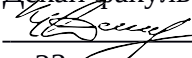


МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерных сетей

УТВЕРЖДАЮ
Декан факультета ИВТ
 Д.Ю. Чалый
« 23 » мая 2023 г.

Рабочая программа дисциплины
«Основы тестирования программного обеспечения»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Профиль
«Информатика и компьютерные науки»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 17 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Основы тестирования программного обеспечения» являются обучение базовым подходам к тестированию ПО с использованием современных методов и технологий, обучение использованию автоматизированных инструментов тестирования

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Основы тестирования программного обеспечения» относится к вариативной части (дисциплина по выбору) ОП бакалавриата.

В курсе рассматриваются виды и уровни тестирования программного обеспечения, вырабатываются навыки работы с инструментами тестирования. Содержание курса тесно связано фактически со всеми дисциплинами, которые изучались студентами. Освоению данной программы предшествуют учебные курсы по программированию и современным информационным технологиям.

Дисциплина «Основы тестирования программного обеспечения» обеспечивает закрепление и углубление теоретических знаний и практических навыков по основным дисциплинам ИТ-цикла. Дисциплина способствует профессиональному росту студентов, повышению их общеметодологического уровня, а также дальнейшему развитию навыков научно-исследовательской деятельности.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ОПК-3.1 Демонстрирует знание и умение применять на практике методы алгоритмизации ОПК-3.2 Умеет решать прикладные задачи с использованием языков программирования	Знать: 1. - основные методы тестирования программных продуктов. Уметь: - разрабатывать тесты для программного обеспечения; Владеть: - практическими навыками работы с инструментами тестирования программного обеспечения.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зач. ед., 144 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости		Форма промежуточной аттестации (по семестрам)
			лекции	практические	лабораторные	консультации	академические занятия	самостоятельная работа	экзамен	
			Контактная работа							
1.	Понятие и предмет тестирования	4	2		4			11	Лабораторная работа	
2.	Тестирование методом чёрного ящика	4	2		4			11	Лабораторная работа	
3.	Тестирование методом белого ящика	4	2		5			11	Лабораторная работа	
4.	Планирование тестирования.	4	2		4			11	Лабораторная работа	
5.	Автоматизация тестирования.	4	3		5			11		
6.	Инструментальные средства тестирования.	4	2		5			12,7	Лабораторная работа	
7.	Методология разработки через тестирование	4	2		4			11		
8.	Средства и методы тестирования приложений определённых типов	4	3		5			11	Тест	
	Всего за 4 семестр		18		36		0,3	89,7	Зачет	
	Всего		18		36		0,3	89,7		

Содержание разделов дисциплины:

1. Понятие и предмет тестирования. Тестирование разработчиками и тестировщиками. Идея и процесс тестирования. Требования и спецификации. Понятие ошибок, дефектов и сбоев. Виды и уровни тестирования. Тест-кейсы и наборы тест-кейсов. Понятие об автоматизации тестирования.

2. Тестирование методом чёрного ящика. Преимущества и недостатки тестирования методом чёрного ящика. Техника тестирования на основе классов эквивалентности. Техника тестирования на основе граничных условий. Техника тестирования по таблицам решений. Техника попарного тестирования. Техника тестирования переходов между состояниями. Техника тестирования сценариев использования.
3. Тестирование методом белого ящика. Преимущества и недостатки тестирования методом белого ящика. Техника тестирования потоков управления. Критерии покрытия. Цикломатическая сложность и базовые пути графа потока управления. Метод базового пути для покрытия операторов и рёбер графа потока управления. Техника тестирования потоков данных.
4. Планирование тестирования. Тестирование по сценарию и исследовательское тестирование, их преимущества и недостатки. Таксономии дефектов. Документы тестирования: план тестирования, спецификация дизайна тестов, спецификации тест-кейсов, спецификация процедуры тестирования, отчёт тестирования, отчёт об инцидентах.
5. Автоматизация тестирования. Выгоды и риски автоматизации. Области применения автоматизации. Метрики тестирования. Инструментальные средства управления тестированием и отчётами о дефектах. Непрерывная интеграция и инструментарий для её поддержки.
6. Инструментальные средства тестирования. Фреймворки модульного тестирования. Техники интеграционного и системного тестирования. Моск-объекты и заглушки, область их применения. Организация кода тестов и инструментальные средства её поддержки.
7. Методология разработки через тестирование (test-driven development, TDD). Мотивация и область применимости методологии разработки через тестирование. Понятие о рефакторинге кода. Основной цикл разработки через тестирование «тест-код-рефакторинг».
8. Средства и методы тестирования приложений определённых типов. Тестирование настольных приложений. Тестирование мобильных приложений. Тестирование веб-приложений.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Мастер-класс – это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний по предложенному алгоритму.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются: для разработки документов, презентаций, для работы с электронными таблицами

