Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(МИ ВлГУ)

Кафедра *ИС*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
19.05.2026

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Тестирование информационных систем

Направление подготовки *09.03.02 Информационные системы и технологии*

Профиль подготовки *Информационные системы и технологии*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
7	**72 / 2**	**16**		**16**	**1,6**	**0,25**	**33,85**	**38,15**	**Зач.**
Итого	**72 / 2**	**16**		**16**	**1,6**	**0,25**	**33,85**	**38,15**	

Муром, 2026 г.

1. Цель освоения дисциплины

Цель изучения курса «Тестирование информационных систем» - ознакомление студентов с основными видами и методами тестирования информационных систем и программного обеспечения.

В курсе изучаются способы обеспечения качества программного продукта, классы критериев тестирования, разновидности тестирования, модульное, интеграционное и системное тестирование, общие принципы автоматизации тестирования, издержки тестирования, методология разработки через тестирование.

2. Место дисциплины в структуре ОПОП ВО

Изучение данной дисциплины базируется на следующих дисциплинах: 1. Основы теории алгоритмов 2. Объектно-ориентированное программирование 3. Стандарты программирования Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин 1. Интерфейсы информационных систем 2. Интеллектуальные системы и технологии 3. Инструментальные средства информационных систем

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ПК-5 Способность выполнять работы по созданию (модификации) и сопровождению информационных систем	ПК-5.1 Применяет современные средства создания, модификации и сопровождения информационных систем	Уметь применять современные средства создания, модификации и сопровождения информационных систем (ПК-5.1)	тест
ПК-1 Способность проводить исследования на всех этапах жизненного цикла программных средств	ПК-1.1 Применяет основные подходы исследования на всех этапах жизненного цикла программых средств	Уметь применять основные подходы исследования на всех этапах жизненного цикла программных средств (ПК-1.1)	тест
ОПК-8 Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем.	ОПК-8.1 Демонстриует знания методологий и основных методов математического моделирования, классификации и условия применения моделей, основных методов и средств проектирования информационных и автоматизированных систем, инструментальных средств моделирования и проектирования информационных и автоматизированных систем	Знать методологии и основные методы математического моделирования, классификации и условия применения моделей, основных методов и средств проектирования информационных и автоматизированных систем, инструментальных средств моделирования и проектирования информационных и автоматизированных систем (ОПК-8.1)	тест
	ОПК-8.2 Моделирует и проектирует информационные и автоматизированные системы	Владеть навыками моделирования и проектирования информационных и автоматизированных	

		систем (ОПК-8.2)	
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;	ОПК-6.3 Программирует, отлаживает и тестирует прототипы программно-технических комплексов	Владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов (ОПК-6.3)	тест

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетных единицы, 72 часа.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Введение. Функциональное тестирование	7	12		12					34	тестирование
2	Нефункциональное тестирование.	7	4		4					4,15	тестирование
Всего за семестр		72	16		16			1,6	0,25	38,15	Зач.
Итого		72	16		16			1,6	0,25	38,15	

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 7

Раздел 1. Введение. Функциональное тестирование

Лекция 1.

Введение. Основные понятия. Характеристики качества ПО. Классификация видов тестирования (2 часа).

Лекция 2.

Преимущества и недостатки автоматизации тестирования. Пирамида тестирования. Требования к тестам. Рассмотрение примеров хорошего и плохого написания модульных тестов (2 часа).

Лекция 3.

Методология разработки через тестирование TDD (2 часа).

Лекция 4.

Ручное тестирование. Метод фокус групп. Альфа и бета тестирование (2 часа).

Лекция 5.

Инструменты тестировщика ПО (QA-инженера) (2 часа).

Лекция 6.

Тестирование юзабилити. Инструменты для тестирования юзабилити (2 часа).

Раздел 2. Нефункциональное тестирование.

Лекция 7.

Тестирование API. Тестирование REST API. Инструмент Postman (2 часа).

Лекция 8.

Организация тестирования в команде разработчиков (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ

Семестр 7

Раздел 1. Введение. Функциональное тестирование

Лабораторная 1.

Модульное тестирование (4 часа).

Лабораторная 2.

Интеграционное тестирование (4 часа).

Лабораторная 3.

Приемочное тестирование (4 часа).

Раздел 2. Нефункциональное тестирование.

Лабораторная 4.

