес-коммуникаций и информатики:

Рекомендовано кафедрой естественнонаучных дисциплин:

Протокол № 7 от «19» марта 2025 г.

Протокол № 2 от «02» марта 2026 г.

Председатель

М.Г. Синчурина

и.о. зав. кафедрой

А.Г. Балахчи

СОДЕРЖАНИЕ

	<i>стр.</i>
I. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ (МОДУЛЯ)	3
II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО	3
III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ	6
4.1 Содержание дисциплины, структурированное по темам, с указанием видов учебных занятий и СРС, отведенного на них количества академических часов	6
4.2 План внеаудиторной самостоятельной работы обучающихся по дисциплине	7
4.3 Содержание учебного материала	8
4.3.1. Перечень семинарских, практических занятий и лабораторных работ	8
4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение самостоятельной работы студентов	9
4.4. Методические указания по организации самостоятельной работы студентов	10
4.5. Примерная тематика курсовых работ (проектов)	14
V. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	14
а) основная литература	14
б) дополнительная литература	14
в) периодическая литература	14
г) базы данных, информационно-справочные и поисковые системы	14
VI. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	15
6.1. Учебно-лабораторное оборудование	15
6.2. Программное обеспечение	17
6.3. Технические и электронные средства	17
VII. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	17
VIII. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	19
8.1. Оценочные средства текущего контроля	19
8.2. Оценочные средства для промежуточной аттестации	21

I. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ (МОДУЛЯ)

Цели: Формирование у студентов системных знаний и практических навыков в области тестирования программного обеспечения на всех этапах жизненного цикла ПО, а также освоение современных методов, техник и инструментов обеспечения качества программных продуктов.

Задачи:

- изучить роль тестирования в жизненном цикле программного обеспечения;
- освоить основные виды, уровни и методы тестирования;
- сформировать навыки проектирования тестов и анализа дефектов;
- научиться применять инструменты ручного и автоматизированного тестирования.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) «Тестирование программного обеспечения» относится к части, формируемой участниками образовательных отношений «Блок 1. Дисциплины (модули)».

Дисциплина «Тестирование программного обеспечения» занимает важное место в образовательной программе подготовки инженера-программиста и направлена на формирование профессиональных компетенций в области обеспечения качества программных продуктов. Дисциплина ориентирована на системное понимание процессов проверки, анализа и сопровождения программного обеспечения в условиях промышленной разработки. Содержание дисциплины обеспечивает интеграцию знаний о программировании, проектировании и эксплуатации программных систем с практиками выявления и предотвращения дефектов, оценки надёжности и соответствия программного продукта заданным требованиям. Особое внимание уделяется роли тестирования как неотъемлемой части жизненного цикла программного обеспечения и как инструмента повышения устойчивости, безопасности и сопровождаемости программных решений. Освоение дисциплины способствует формированию инженерного мышления, ориентированного на качество, и развитию навыков аналитической оценки программных систем, подготовки тестовой документации и применения современных методов и средств тестирования. Дисциплина подготавливает обучающихся к участию в командной разработке программного обеспечения, взаимодействию с другими участниками процесса разработки и принятию технических решений с учётом требований к качеству и надёжности программных продуктов.

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

- Объектно-ориентированный анализ и программирование;
- Программирование;
- Культура разработки программного обеспечения с открытым исходным кодом.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной:

- Базы данных (продвинутый уровень);
- Технологии создания и отладки сценариев интерактивного контента;
- Автоматизация разработки и тестирования с помощью ИИ;
- Шаблоны проектирования.

III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов

следующих компетенций) в соответствии с ФГОС ВО и ОП ВО по данному направлению подготовки:

**Перечень планируемых результатов обучения по дисциплине (модулю),
соотнесенных с индикаторами достижения компетенций**

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-14 Способен руководить процессами разработки компьютерного программного обеспечения	ПК-14.1 Руководит разработкой программного кода	<ul style="list-style-type: none"> • Знает методы принятия управленческих решений. • Знает основные принципы и методы управления персоналом • Способен распределять задачи на разработку программного кода между исполнителями • Применяет лучшие мировые практики оформления программного кода • Оценивает качество и эффективность программного кода
	ПК-14.2 Руководит проверкой работоспособности компьютерного программного обеспечения	<ul style="list-style-type: none"> • ПК-14.2. З-3. Знает методы и средства проверки работоспособности компьютерного программного обеспечения • ПК-14.2. З-3. Знает методы и средства рефакторинга и оптимизации программного кода • ПК-14.2. У-1. Умеет распределять задачи на проверку работоспособности компьютерного программного обеспечения между исполнителями. • ПК-14.2. У-2. Умеет оценивать результаты проверки работоспособности компьютерного программного обеспечения • ПК-14.2. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности компьютерного программного обеспечения об исправлении ошибок, рефакторинге, оптимизации и инспекции кода • ПК-14.2. У-4. Умеет включать поиск возможных отказов из воздействий внутреннего или внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) в процедуры инспекции программного кода.
	ПК-14.3 Руководит интеграцией программных модулей и компонентов компьютерного программного обеспечения	<ul style="list-style-type: none"> • ПК-14.3. У-2. Умеет оценивать результаты выполнения назначенных заданий на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта • ПК-14.3. У-3. Умеет принимать управленческие решения по результатам проверки работоспособности выпусков программного продукта (решение о выпуске/невыпуске версии, отправка задач на доработку, добавление новых задач, передача на тестирование)