OfficeStd 2013 RUS OLP NL Acdmc 021-10232

LibreOffice (свободное)

издательская система LaTeX;

- Среда разработки программных проектов IntelliJ IDEA
<http://www.jetbrains.com/idea/> Доступ свободный.

– для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Старолетов, С. М. Основы тестирования и верификации программного обеспечения : учебное пособие / С. М. Старолетов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 344 с. — ISBN 978-5-8114-5239-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/138181>

2. Лагутина, Н. С., Разработка программных приложений : практикум для студентов, обучающихся по направлению Фундаментальная информатика и информационные технологии / Н. С. Лагутина, Ю. А. Ларина, А. М. Васильев; Яросл. гос. ун-т., Ярославль, ЯрГУ, 2014, 71с

3. Лагутина, Н. С., Разработка программных приложений [Электронный ресурс] : практикум для студентов, обучающихся по направлению Фундаментальная информатика и информационные технологии / Н. С. Лагутина, Ю. А. Ларина, А. М. Васильев; Яросл. гос. ун-т., Ярославль, ЯрГУ, 2014, 71с. <http://www.lib.uniyar.ac.ru/edocs/iuni/20140402.pdf>

б) дополнительная:

1. Вигерс, К., Разработка требований к программному обеспечению = Software requirements : пер. с англ / К. Вигерс, Д. Битти - 3-е изд., доп., М., Русская редакция, 2015, 718с

в) ресурсы сети «Интернет»

- Среда разработки программных проектов IntelliJ IDEA
<http://www.jetbrains.com/idea/> Доступ свободный.

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

-учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);

-учебные аудитории для проведения лабораторных занятий;

- учебные аудитории для проведения групповых и индивидуальных консультаций,

- учебные аудитории для проведения текущего контроля и промежуточной аттестации;

-помещения для самостоятельной работы;

-помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и

обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

Автор(ы) :

Доцент кафедры КС _____ И. В. Парамонов

**Приложение №1 к рабочей программе дисциплины
«Основы тестирования программного обеспечения»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для
оценки знаний, умений, навыков и (или) опыта деятельности,
характеризующих этапы
формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе
текущей аттестации**

Примеры заданий для лабораторных работ

Пример задания по теме «Понятие и предмет тестирования»:

Напишите тест-кейсы и код для тестирования следующих свойств класса HashMap из стандартной библиотеки Java:

1. объект, сохраняемый с помощью метода put(), может быть получен с помощью метода get();
2. добавление значения, соответствующего существующему значению ключа, затирает старое значение;
3. метод clear() удаляет всё содержимое контейнера;
4. null может быть использован в качестве ключа. Ожидаемый результат:
 1. zip-архив, содержащий Java-проект с разработанными тестами;
 2. описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;
 3. файл build.gradle должен размещаться в корне архива без дополнительных подкаталогов;
 4. тесты должны успешно выполняться с помощью вызова ./gradlew test или gradlew.bat test из каталога, в котором был распакован архив.