Нефункциональное тестирование (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Качество ПО, задача и роль тестирования. Требования, предъявляемые к тестировщику.
2. Организация тестирования. Взаимодействие тестировщика с другими участниками команды разработки. Жизненный цикл дефекта. Оформление отчета об ошибке.
3. Задачи тестировщика на различных этапах разработки ПО. План тестирования. Тестовый сценарий. Регрессионное тестирование. Покрытие требований.
4. Виды тестирования: функциональное, нагрузочное, стабильности, локализации, usability, совместимости и др. Примеры тестовых сценариев и дефектов каждого вида.
5. Схема динамического тестирования. Основные проблемы тестирования. Подготовка тестовых данных. Понятие оракула. Типичные ошибки.
6. Статическое тестирование. Объекты статического тестирования. Методы статического тестирования. Роль и место в производственном процессе.
7. Автоматизация тестирования. Достоинства и недостатки. Средства автоматизации тестирования и виды тестирования. Планирование автоматизации.
8. Специфика проектов разработки ПО и стратегии тестирования при различных уровнях формализации в проекте. Тестирование и требования к ПО.
9. Организация тестирования в гибких методологиях. Разработка, управляемая тестированием и модульное тестирование.
10. Управление тестированием и отчетность. Задачи управления. Цели и виды отчетов, периодичность формирования. Прогнозирование рисков.
11. Классификация тестирования по степени полноты. Модульное, интеграционное и системное тестирование. Задачи и целесообразность применения. Понятие покрытия кода. Влияние каждого вида на процесс разработки и качество ПО.
12. Виды нагрузочного тестирования и их задачи. Критерии производительности. Инструменты автоматизации нагрузочного тестирования.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

4.2 Форма обучения: заочная

Уровень базового образования: среднее профессиональное.
Срок обучения 3г 6м.

Семестр	Трудоемкость, час./ зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль,час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
7	**72 / 2**	**4**		**4**	**2**	**0,5**	**10,5**	**57,75**	**Зач.(3,75)**
Итого	**72 / 2**	**4**		**4**	**2**	**0,5**	**10,5**	**57,75**	**3,75**

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Введение. Функциональное тестирование	7	2		4					53	Устный опрос
2	Нефункциональное тестирование.	7	2							4,75	Устный опрос
Всего за семестр		72	4		4	+		2	0,5	57,75	Зач.(3,75)
Итого		72	4		4			2	0,5	57,75	3,75

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 7

Раздел 1. Введение. Функциональное тестирование

Лекция 1.

Введение. Функциональное тестирование. Методы функционального тестирования (2 часа).

Раздел 2. Нефункциональное тестирование.

Лекция 2.

Нефункциональное тестирование. Методология разработки через тестирование (2 часа).

4.2.2.2. Перечень практических занятий

Не планируется.

4.2.2.3. Перечень лабораторных работ

Семестр 7

Раздел 1. Введение. Функциональное тестирование

Лабораторная 1.

Модульное тестирование (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Качество ПО, задача и роль тестирования. Требования, предъявляемые к тестировщику.
2. Организация тестирования. Взаимодействие тестировщика с другими участниками команды разработки. Жизненный цикл дефекта. Оформление отчета об ошибке.
3. Задачи тестировщика на различных этапах разработки ПО. План тестирования. Тестовый сценарий. Регрессионное тестирование. Покрытие требований.
4. Виды тестирования: функциональное, нагрузочное, стабильности, локализации, usability, совместимости и др. Примеры тестовых сценариев и дефектов каждого вида.
5. Схема динамического тестирования. Основные проблемы тестирования. Подготовка тестовых данных. Понятие оракула. Типичные ошибки.
6. Статическое тестирование. Объекты статического тестирования. Методы статического тестирования. Роль и место в производственном процессе.
7. Автоматизация тестирования. Достоинства и недостатки. Средства автоматизации тестирования и виды тестирования. Планирование автоматизации.
8. Специфика проектов разработки ПО и стратегии тестирования при различных уровнях формализации в проекте. Тестирование и требования к ПО.
9. Организация тестирования в гибких методологиях. Разработка, управляемая тестированием и модульное тестирование.
10. Управление тестированием и отчетность. Задачи управления. Цели и виды отчетов, периодичность формирования. Прогнозирование рисков.
11. Классификация тестирования по степени полноты. Модульное, интеграционное и системное тестирование. Задачи и целесообразность применения. Понятие покрытия кода. Влияние каждого вида на процесс разработки и качество ПО.
12. Виды нагрузочного тестирования и их задачи. Критерии производительности. Инструменты автоматизации нагрузочного тестирования.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