Компетенция	Индикаторы компетенций	Результаты обучения
<p>ПК-2 Способен проверять работоспособность и проводить рефакторинг кода программного обеспечения</p>	<p>ПК-2.1 Разрабатывает тестовые наборы данных для проверки работоспособности компьютерного программного обеспечения</p>	<ul style="list-style-type: none"> • ПК-2.1. З-1. Знает методы создания и документирования контрольных примеров и тестовых наборов данных. • ПК-2.1. З-2. Знает требования к структуре и форматам хранения тестовых наборов данных. • ПК-2.1. З-3. Знает правила, алгоритмы и технологии создания тестовых наборов данных. • ПК-2.1. У-1. Умеет разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения. • ПК-2.1. У-2. Умеет готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения.
	<p>ПК-2.2 Проверяет работоспособность компьютерного программного обеспечения</p>	<ul style="list-style-type: none"> • ПК-2.2. З-1. Знает методы и средства проверки работоспособности компьютерного программного обеспечения. • ПК-2.2. З-2. Знает государственные стандарты испытания автоматизированных систем. • ПК-2.2. З-3. Знает руководящие документы по стандартизации требований к документам автоматизированных систем. • ПК-2.2. У-1. Умеет применять методы и средства проверки работоспособности компьютерного программного обеспечения. • ПК-2.2. У-2. Умеет интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения • ПК-2.2. У-3. Умеет анализировать значения полученных характеристик компьютерного программного обеспечения.
	<p>ПК-2.3 Исправляет дефекты программного кода, зафиксированных в базе данных дефектов</p>	<ul style="list-style-type: none"> • ПК-2.3. З-1. Знает типичные ошибки, возникающие при разработке компьютерного программного обеспечения, методы их диагностики и исправления. • ПК-2.3. З-1. Знает методы и приемы отладки программного кода • ПК-2.3. У-1. Умеет воспроизводить дефекты программного кода, зафиксированных в базе данных дефектов. • ПК-2.3. У-2. Умеет выяснять причины возникновения дефектов программного кода. • ПК-2.3. У-3. Умеет вносить изменений в программный код для устранения выявленных дефектов.
	<p>ПК-2.4 Выполняет рефакторинг и инспекцию программного кода (код-ревью)</p>	<ul style="list-style-type: none"> • ПК-2.4. З-1. Знает методы и средства рефакторинга и инспекции программного кода. • ПК-2.4. З-2. Знает нормативно-технические документы (стандарты и регламенты), регламентирующие требования к программному коду, порядок отражения изменений в системе управления версиями, порядок отражения результатов рефакторинга, оптимизации и инспекции в коллективной базе знаний. • ПК-2.4. З-3. Знает о возможности отказа частей системы в результате воздействий внутреннего и внешнего нарушителя (хакер, неосторожный пользователь, программист, поставщик компонентов) • ПК-2.4. У-1. Умеет анализировать программный код на соответствие требованиям по читаемости и производительности. • ПК-2.4. У-2. Умеет проводить инспекцию программного кода для поиска не обнаруженных на ранних стадиях разработки компьютерного программного обеспечения ошибок и критических мест. • ПК-2.4. У-3. Умеет применять методы и средства рефакторинга и инспекции программного кода. • ПК-2.4. У-4. Умеет публиковать результаты рефакторинга и инспекции в коллективной базе знаний.

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-6 Способен оптимизировать производительность программного обеспечения	ПК-6.1 Выполняет мониторинг производительности программного обеспечения	<ul style="list-style-type: none"> • ПК-6.1. З-1. Знает методы и средства мониторинга производительности компьютерного программного обеспечения. • ПК-6.1. З-2. Знает метрики производительности программного обеспечения. • ПК-6.1. З-3. Знает современные инструменты мониторинга производительности программного обеспечения. • ПК-6.1. У-1. Умеет применять методы и средства мониторинга производительности компьютерного программного обеспечения. • ПК-6.1. У-2. Умеет интерпретировать диагностические данные мониторинга производительности компьютерного программного обеспечения. • ПК-6.1. У-3. Умеет определять "узкие места" программного обеспечения
	ПК-6.2 Выполняет оптимизацию программного кода	<ul style="list-style-type: none"> • ПК-6.2. З-1. Знает методы и средства оптимизации производительности компьютерного программного обеспечения. • ПК-6.2. З-2. Знает современные инструменты оптимизации производительности программного обеспечения. • ПК-6.2. У-1. Умеет оптимизировать программный код с использованием специализированных программных средств. • ПК-6.2. У-2. Умеет вырабатывать варианты оптимизации производительности компьютерного программного обеспечения. • ПК-6.2. У-3. Умеет проводить оценку и обоснование рекомендуемых решений по оптимизации производительности.
ПК-8 Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода	ПК-8.1 Применяет ИИ-инструменты для генерации программного кода	<ul style="list-style-type: none"> • Знает принципы работы современных генеративных ИИ-моделей для генерации кода. • Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование)
	ПК-8.2 Использует ИИ для анализа и отладки кода	<ul style="list-style-type: none"> • ПК-8.2. З-1. Знает методы ИИ-анализа кода. • ПК-8.2. З-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ. • ПК-8.2. У-1. Умеет настраивать ИИ-инструменты для поиска уязвимостей. • ПК-8.2. У-2. Умеет интерпретировать рекомендации ИИ по исправлению кода.
	ПК-8.3 Оптимизирует код с помощью ИИ	<ul style="list-style-type: none"> • ПК-8.3. З-1. Знает методы ИИ-оптимизации. • ПК-8.3. З-2. Знает критерии качества кода, применяемые ИИ-системами. • ПК-8.3. У-1. Умеет использовать ИИ для рефакторинга. • ПК-8.3. У-2. Умеет проверять корректность оптимизаций, предложенных ИИ.
	ПК-8.4 Оценивает этические и профессиональные аспекты применения ИИ в разработке	<ul style="list-style-type: none"> • Знает этические нормы использования ИИ (конфиденциальность, плагиат кода и т.п.). • Умеет проверять код на соответствие стандартам после ИИ-генерации.

IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов, в том числе 8 часов на контроль.

Из них реализуется с использованием электронного обучения и дистанционных образовательных технологий 54 часа контактной работы и 46 часов самостоятельной работы.

Форма промежуточной аттестации: зачет.

4.1 Содержание дисциплины, структурированное по темам, с указанием видов учебных занятий и СРС, отведенного на них количества академических часов

п/п	Раздел дисциплины/темы	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости; Форма промежуточной аттестации (по семестрам)
			Контактная работа преподавателя с обучающимися			Консультации		
			Лекции (из них электронные часы)	Семинарские (практические) занятия (из них электронные часы)				
<i>Теоретические основы тестирования программного обеспечения</i>			8 (8)	18 (18)	0	24		
1	Инструменты и автоматизация тестирования	3	4 (4)	9 (9)	0	12		
2	Виды и уровни тестирования	3	4 (4)	9 (9)	0	12		
<i>Методы и инструменты тестирования</i>			10 (10)	18 (18)	0	22		
3	Тест-дизайн и управление дефектами	3	4 (4)	9 (9)	0	11		
4	Инструменты и автоматизация тестирования	3	6 (6)	9 (9)	0	11		
Итого за 3 семестр			18 (18)	36 (36)	0	46	Зач (8)	
Итого часов			18 (18)	36 (36)	0	46		