Критерии оценивания:

Критерий	Баллы
Проект правильно организован. Тесты успешно выполняются.	10
Присутствует тест на проверку элемента в коллекцию, который был туда помещён ранее.	5
Присутствует тест на получение элемента коллекции, который туда не помещали.	5
Присутствует тест на повторное добавление элемента с тем же значением ключа.	5
Присутствует тест на очистку непустой коллекции.	5
Присутствует тест на очистку пустой коллекции.	5
Присутствует тест на использование null в качестве ключа.	5
Имена методов понятны и верно описывают назначение соответствующих тестов.	4
Описания тестов понятны.	4
Тест-кейсы сгруппированы по требованиям и явно указаны требования, к которым эти тест-кейсы относятся.	4

Инициализация тестируемого объекта выполняется в блоке @Before или при инициализации поля класса.	4
В тестах с изменением состояния (перезапись значения по ключу, удаление) состояние тестируемого объекта проверяется до изменения, а не только после него.	4

Баллы суммируются, максимальная сумма баллов – 60.

Итоговая оценка:

Баллы	Оценка
0-29	Неудовлетворительно
30-39	Удовлетворительно
40-49	Хорошо
50-60	Отлично

Пример задания по теме «Тестирование методом чёрного ящика»:

Напишите тест-кейсы и код для тестирования класса с состояниями из проекта course-enrollment (<https://github.com/ivparamonov/software-testing-course-materials/tree/master/course-enrollment>). Для формирования тест-кейсов используйте диаграмму состояний, а также техники классов эквивалентности и граничных условий. При обнаружении ошибок в тестируемом классе, сразу же их исправляйте.

Ожидаемый результат:

zip-архив, содержащий исправленный проект course-enrollment с разработанными тестами;

описание исправлений кода должно размещаться в текстовом файле fixes.txt в корне проекта;

описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;

файл build.gradle должен размещаться в корне архива без дополнительных подкаталогов;

тесты должны успешно выполняться с помощью вызова ./gradlew test или gradlew.bat test из каталога, в котором был распакован архив.

Критерии оценивания:

Критерий	Баллы	
Проект правильно организован. Тесты успешно выполняются.	10	
Ошибки в классе найдены и исправлены	10	
Есть файл с описанием исправленных ошибок	3	
Добавлена проверка на <code>maxStudents <= 0</code>	3	
Присутствует и верен тест (комментарий к состояниям: NF — not full, курс не заполнен; F — full, курс заполнен; WLE — waiting list exists, существует список ожидания)	"NF → NF enroll new"	2
	"NF → NF enroll existent"	2
	"NF → F enroll new"	2
	"NF → NF unenroll existent"	2
	"NF → NF unenroll non-existent"	2
	"F → F enroll existent"	2
	"F → F unenroll non-existent"	2

	"F → NF unenroll existent"	2
	"F → WLE enroll new"	2
	"WLE → WLE enroll"	2
	"WLE → WLE unenroll existent"	2
	"WLE → WLE unenroll non-existent"	2
	"WLE → F unenroll existent"	2
Инициализация тестируемого объекта выполняется в блоке @Before или при инициализации поля класса.		2
Имена методов понятны и верно описывают назначение соответствующих тестов.		2
Описания тестов понятны.		2
Тест-кейсы сгруппированы по требованиям и явно указаны требования, к которым эти тест-кейсы относятся.		2

Баллы суммируются, максимальная сумма баллов – 60.

Итоговая оценка:

Баллы	Оценка
0-29	Неудовлетворительно
30-39	Удовлетворительно
40-49	Хорошо
50-60	Отлично

Пример задания по теме «Тестирование методом белого ящика»:

Напишите тест-кейсы и код для тестирования класса методом белого ящика для проекта course-enrollment (<https://github.com/ivparamonov/software-testing-course-materials/tree/master/course-enrollment>). При обнаружении ошибок в тестируемом классе, сразу же их исправляйте.