1. Дефект, стоимость дефекта. Полное тестирование. Серьезность дефекта. Статическое и динамическое тестирование. Тестовые условия и тестовые сценарии. Тестирование и требования. Альфа- и Бета-тестирование. Тестирование продукта в процессе эксплуатации.
2. Тестовые данные и качество тестирования. Классы эквивалентности, граничные условия, попарное тестирование. PICT.
3. Классификация по видам тестирования. Функциональное и не функциональное тестирование. Виды нефункционального тестирования.
4. Жизненный цикл дефекта. Инструменты поддержки тестирования. Взаимодействие в команде в процессе тестирования. Баг-треккер. Атрибуты дефекта. Требования к отчету о выявленном дефекте. Состояния дефекта в процессе жизненного цикла.
5. Тестирование безопасности, нагрузочное тестирование и тестирование usability. Требования безопасности. Инструменты для поиска уязвимостей. SQL и JS инъекции. Роль

тестирования usability. Тестирование с т.з. применимости для людей с ограниченными возможностями.

6. Технология нагрузочного тестирования. Основы HTTP. Виды и задачи нагрузочного тестирования. Тестирование производительности, краш-тестирование. Отчетность при нагрузочном тестировании.

7. Технология нагрузочного тестирования. Виды и задачи нагрузочного тестирования.

8. Классификация тестов по стадии разработки. Автоматизация тестирования. Основные понятий и планирование автоматизации.

9. Инструменты автоматизации функционального и нагрузочного тестирования.

10. Формализация тестирования. Листы проверки, тест-кейсы, матрицы трассировки. Покрытия. Тест-план. Содержание и назначение. Мониторинг и управление тестированием.

11. Тестирование в гибких производственных процессах. Scrum.

12. Классификация тестов по стадии разработки. Модульное тестирование, интеграционное, системное и приемочное тестирование. Регрессионное тестирование. Автоматизация тестирования. Основные понятий и планирование автоматизации.

13. Инструменты автоматизации функционального и нагрузочного тестирования. JMeter. Автоматизация тестирования web-приложений на основе Katalon Recorder.

14. Формализация тестирования. Листы проверки, тест-кейсы, матрицы трассировки. Покрытия. Формы представления требований к ПО. Определения покрытия. Покрытие требований, покрытие кода.

15. Тест-план. Содержание и назначение. Мониторинг и управление тестированием. Регулярность и содержание отчетов. Сравнение текущего состояния с планом и прогноз.

16. Тестирование в гибких производственных процессах. Scrum. Работа тестировщика в “гибкой команде”. Роль взаимодействия с Заказчиком. Роль автоматизации.

17. Вопросы с собеседований. Рассмотрение практических проблем. Анализ пула вопросов и качества тестировщика.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности студентов для достижения запланированных результатов обучения и реализации компетентностного подхода. В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лабораторных работ применяются имитационный или симуляционный подход, когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания.

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

- изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;
- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;
- закрепление теоретического материала при проведении лабораторных работ с использованием учебного и научного оборудования и приборов, выполнения проблемно-ориентированных, поисковых, творческих заданий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения : учебное пособие / А. В. Проскуряков. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2022. — 197 с. - https://www.iprbookshop.ru/125702.html

2. Сухов, В. Д. Экономическое обоснование разработки программного продукта : учебно-методическое пособие для бакалавров / В. Д. Сухов, А. А. Киселев, А. И. Сазонов. — Москва : Ай Пи Ар Медиа, 2022. — 108 с. - https://www.iprbookshop.ru/122612.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Канер С., Фолк Дж., Кек Нгуен Е. Тестирование программного обеспечения: Пер. с англ. -- К: ДиаСофт, 2000. - 544 с. - 25 экз.

2. Луиза Тамре. Введение в тестирование программного обеспечения. - М.: Издательский дом «Вильямс», 2003. - С. 368 - 25 экз.