4.2 План внеаудиторной самостоятельной работы обучающихся по дисциплине

Се- местр	Название раздела, темы	Самостоятельная работа обучающихся			Оце- ночное сред- ство	Учебно- методи- ческое обеспе- чение само- стоя- тельной работы
		Вид самостоятельной работы	Сроки выпол- нения	Зат- раты вре- мени, час. (из них с при- мене- нием ДОТ)		
3	Инструменты и автоматизация тестирования	Для формирования умений: подготовка проекта или творческой работы	4 недели	12 (12)	Пз	ЭОС Forlabs, ОЛ, ДЛ.
3	Виды и уровни тестирования	Для формирования умений: подготовка проекта или творческой работы	4 недели	12 (12)	Тест, Пз	ЭОС Forlabs, ОЛ, ДЛ.
3	Тест-дизайн и управление дефектами	Для формирования умений: подготовка проекта или творческой работы	5 недель	11 (11)	Пз	ЭОС Forlabs, ОЛ, ДЛ.
3	Инструменты и автоматизация тестирования	Для формирования умений: подготовка проекта или творческой работы	5 недель	11 (11)	Тест, РЗ	ЭОС Forlabs, ОЛ, ДЛ.
Общая трудоемкость самостоятельной работы по дисциплине (час)				46		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				46		
Бюджет времени самостоятельной работы, предусмотренный учебным планом для данной дисциплины (час)				46		

4.3 Содержание учебного материала

Трудоемкость дисциплины (з.е.)	3
Наименование основных разделов (модулей)	Теоретические основы тестирования программного обеспечения Методы и инструменты тестирования
Формы текущего контроля	Практическое задание, тест, решение задач
Форма промежуточной аттестации	Зачет

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/п	№ раздела и темы дисциплины (модуля)	Наименование семинаров, практических и лабораторных работ	Трудоемкость, час. (из них электронные часы)	Оценочные средства	Формируемые компетенции
1	1	Введение в тестирование программного обеспечения	9 (9)	Пз	ПК-2.1, ПК-2.2, ПК-2.3, ПК-2.4
2	2	Виды и уровни тестирования программного обеспечения	9 (9)	Тест, Пз	ПК-2.2, ПК-2.1, ПК-2.3, ПК-6.1, ПК-6.2, ПК-8.2, ПК-2.4, ПК-8.3, ПК-14.2, ПК-14.3
3	3	Тест-дизайн и управление дефектами	9 (9)	Пз	ПК-2.1, ПК-2.2, ПК-2.3, ПК-6.1, ПК-6.2, ПК-2.4, ПК-14.2, ПК-14.3
4	4	Тестирование и автоматизация для веб-сервиса To-Do List API	9 (9)	Тест, Пз	ПК-2.1, ПК-2.2, ПК-2.3, ПК-2.4, ПК-6.1, ПК-6.2, ПК-8.2, ПК-8.3, ПК-14.2, ПК-14.3

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение самостоятельной работы студентов

№ п/п	Тема	Задание	Формируемая компетенция	ИДК
1	Инструменты и автоматизация тестирования	Подготовка рабочего места	ПК-2, ПК-6, ПК-8, ПК-14	ПК-2.1 ПК-2.2 ПК-2.3 ПК-6.1 ПК-6.2 ПК-8.2 ПК-2.4 ПК-8.3 ПК-14.2 ПК-14.3

№ п/п	Тема	Задание	Формируемая компетенция	ИДК
2	Виды и уровни тестирования	Проектирование и реализация тестирования модуля сервиса EnergyMonitor	ПК-2, ПК-6, ПК-8, ПК-14	ПК-2.1 ПК-2.2 ПК-2.3 ПК-6.1 ПК-6.2 ПК-8.2 ПК-2.4 ПК-8.3 ПК-14.2 ПК-14.3
3	Тест-дизайн и управление дефектами	Проектирование тест-дизайна и управление дефектами	ПК-2, ПК-6, ПК-14	ПК-2.1 ПК-2.2 ПК-2.3 ПК-6.1 ПК-6.2 ПК-2.4 ПК-14.2 ПК-14.3
4	Инструменты и автоматизация тестирования	Инструменты и автоматизация тестирования	ПК-2, ПК-6, ПК-8, ПК-14	ПК-2.1 ПК-2.2 ПК-2.3 ПК-6.1 ПК-6.2 ПК-8.2 ПК-2.4 ПК-8.3 ПК-14.2

4.4. Методические указания по организации самостоятельной работы студентов

Самостоятельная работа студентов всех форм и видов обучения является одним из обязательных видов образовательной деятельности, обеспечивающей реализацию требований Федеральных государственных стандартов высшего профессионального образования. Согласно требованиям нормативных документов самостоятельная работа студентов является обязательным компонентом образовательного процесса, так как она обеспечивает закрепление получаемых на лекционных занятиях знаний путем приобретения навыков осмысления и расширения их содержания, навыков решения актуальных проблем формирования общекультурных и профессиональных компетенций, научно-исследовательской деятельности, подготовки к семинарам, лабораторным работам, сдаче зачетов и экзаменов. Самостоятельная работа студентов представляет собой совокупность аудиторных и внеаудиторных занятий и работ. Самостоятельная работа в рамках образовательного процесса в вузе решает следующие задачи:

— закрепление и расширение знаний, умений, полученных студентами во время аудиторных и внеаудиторных занятий, превращение их в стереотипы умственной и физической деятельности;

- приобретение дополнительных знаний и навыков по дисциплинам учебного плана;
- формирование и развитие знаний и навыков, связанных с научно-исследовательской деятельностью;
- развитие ориентации и установки на качественное освоение образовательной программы;
- развитие навыков самоорганизации;
- формирование самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации;
- выработка навыков эффективной самостоятельной профессиональной теоретической, практической и учебно-исследовательской деятельности.

Подготовка к лекции. Качество освоения содержания конкретной дисциплины прямо зависит от того, насколько студент сам, без внешнего принуждения формирует у себя установку на получение на лекциях новых знаний, дополняющих уже имеющиеся по данной дисциплине. Время на подготовку студентов к двухчасовой лекции по нормативам составляет не менее 0,2 часа.

Подготовка к практическому занятию. Подготовка к практическому занятию включает следующие элементы самостоятельной деятельности: четкое представление цели и задач его проведения; выделение навыков умственной, аналитической, научной деятельности, которые станут результатом предстоящей работы. Выработка навыков осуществляется с помощью получения новой информации об изучаемых процессах и с помощью знания о том, в какой степени в данное время студент владеет методами исследовательской деятельности, которыми он станет пользоваться на практическом занятии. Подготовка к практическому занятию нередко требует подбора материала, данных и специальных источников, с которыми предстоит учебная работа. Студенты должны дома подготовить к занятию 3–4 примера формулировки темы исследования, представленного в монографиях, научных статьях, отчетах. Затем они самостоятельно осуществляют поиск соответствующих источников, определяют актуальность конкретного исследования процессов и явлений, выделяют основные способы доказательства авторами научных работ ценности того, чем они занимаются. В ходе самого практического занятия студенты сначала представляют найденные ими варианты формулировки актуальности исследования, обсуждают их и обосновывают свое мнение о наилучшем варианте. Время на подготовку к практическому занятию по нормативам составляет не менее 0,2 часа.

Подготовка к контрольной работе. Контрольная работа назначается после изучения определенного раздела (разделов) дисциплины и представляет собой совокупность развернутых письменных ответов студентов на вопросы, которые они заранее получают от преподавателя. Самостоятельная подготовка к контрольной работе включает в себя: — изучение конспектов лекций, раскрывающих материал, знание которого проверяется контрольной работой; повторение учебного материала, полученного при подготовке к семинарским, практическим занятиям и во время их проведения; изучение дополнительной литературы, в которой конкретизируется содержание проверяемых знаний; составление в мысленной форме ответов на поставленные в контрольной работе вопросы; формирование психологической установки на успешное выполнение всех заданий. Время на подготовку к контрольной работе по нормативам составляет 2 часа.

Подготовка к экзамену. Самостоятельная подготовка к экзамену схожа с подготовкой к зачету, особенно если он дифференцированный. Но объем учебного материала, который нужно восстановить в памяти к экзамену, вновь осмыслить и понять,

значительно больше, поэтому требуется больше времени и умственных усилий. Важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных научных трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их противоречивость, знание имен ученых, изучавших обсуждаемую проблему. Необходимо также привести информацию о материалах эмпирических исследований, что указывает на всестороннюю подготовку студента к экзамену. Время на подготовку к экзамену по нормативам составляет 36 часов для бакалавров.

Формы внеаудиторной самостоятельной работы

Составление глоссария Цель самостоятельной работы: повысить уровень информационный культуры; приобрести новые знания; отработать необходимые навыки в предметной области учебного курса. Глоссарий — словарь специализированных терминов и их определений. Статья глоссария — определение термина. Содержание задания: сбор и систематизация понятий или терминов, объединенных общей специфической тематикой, по одному либо нескольким источникам. Выполнение задания: 1) внимательно прочесть работу; 2) определить наиболее часто встречающиеся термины; 3) составить список терминов, объединенных общей тематикой; 4) расположить термины в алфавитном порядке; 5) составить статьи глоссария: — дать точную формулировку термина в именительном падеже; — объемно раскрыть смысл данного термина. Планируемые результаты самостоятельной работы: способность студентов решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

Разработка проекта (индивидуального, группового) Цель самостоятельной работы: развитие способности прогнозировать, проектировать, моделировать. Проект — «ограниченное во времени целенаправленное изменение отдельной системы с установленными требованиями к качеству результатов, возможными рамками расхода средств и ресурсов и специфической организацией». Выполнение задания: 1) диагностика ситуации (проблематизация, целеполагание, конкретизация цели, форматирование проекта); 2) проектирование (уточнение цели, функций, задач и плана работы; теоретическое моделирование методов и средств решения задач; детальная проработка этапов решения конкретных задач; пошаговое выполнение запланированных проектных действий; систематизация и обобщение полученных результатов, конструирование предполагаемого результата, пошаговое выполнение проектных действий); 3) рефлексия (выяснение соответствия полученного результата замыслу; определение качества полученного продукта; перспективы его развития и использования). Предполагаемые результаты самостоятельной работы: готовность студентов использовать знание современных проблем науки и образования при решении образовательных и профессиональных задач; готовность использовать индивидуальные креативные способности для оригинального решения исследовательских задач; — способность прогнозировать, проектировать, моделировать.

Информационный поиск Цель самостоятельной работы: развитие способности к проектированию и преобразованию учебных действий на основе различных видов информационного поиска. Информационный поиск — поиск неструктурированной документальной информации. Список современных задач информационного поиска: решение вопросов моделирования; классификация документов; фильтрация, классификация документов; проектирование архитектур поисковых систем и пользовательских интерфейсов; извлечение информации (аннотирование и реферирование документов); выбор

информационно-поискового языка запроса в поисковых системах. Содержание задания по видам поиска: поиск библиографический — поиск необходимых сведений об источнике и установление его наличия в системе других источников. Ведется путем разыскания библиографической информации и библиографических пособий (информационных изданий); поиск самих информационных источников (документов и изданий), в которых есть или может содержаться нужная информация; — поиск фактических сведений, содержащихся в литературе, книге (например, об исторических фактах и событиях, о биографических данных из жизни и деятельности писателя, ученого и т. п.). Выполнение задания:

- 1) определение области знаний;
- 2) выбор типа и источников данных;
- 3) сбор материалов, необходимых для наполнения информационной модели;
- 4) отбор наиболее полезной информации;
- 5) выбор метода обработки информации (классификация, кластеризация, регрессионный анализ и т.д.);
- 6) выбор алгоритма поиска закономерностей;
- 7) поиск закономерностей, формальных правил и структурных связей в собранной информации;
- 8) творческая интерпретация полученных результатов.

Планируемые результаты самостоятельной работы: — способность студентов решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; готовность использовать знание современных проблем науки и образования при решении образовательных и профессиональных задач.

Разработка мультимедийной презентации Цели самостоятельной работы (варианты): — освоение (закрепление, обобщение, систематизация) учебного материала; — обеспечение контроля качества знаний; — формирование специальных компетенций, обеспечивающих возможность работы с информационными технологиями; — становление общекультурных компетенций. Мультимедийная презентация — представление содержания учебного материала, учебной задачи с использованием мультимедийных технологий.

Выполнение задания:

1. Этап проектирования: — определение целей использования презентации; — сбор необходимого материала (тексты, рисунки, схемы и др.); — формирование структуры и логики подачи материала; — создание папки, в которую помещен собранный материал.

2. Этап конструирования: — выбор программы MS PowerPoint в меню компьютера; — определение дизайна слайдов; — наполнение слайдов собранной текстовой и наглядной информацией; — включение эффектов анимации и музыкального сопровождения (при необходимости); — установка режима показа слайдов (титольный слайд, включающий наименование кафедры, где выполнена работа, название презентации, город и год; содержательный — список слайдов презентации, сгруппированных по темам сообщения; заключительный слайд содержит выводы, пожелания, список литературы и пр.).

3. Этап моделирования — проверка и коррекция подготовленного материала, определение продолжительности его демонстрации.

Планируемые результаты самостоятельной работы: — повышение информационной культуры студентов и обеспечение их готовности к интеграции в современное

информационное пространство; — способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; — способность к критическому восприятию, обобщению, анализу профессиональной информации, постановке цели и выбору путей ее достижения; — способность применять современные методики и технологии организации и реализации образовательного процесса на различных образовательных ступенях в различных образовательных учреждениях; — готовность использовать индивидуальные креативные способности для оригинального решения исследовательских задач.

В ФБГОУ ВО «ИГУ» организация самостоятельной работы студентов регламентируется Положением о самостоятельной работе студентов, принятым Ученым советом ИГУ 22 июня 2012 г.

4.5. Примерная тематика курсовых работ (проектов)

По данной дисциплине выполнение курсовых проектов (работ) не предусматривается.