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

iprbookshop.ru

mivlgu.ru/iop

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

7-Zip (GNU LGPL)

Mozilla Firefox (MPL)

Notepad++ (GNU GPL 3)

РЕД ОС (Соглашение №140/05-21У от 18.05.2021 года о сотрудничестве в области науки, развития инновационной деятельности)

РЕД "База данных" (Соглашение №140/05-21У от 18.05.2021 года о сотрудничестве в области науки, развития инновационной деятельности)

РЕД Эксперт (Соглашение №140/05-21У от 18.05.2024 года о сотрудничестве в области науки, развития инновационной деятельности)

Double Commander (GNU GPL 2+)

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

iprbookshop.ru

mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория распределенных систем

12 персональных компьютеров; проектор Nec V300X; экран настенный Lumien Master Picture

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу, связанную с тестированием информационной системы или ее подсистем в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – зачет. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.02 Информационные системы и технологии* и профилю подготовки *Информационные системы и технологии*
Рабочую программу составил *к.т.н., доцент Канунова Е.Е.*_____

Программа рассмотрена и одобрена на заседании кафедры *ИС*

протокол № 21 от 05.05.2026 года.
Заведующий кафедрой *ИС* _____*Андрианов Д.Е.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 12.05.2026 года.
Председатель комиссии ФИТР _____ *Кутарова Е.И.*
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине
Тестирование информационных систем

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Рейтинг-контроль 1
Какие пункты содержит спецификация программного обеспечения?
Что такое управляющий граф программы?
Какие существуют критерии выбора тестов?
Какие предъявляются требования к идеальному критерию?
Какие существуют классы критериев?
Структурные критерии выбора тестов
Функциональные критерии выбора тестов
Стохастические критерии выбора тестов
Мутационный критерий выбора тестов
Рейтинг-контроль 2
Оценка покрытия программы и проекта.
Методика интегральной оценки тестированности.
Модульное тестирование.
Интеграционное тестирование.
Особенности интеграционного тестирования при процедурном программировании.
Особенности интеграционного тестирования для объектно-ориентированного программирования.
Системное тестирование.
Регрессионное тестирование.
Рейтинг-контроль 3
Автоматизация тестирования.
Издержки тестирования.
Фазы процесса тестирования.
Тестовый цикл.
Планирование тестирования.
Типы тестирования.
Подходы к разработке тестов.
Тестирование спецификации.
Тестирование сценариев.
Документация и оценка индустриального тестирования.
Жизненный цикл дефекта.
Тестовый отчет.
Оценка качества тестов.
Цели и задачи регрессионного тестирования
Виды регрессионного тестирования.
Управляемое регрессионное тестирование.
Классификация тестов при отборе
Возможности повторного использования тестов.
Классификация выборочных методов.

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Устный опрос 5 вопросов, тест	До 20 баллов
Рейтинг-контроль 2	Устный опрос 5 вопросов, тест	До 20 баллов

Рейтинг-контроль 3	Устный опрос 5 вопросов, тест	До 30 баллов
Посещение занятий студентом		5
Дополнительные баллы (бонусы)		5
Выполнение семестрового плана самостоятельной работы		20

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

Методические материалы, характеризующие процедуры оценивания

Из перечня вопросов для проведения зачета формируется индивидуальное задание студенту, по результатом которого, а также с учетом работы в семестре, формируется семестровый рейтинг студента, на основе которого студенту выставляется зачет при достижении количества баллов рейтинга пороговой величины.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	***Продвинутый уровень***

50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. При каком тестировании оценивается работа ПО под определённой нагрузкой?
- *Тестирование производительности
- Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

2. При каком тестировании оценивается удобство пользователя при взаимодействии с разными параметрами интерфейса (кнопки, цвета, выравнивание и так далее)?
- Тестирование производительности
- *Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

3. При каком тестировании оценивается правильность логики использования программного продукта?
- Тестирование производительности
- Тестирование пользовательского интерфейса
- *Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

4. При каком тестировании определяется безопасность ПО (защищено ли оно от атак хакеров, несанкционированного доступа к данным и так далее)?
- Тестирование производительности

- Тестирование пользовательского интерфейса
- Тестирование UX
- *Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

5. При каком тестировании оценивается вероятность возникновения проблем при установке, удалении, а также обновлении ПО?
- Тестирование производительности
- Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- *Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

6. При каком тестировании проверяется работа программного продукта в определённом окружении?
- Тестирование производительности
- Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- *Тестирование совместимости
- Тестирование надежности
- Тестирование локализации

7. При каком тестировании проверяется работа системы при длительной средней ожидаемой нагрузке?
- Тестирование производительности
- Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- *Тестирование надежности
- Тестирование локализации

8. При каком тестировании выполняется оценка правильности версии программного продукта (языковой и культурный аспекты)?
- Тестирование производительности
- Тестирование пользовательского интерфейса
- Тестирование UX
- Тестирование защищенности
- Инсталляционное тестирование
- Тестирование совместимости
- Тестирование надежности
- *Тестирование локализации

9. В какой последовательности выполняются этапы разработки новой функции в проекте в рамках методолгии TDD? (Green - этап написания кода, который проходит тест; Red - написание теста, который проваливается; Refactoring - доработка программнго кода).