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

а) основная литература

1. Игнатъев А. В. Тестирование программного обеспечения : учебное пособие для вузов / А. В. Игнатъев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2025. — 56 с. — ISBN 978-5-507-50858-7

2. Старолетов, С. М. Основы тестирования и верификации программного обеспечения [Электронный ресурс] : учебное пособие / С. М. Старолетов. - Электрон. текстовые дан. - Санкт-Петербург : Лань, 2018. - 344 с. - ЭБС "Лань". - неогранич. доступ. - ISBN 978-5-8114-3041-3 : Б. ц.

3. Зубкова, Т. М. Технология разработки программного обеспечения [Электронный ресурс] : учебное пособие / Т. М. Зубкова. - Электрон. текстовые дан. - Санкт-Петербург : Лань, 2019. - 324 с. - ЭБС "Лань". - Неогранич. доступ. - ISBN 978-5-8114-3842-6 : Б. ц.

б) дополнительная литература

1. Старолетов, С. М. Основы тестирования и верификации программного обеспечения [Электронный ресурс] / С. М. Старолетов. - 2-е изд., стер. - Электрон. текстовые дан. - Санкт-Петербург : Лань, 2020. - 344 с. - ЭБС "Лань". - неогранич. доступ. - ISBN 978-5-8114-5239-2 : Б. ц.

2. Бек, Кент. Экстремальное программирование. Разработка через тестирование [Текст] : науч. изд. / К. Бек. - СПб. : Питер, 2020. - 221 с. ; 23 см. - (Библиотека программиста). - Пер. изд. : Test-driven development by example / Kent Beck. - Reading (Ma). - ISBN 978-5-4461-1439-9 : 841.20 р.

в) периодическая литература

Нет.

г) базы данных, информационно-справочные и поисковые системы

Нет.

В соответствии с п. 4.3.4. ФГОС ВО, обучающимся в течение всего периода обучения обеспечен неограниченный доступ (удаленный доступ) к электронно-библиотечным системам:

— Открытая электронная база ресурсов и исследований «Университетская

информационная система РОССИЯ» [Электронный ресурс] : сайт. – Режим доступа: <http://uisrussia.msu.ru> бессрочный

— Государственная информационная система «Национальная электронная библиотека» [Электронный ресурс] : сайт. – Режим доступа: <http://нэб.рф>. бессрочный

— Научная электронная библиотека «ELIBRARY.RU» [Электронный ресурс] : сайт. - Контракт № 148 от 23.12.2020 г. Акт от 24.12.2020 г. Срок действия по 31.12.2022 г. – Режим доступа: <http://elibrary.ru/>

— ЭБС «Издательство Лань». Контракт № 04-Е-0346 от 12.11.2021 г. № 976 от 14.11.2021 г. Срок действия по 13.11.2022 г. – Режим доступа: <https://www.e.lanbook.com>

— ЭБС ЭЧЗ «Библиотех». Государственный контракт № 019 от 22.02.2011 г. ООО «Библиотех». Лицензионное соглашение к Государственному контракту № 019 от 22.02.2011. Срок действия: бессрочный. – Режим доступа: <https://isu.bibliotech.ru/>

— ЭБС «Рукопт» ЦКБ «Бибком». № 04-Е-0343 от 12.11.2021 г. Акт № 6К-5195 от 14.11.2021 г. Срок действия по 13.11.2022г. – Режим доступа: <http://rucont.ru>

— ЭБС «Айбукс.ру/ibooks.ru» ООО «Айбукс». Контракт № 04-Е-0344 от 12.11.2021 г.; Акт от 14.11.2021 г. Срок действия по 13.11.2022 г. – Режим доступа: <http://ibooks.ru>

— Электронно-библиотечная система «ЭБС Юрайт». ООО «Электронное издательство Юрайт». Контракт № 04-Е-0258 от 20.09.2021г. Контракт № 04-Е-0258 от 20.09.2021 г. Срок действия по 17.10. 2022 г. – Режим доступа: <https://urait.ru>

— УБД ИВИС. Контракт № 04-Е-0347 от 12.11.2021 г. Акт от 15.11.2021 г. Срок действия с 01.01.2022 по 31.12.2022 г. – Режим доступа: <http://dlib.eastview.com>

— Электронная библиотека ИД Гребенников. Контракт № 04-Е-0348 от 12.11.2021г.; Акт № 348 от 15.11.2021 г. Срок действия с 01.01.2022 по 31.12.2022 – Режим доступа: <http://grebennikon.ru>

VI. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-лабораторное оборудование

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
---	---	--

<p>Специальные помещения: Учебная аудитория для проведения занятий лекционного и семинарского типа, текущего контроля, промежуточной аттестации.</p>	<p>Аудитория оборудована специализированной учебной мебелью, техническими средствами обучения, служащими для представления информации большой аудитории:</p> <p>Ноутбук(AserAspirev3-5516 (AMDA10-4600M 2300 МГц)) (1 штука) с неограниченным доступом к сети Интернет; Проектор Vivitek, экран ScreenVtdiaEcot- 3200*200MW 1:1, колонки, наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие рабочей программе дисциплины «Архитектурный подход к развитию предприятий и информационных систем».</p> <p>Учебная лаборатория: компьютеры для проведения практических работ (Системный блок AMDAthlon-64 X3 445 3100 МГц), Монитор LG F1742S (2 штуки), Монитор ViewSonic VA703b(24 штуки) с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации; проектор Sony XGA VPLSX535, экран ScreenVtdiaEcot- 3200*200MW 1:1</p>	<p>ОС Windows: DreamSpark Premium, Договор № 03-016-14 от 30.10.2014</p> <p>Microsoft Office: 0365ProPiusOpenStudents ShrdSvr ALNG subs VL NL I MthAcadmsStdnt w/Faculty (15000 лицензий)</p> <p>Kaspersky Endpoint Security длябизнеса- стандартный Russian Edition. 15002499 Node 1 year Educational License № 1B08-170221-054045-730-177</p> <p>BusinessStudio Лицензия № 7464 (бессрочно)</p>
--	---	---

Специальные помещения: компьютерный класс (учебная аудитория) для групповых и индивидуальных консультаций, курсового проектирования (выполнения курсовых работ), организации самостоятельной работы, в том числе, научно-исследовательской	Аудитория оборудована специализированной учебной мебелью, техническими средствами обучения: компьютеры (системный блок AMD Athlon 64 X2 DualCore 3600+ 1900 МГц (15 штук), Монитор LGFlatron L1742SE (14 штук), Монитор ViewSonic VG720) с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.	ОС Windows: DreamSpark Premium, Договор № 03-016-14 от 30.10.2014 Microsoft Office: 0365ProPiusOpenStudents ShrdSvr ALNG subs VL NL I MthAcddsStdnt w/Faculty (15000 лицензий) Kaspersky Endpoint Security для бизнеса- стандартный Russian Edition. 15002499 Node 1 year Educational License № 1B08-170221-054045-730-177
--	---	--

6.2. Программное обеспечение

№	Наименование Программного продукта	Кол-во	Обоснование для пользования ПО	Дата выдачи лицензии	Срок действия права пользования
1	UbuntuLinux 16.04.1	Условия правообладателя	Условия использования по ссылке: https://www.ubuntu.com/legal/terms-and-policies/terms	Условия правообладателя	Условия правообладателя
2	Python 3.x	50	GNU General Public License (GPL)	2024	Условия правообладателя
3	PyTest	50	это фреймворк для автоматизированного тестирования на языке Python, широко используемый для модульного, интеграционного и функционального тестирования программного обеспечения. PyTest распространяется по лицензии MIT License — свободной лицензии с открытым исходным кодом, допускающей свободное использование, модификацию и распространение программного обеспечения, включая коммерческое применение, при условии сохранения уведомления об авторских правах и текста лицензии.	01.01.2025	Условия правообладателя
4	Распределенная система управления версиями Git (распространяется бесплатно, лицензия GNU GPL 2)	Условия правообладателя	распространяется бесплатно, лицензия GNU GPL 2	Условия правообладателя	Условия правообладателя
5	Docker	Условия правообладателя	Условия правообладателя	Условия правообладателя	Условия правообладателя

6.3. Технические и электронные средства

Методической системой преподавания предусмотрено использование технических и электронных средств обучения и контроля знаний студентов: мультимедийные презентации, фрагменты фильмов.

VII. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации программы данной дисциплины используются различные образовательные технологии, в том числе электронное обучение и дистанционные образовательные технологии.

Проблемное обучение	Создание в учебной деятельности проблемных ситуаций и организация активной самостоятельной деятельности учащихся по их разрешению, в результате чего происходит творческое овладение знаниями, умениями, навыками, развиваются мыслительные способности
Разноуровневое обучение	У преподавателя появляется возможность помогать слабому, уделять внимание сильному, реализуется желание сильных учащихся быстрее и глубже продвигаться в образовании. Сильные учащиеся утверждают в своих способностях, слабые получают возможность испытывать учебный успех, повышается уровень мотивации ученья.
Проектные методы обучения	Работа по данной методике дает возможность развивать индивидуальные творческие способности учащихся, более осознанно подходить к профессиональному и социальному самоопределению
Исследовательские методы в обучении	Дает возможность учащимся самостоятельно пополнять свои знания, глубоко вникать в изучаемую проблему и предполагать пути ее решения, что важно при формировании мировоззрения. Это важно для определения индивидуальной траектории развития каждого обучающегося
Лекционно-семинарскозачетная система	Данная система дает возможность сконцентрировать материал в блоки и преподносить его как единое целое, а контроль проводить по предварительной подготовке обучающихся
Информационно-коммуникационные технологии	Изменение и неограниченное обогащение содержания образования, использование интегрированных курсов, доступ в ИНТЕРНЕТ.

Наименование тем занятий с использованием активных форм обучения:

№	Тема занятия	Вид занятия	Форма / Методы интерактивного обучения	Кол-во часов (из них электронные часы)
1	Инструменты и автоматизация тестирования	Практика	Проектная работа	4
2	Виды и уровни тестирования	Практика	Проектная работа	4
3	Тест-дизайн и управление дефектами	Практика	Проектная работа	4

№	Тема занятия	Вид занятия	Форма / Методы интерактивного обучения	Кол-во часов (из них электронные часы)
4	Тест-дизайн и управление дефектами	Практика	Контекст	2
5	Инструменты и автоматизация тестирования	Лекция	конференция (доклады)	2
6	Инструменты и автоматизация тестирования	Практика	Проектная работа	4
7	Инструменты и автоматизация тестирования	Практика	Симуляция контекста во время собеседования	2

VIII. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Оценочные средства текущего контроля

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Практическое задание	Инструменты и автоматизация тестирования. Виды и уровни тестирования. Тест-дизайн и управление дефектами.	ПК-2.1, ПК-2.2, ПК-2.3, ПК-2.4, ПК-6.1, ПК-6.2, ПК-8.2, ПК-8.3, ПК-14.2, ПК-14.3
2	Тест	Виды и уровни тестирования. Инструменты и автоматизация тестирования.	ПК-2.2, ПК-2.1, ПК-2.3, ПК-2.4, ПК-6.1, ПК-6.2, ПК-8.2, ПК-8.3, ПК-14.2, ПК-14.3
3	Решение задач	Инструменты и автоматизация тестирования.	ПК-2.1, ПК-2.2, ПК-2.3, ПК-6.1, ПК-6.2, ПК-8.2, ПК-2.4, ПК-8.3, ПК-14.2

Примеры оценочных средств для текущего контроля

Демонстрационный вариант теста

1. Задание с единственным выбором. Выберите один правильный ответ.

Какой набор входных данных наиболее корректно использовать для проверки функции, вычисляющей среднее значение массива чисел?

- a. Только массив положительных чисел
- b. Пустой массив
- c. Массив с одинаковыми значениями
- d. Набор данных, включающий нормальные, граничные и некорректные значения

2. Задание с единственным выбором. Выберите один правильный ответ.

Какое действие относится непосредственно к проверке работоспособности

программного обеспечения?

- a. Оптимизация алгоритма
- b. Запуск программы с тестовыми данными и анализ фактического результата
- c. Рефакторинг кода
- d. Написание технического задания

3. *Задание с единственным выбором. Выберите один правильный ответ.*

Какой порядок действий является наиболее корректным при исправлении дефекта, зафиксированного в системе баг-трекинга?

- a. Закрыть баг → исправить код → проверить
- b. Исправить код → закрыть баг → написать тест
- c. Исправить код без воспроизведения дефекта
- d. Проанализировать баг → воспроизвести → исправить → проверить → закрыть

4. *Задание с единственным выбором. Выберите один правильный ответ.*

Какое из перечисленных действий относится к рефакторингу, а не к исправлению дефекта?

- a. Исправление деления на ноль
- b. Переименование переменных для улучшения читаемости без изменения логики
- c. Добавление проверки входных данных
- d. Изменение условия ветвления для корректной работы

5. *Задание с единственным выбором. Выберите один правильный ответ.*

Какой показатель наиболее непосредственно используется при мониторинге производительности программного обеспечения?

- a. Количество комментариев
- b. Стиль оформления кода
- c. Количество строк кода
- d. Время отклика и использование ресурсов

6. *Задание с единственным выбором. Выберите один правильный ответ.*

Какое действие является примером оптимизации программного кода?