- Green-Red-Refactoring.
- Green-Refactoring-Red.
- *Red-Green-Refactoring.
- Red-Refactoring-Green.
- Refactoring-Red-Green.
- Refactoring-Green-Red.

10. Выберите главное правило методологии разработки через тестирование.

- Нужно проводить все виды тестирования программного обеспечения;
- *Никогда не писать рабочий код без неудачно завершившегося теста;
- Программный код следует максимально покрывать тестами;
- Во время работы над проектом не увольнять тестировщиков.

11. К какому виду тестирования относится тестирование API?

- к UI -тестированию
- к дымовому тестированию
- к модульному тестированию
- *к интеграционному тестированию

12. Какой метод тестирования применяется когда тестировщик не знает как устроен продукт, как его нужно использовать и действует как бы вслепую — то есть воспроизводит действия обычного пользователя без оглядки на инструкции?

- *Метод черного ящика
- Метод белого ящика
- Метод серого ящика
- Метод слепого ящика

13. Кто такой QA-инженер?

- Специалист по тестированию ПО
- Специалист по проектированию ПО
- Программист
- Аналитик
- *Специалист по обеспечению качества разработки ПО

14. Как называется тестирование всей компьютерной системы перед этапом пользовательского тестирования (UAT)?

- *альфа-тестирование?
- бета-тестирование?
- гамма-тестирование?
- дельта-тестирование?

15. Какое тестирование проводится когда ПО уже готово к релизу (проверяется соответствие ПО требованиям)?

- альфа-тестирование?
- бета-тестирование?
- *гамма-тестирование?
- дельта-тестирование?

16. Как называется тип тестирования при котором проверяется какие функции ПО реализованы и насколько верно они реализованы?

Функциональное

17. Напишите аббревиатуру методологии разработки программного обеспечения, которая основывается на повторении коротких циклов разработки: изначально пишется тест, покрывающий желаемое изменение, затем пишется программный код, который реализует желаемое поведение системы и позволит пройти написанный тест, а затем проводится рефакторинг написанного кода с постоянной проверкой прохождения всех тестов.

TDD

18. Напишите аббревиатуру методологии разработки ПО на основе поведения, которая строится на написании нескольких пользовательских сценариев, под которые составляются тесты

BDD

19. Как называется самое популярное инструментальное средство тестирования API

Postman

20. Какие тесты находятся в основании пирамиды тестирования? (Напишите прилагательное на русском языке)

Модульные

21. Как называется тестирование, которое выполняется для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение, стартует и выполняет основные функции?

Дымовое

22. Как называется процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта?

Тестирование

23. Если очень долго проводить одни и те же проверки, то скорее всего новых багов найти не удастся. Поэтому нужно обновлять тестовую базу. Этот эффект называется "парадоксом ..."

пестицида

24. Как называется любое несовершенство в работе софта или когда ожидаемый результат не соответствует фактической работе приложения?

дефект

25. Как называется тестирование, при котором выполняется верификация существующих функциональных и нефункциональных зон после того, как были изменены отдельные части приложения или добавлены новые функциональные возможности?

Регрессионное

26. Правило ... гласит, что 80% дефектов как правило таятся в 20% модулей программы

Парето

27. Как называют тестирование, при котором проверяют поведение системы без взаимодействия с программой или исходным кодом? У специалиста нет сведений об исходных тестовых данных и состоянии системы. Он просматривает системные журналы и журнал событий приложения. Так ищет шаблоны и последовательности записей, которые укажут на корректное или некорректное поведение программы.

Пассивное

28. Как называют тестирование, при котором программу тестируют без запуска? Находят ошибки, когда повторно проверяют код или используют утилиту для анализа: находят конструкции или последовательности операторов, которые приводят к отказу работы приложения.

Статическое

29. Как называют тестирование, при котором программу тестируют при запуске? Иногда даже до ее полной готовности. Так проверяют участки кода, тестовые сценарии применяют к отдельным функциям или модулям программы.

Динамическое

30. Как называется методология, разработанная Джеффом Де Люка иПитером Коадом, основной целью которой является разработка реального, работающего программного обеспечения систематически, в поставленные сроки.

FDD

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/mod/quiz/view.php?id=59085

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.