- a. Замена линейного поиска на хеш-таблицу
- b. Добавление комментариев
- c. Добавление логирования
- d. Переименование функций

7. *Задание с единственным выбором. Выберите один правильный ответ.*

В каком случае использование ИИ наиболее оправдано для анализа и отладки кода?

- a. Для поиска синтаксических ошибок компилятором
- b. Для запуска программы
- c. Для форматирования кода
- d. Для анализа стектрейса и предложения возможной причины ошибки

8. *Задание с единственным выбором. Выберите один правильный ответ.*

Какой результат свидетельствует об эффективном использовании ИИ для оптимизации кода?

- a. Снижено время выполнения алгоритма при сохранении корректности
- b. Код стал короче, но работает медленнее
- c. ИИ переписал код без объяснений
- d. Добавлены комментарии

9. Задание с единственным выбором. Выберите один правильный ответ.

Какое действие относится к руководству проверкой работоспособности программного обеспечения, а не к индивидуальному тестированию?

- a. Запуск модульного теста
- b. Исправление одного дефекта
- c. Написание одного тест-кейса
- d. Планирование этапов тестирования и распределение задач в команде

10. Задание с единственным выбором. Выберите один правильный ответ.

Какое решение должен принять руководитель при интеграции нескольких программных модулей?

- a. Назначить интеграционное тестирование и определить порядок объединения модулей
- b. Передать интеграцию одному разработчику без проверки
- c. Проверить каждый модуль только отдельно
- d. Отложить тестирование до релиза

8.2. Оценочные средства для промежуточной аттестации

Вопросы к зачету:

1. Теоретический вопрос: опишите жизненный цикл дефекта (bug lifecycle) в системе отслеживания. Какие статусы он проходит? Что должно содержаться в хорошем отчете об ошибке? Практическое задание В проекте в Issue Tracker заведен дефект #CONV-101: «При конвертации очень больших сумм (например, $1e30$) возвращается некорректный результат (Infinity или ошибка), а не сообщение об ограничении». Воспроизведение: Напишите простой скрипт или используйте Postman, чтобы воспроизвести ошибку. Зафиксируйте факт ошибки (скриншот/логи). Исправление: Найдите в коде сервиса место, отвечающее за проверку входных данных. Исправьте дефект, добавив валидацию максимальной суммы (например, не более $1e12$). Исправление оформите через git: создайте ветку fix/CONV-101, коммит, а затем без создания Merge Request. Тест на исправление: Дополните или создайте модульный тест, который проверяет новое поведение (успешная конвертация до лимита, ошибка — сверх лимита).

2. Теоретический вопрос: что такое «запах кода» (code smell)? Приведите три примера «запахов», относящихся к методам/функциям, и по одному возможному способу рефакторинга для каждого. Практическое задание Вам предоставлен файл CurrencyService.java (или .py) с методом convert, содержащим «запахи» кода. Инспекция: запустите статический анализатор (SonarLint в IDE или через CLI) на этом файле. Сохраните топ-3 наиболее критических замечания (не связанных с форматированием). Код-ревью: в качестве ревьюера изучите предоставленный фрагмент кода (другого студента или заранее подготовленный). В формате комментариев укажите две конкретные проблемы: одну — касающуюся потенциальной ошибки (например, отсутствие обработки null), вторую — касающуюся читаемости/архитектуры (например, слишком длинный метод). Рефакторинг: выполните рефакторинг метода convert, устранив один выявленный «запах» (например,

«Длинный метод» через выделение метода валидации). Код после рефакторинга должен проходить все существующие тесты.

3. Теоретический вопрос: опишите стратегию создания тестовых данных (Test Data Design). Какие основные категории/классы эквивалентности данных вы выделите для метода API, который конвертирует сумму из одной валюты в другую? Объясните на примерах. Практическое задание Вам дан эндпоинт POST /api/convert. Параметры: amount (число), from_currency (строка, код ISO), to_currency (строка, код ISO). Создайте структурированный файл test_data.json, содержащий массив JSON-объектов для тестирования этого эндпоинта. Включите не менее 10 тестовых случаев, покрывающих: Валидные данные (положительные, отрицательные, нулевая сумма). Невалидные данные (несуществующие коды валют, некорректный тип amount, отсутствующие поля). Граничные/особые значения (очень большое число, валюта из трех букв, одинаковые from и to). Кратко обоснуйте (в комментарии к файлу) выбор каждого тестового случая.

4. Теоретический вопрос: в чем разница между модульным, интеграционным и сквозным (E2E) тестированием? Какой тип тестирования наиболее релевантен для проверки REST API и почему? Практическое задание (ПК-2.2): Используя готовый набор данных из Пакета 1 или создав свой, выполните: Автоматизация в Postman: Создайте коллекцию Postman для эндпоинта /api/convert. Настройте автоматические проверки (assertions) в тестах Tab для: Статус-кода ответа. Структуры JSON-ответа (наличие полей result, rate). Корректности вычисления ($result \approx amount * rate$ для успешных случаев). Интеграционный тест: Напишите один интеграционный тест на Python (pytest + requests) или Java (JUnit + RestAssured), который: Проверяет успешную конвертацию. Проверяет обработку ошибки при несуществующей валюте. Запустите тест и предоставьте отчет о прохождении.

5. Теоретический вопрос: опишите жизненный цикл дефекта (bug lifecycle) в системе отслеживания. Какие статусы он проходит? Что должно содержаться в хорошем отчете об ошибке? Практическое задание В проекте в Issue Tracker заведен дефект #CONV-101: «При конвертации очень больших сумм (например, $1e30$) возвращается некорректный результат (Infinity или ошибка), а не сообщение об ограничении». Воспроизведение: Напишите простой скрипт или используйте Postman, чтобы воспроизвести ошибку. Зафиксируйте факт ошибки (скриншот/логи). Исправление: Найдите в коде сервиса место, отвечающее за проверку входных данных. Исправьте дефект, добавив валидацию максимальной суммы (например, не более $1e12$). Исправление оформите через git: создайте ветку fix/CONV-101, коммит, а затем без создания Merge Request. Тест на исправление: Дополните или создайте модульный тест, который проверяет новое поведение (успешная конвертация до лимита, ошибка — сверх лимита).

6. Теоретический вопрос: что такое «запах кода» (code smell)? Приведите три примера «запахов», относящихся к методам/функциям, и по одному возможному способу рефакторинга для каждого. Практическое задание Вам предоставлен файл CurrencyService.java (или .py) с методом convert, содержащим «запахи» кода. Инспекция: запустите статический анализатор (SonarLint в IDE или через CLI) на этом файле. Сохраните топ-3 наиболее критических замечания (не связанных с форматированием). Код-ревью: в качестве ревьюера изучите предоставленный фрагмент кода (другого студента или заранее подготовленный). В формате комментариев укажите две конкретные проблемы: одну — касающуюся потенциальной ошибки (например, отсутствие обработки null), вторую — касающуюся читаемости/архитектуры (например, слишком длинный метод). Рефакторинг: выполните рефакторинг метода convert, устранив один выявленный «запах» (например, «Длинный метод» через выделение метода валидации). Код после рефакторинга должен

проходить все существующие тесты.

7. Мониторинг производительности Теоретический вопрос: какие метрики являются ключевыми для оценки производительности веб-сервиса? Дайте определение времени отклика (response time), пропускной способности (throughput) и процентилям (p95, p99). Практическое задание Настройка мониторинга: используя инструмент k6, напишите скрипт нагрузочного теста для эндпоинта /api/convert. Сценарий: 15 виртуальных пользователей (VUs) в течение 30 секунд отправляют запросы на конвертацию случайных (но валидных) сумм. Сбор метрик: запустите тест. Сохраните сводный отчет (stdout). Обратите внимание на ключевые метрики: http_req_duration (среднее, p95), http_reqs (общее количество), iterations. Анализ: на основе полученных данных определите, удовлетворяет ли сервис требованиям производительности (Требование: p95 времени ответа < 500 мс). Сформулируйте краткий вывод.

8. Оптимизация кода Теоретический вопрос: Объясните разницу между оптимизацией «до того как» (premature optimization) и обоснованной оптимизацией. На каких данных должна основываться обоснованная оптимизация? Практическое задание В ходе нагрузочного теста выявлено «узкое место» — метод getExchangeRate(from, to), который каждый раз делает «дорогой» синхронный вызов к внешнему API или читает весь файл с курсами. Профилирование (демо): изучите предоставленный искусственный отчет профилировщика, где видно, что >80% времени тратится в этом методе. Оптимизация: не меняя сигнатуру метода, предложите и реализуйте одну оптимизацию для устранения узкого места. Варианты: добавление in-memoю кэша (например, HashMap/dict) на 5 минут, ленивая загрузка данных и т.д. Верификация: запустите упрощенный нагрузочный тест (например, 10 последовательных вызовов) и замерьте среднее время до и после оптимизации. Сделайте вывод об эффективности.

9. Использование ИИ для анализа и отладки Теоретический вопрос: каковы основные преимущества и риски использования ИИ (например, больших языковых моделей) в процессе отладки и анализа кода? Практическое задание. В проекте есть неочевидная ошибка: при определенной последовательности запросов (например, конвертация USD->EUR, затем EUR->JPY) курс для одной из пар может рассчитаться неверно. Постановка задачи ИИ: смоделируйте диалог с ИИ-помощником (например, в ChatGPT). Предоставьте ему: Фрагмент кода ключевого метода. Пример failing-теста или описание симптомов. Попросите проанализировать возможные причины. Анализ ответа: на основе ответа ИИ сформулируйте гипотезу о потенциальной причине ошибки (например, «проблема округления», «неверное обновление кэша»). Локализация: используя гипотезу, найдите в реальном коде потенциально проблемную строку. Укажите ее номер и объясните, почему она может быть причиной.

10. Оптимизация кода с помощью ИИ Теоретический вопрос: Как можно использовать ИИ не для генерации кода «с нуля», а для его улучшения (оптимизации)? Приведите пример конкретной задачи по оптимизации, которую можно поручить ИИ. Практическое задание. Вам дан метод validateCurrencyCode(String code), который работает корректно, но написан неэффективно (например, использует цепочку if-else или регулярное выражение при каждом вызове). Запрос к ИИ: сформулируйте и выполните запрос к ИИ-инструменту (например, GitHub Copilot в чате): «Проанализируй этот метод валидации кода валюты [вставь код]. Предложи две альтернативные, более производительные реализации. Объясни, почему они быстрее». Оценка предложений: из предложенных ИИ вариантов выберите один, который, на ваш взгляд, является лучшим по соотношению производительность/читаемость. Реализация и проверка: замените текущую реализацию метода на выбранный оптимизированный

вариант. Убедитесь, что существующие тесты проходят. Кратко аргументируйте свой выбор.

11. Руководство проверкой работоспособности Теоретический вопрос: каковы ключевые обязанности и зона ответственности руководителя проверки работоспособности (Test Lead/QA Lead) на этапе подготовки к релизу? Практическое задание. Вы — руководитель проверки для сервиса Currency Converter перед выпуском версии v1.1. Анализ артефактов. Вам предоставлены: Отчет о выполнении тестов (85% passed, 15% failed). Dashboard SonarQube (качество кода: А, есть 2 критические уязвимости). Отчет о нагрузочном тесте (p95 = 620 мс). Список открытых дефектов в Jira (3 штуки, приоритет Medium). Принятие решения: Напишите краткое, но структурированное заключение (Approval/Rejection) на имя менеджера продукта. В заключении: Резюмируйте текущее состояние. Укажите, является ли какая-либо из перечисленных проблем блокирующей для релиза. Почему? Дайте четкую рекомендацию: Выпускать или Отложить выпуск. Если отложить, укажите обязательные условия для снятия блокировки.

12. Руководство интеграцией. Теоретический вопрос: что такое непрерывная интеграция (CI)? Опишите типичный пайплайн CI для проекта на Python/Java. Какие этапы должны в него входить для обеспечения качества? Практическое задание. Ваша задача — настроить процесс интеграции для нового модуля NotificationService, который должен быть подключен к основному сервису Currency Converter. Проектирование пайплайна: напишите конфигурационный файл CI (.gitlab-ci.yml или GitHub Actions) для проекта Currency Converter. Пайплайн должен включать последовательные стадии: build — сборка/установка зависимостей. static_analysis — запуск линтера и анализатора уязвимостей. unit_test — запуск модульных тестов с генерацией отчета о покрытии. integration_test — запуск интеграционных тестов (с поднятием тестовой БД и зависимостей в Docker). Интеграция модуля: в коде основного сервиса есть заглушка для отправки уведомления. Ваше задание — не код, а планирование. Опишите шаги (checklist) для безопасного включения реального NotificationService в пайплайн интеграции. Включите пункты о: тестировании в изоляции, feature-флагах, мониторинге ошибок после деплоя.

Разработчики:



(подпись)

доцент

(занимаемая должность)

А.Г. Балахчи

(инициалы, фамилия)

Программа составлена в соответствии с требованиями ФГОС ВО и учетом рекомендаций ПООП по направлению подготовки 09.03.03 «Прикладная информатика».

Программа рассмотрена на заседании кафедры естественнонаучных дисциплин

Протокол № 2 от «02» марта 2026 г.

и.о. зав. кафедрой



А.Г. Балахчи

Настоящая программа, не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.