Ожидаемый результат:

- zip-архив, содержащий исправленный проект course-enrollment с разработанными тестами;
- описание исправлений кода должно размещаться в текстовом файле fixes.txt в корне проекта;
- описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;
- файл build.gradle должен размещаться в корне архива без дополнительных подкаталогов; тесты должны успешно выполняться с помощью вызова ./gradlew test или gradlew.bat test из каталога, в котором был распакован архив.

Критерии оценивания:

Критерий	Баллы
Проект правильно организован. Тесты успешно выполняются.	10
Ошибки в классе найдены и исправлены	7
Есть файл с описанием исправленных ошибок	3
В тестах вызываются isFullyEnrolled и hasWaitingList	6
В тестах вызываются Вызываются getEnrollmentList и getWaitingList	6

В тестах используются возможности AssertJ	4
В тестах есть проверка maxStudents на положительность	4
В тестах имеется проверка краевых случаев	4
Присутствует и верен тест Enroll: Тест для уже записанного студента	2
Присутствует и верен тест Enroll: Тест для студента из списка ожидания	2
Присутствует и верен тест Enroll: Тест для заполненного курса	2
Присутствует и верен тест Enroll: Тест для незаполненного курса	2
Присутствует и верен тест Unenroll: Тест для студента из списка ожидания	2
Присутствует и верен тест Unenroll: студент записан на курс и список ожидания не пуст	2
Присутствует и верен тест Unenroll: студент записан на курс и список ожидания пуст	2
Присутствует и верен тест Unenroll: тест для студента не из списков	2

Баллы суммируются, максимальная сумма баллов – 60.

Итоговая оценка:

Баллы	Оценка
0-29	Неудовлетворительно
30-39	Удовлетворительно
40-49	Хорошо
50-60	Отлично

Пример задания по теме «Планирование тестирования»:

На основе результатов написания тестов для класса Course напишите эссе, посвящённое сравнению методов тестирования чёрным и белым ящиком. При написании используйте следующие вопросы в качестве опорных:

- Чем отличаются процедуры написания тестов при использовании различных методов?
- Чем отличаются сами тесты, написанные при использовании различных методов?
- Какие ошибки легко могут быть найдены каждым из методов, а какие с трудом поддаются нахождению?
- В какие моменты при выполнении тестирования находятся ошибки при использовании различных методов?

Ожидаемый результат: документ в одном из допустимых форматов (MS Office, OpenOffice/LibreOffice, Markdown, Plain text), содержащий текст эссе. Обращайте внимание на последовательность, логичность и аргументированность изложения! Рассматривайте поставленные вопросы с точки зрения проделанной работы по тестированию класса. Используйте код конкретных тестов для обоснования своей точки зрения.

Критерии оценивания:

Критерий	Баллы
Детально рассмотрен вопрос «Отличие процедур тестирования»	6

Детально рассмотрен вопрос «Отличие написанных тестов»	6
Детально рассмотрен вопрос «Находимые ошибки»	6
Детально рассмотрен вопрос «Моменты обнаружения ошибок»	6
В рассуждениях студента отсутствуют логические ошибки	10
Проведённый анализ является глубоким	10
Проведённый анализ является оригинальным, содержит явно выраженные собственные мысли студента	8
Сделан адекватный вывод, принимающий во внимание все аргументы, встречающиеся в эссе	8

Баллы суммируются, максимальная сумма баллов – 60.

Итоговая оценка:

Баллы	Оценка
0-29	Неудовлетворительно
30-39	Удовлетворительно
40-49	Хорошо
50-60	Отлично

Пример задания по теме «Инструментальные средства тестирования»:

Напишите тест-кейсы и код для тестирования проекта `average-grade-calculator` (<https://github.com/ivparamonov/software-testing-course-materials/tree/master/average-grade-calculator>). Для проверки функциональности класса используйте `mock` интерфейса `Dao`. Найденные ошибки необходимо исправить.

Указание: для создания списков и ассоциативных массивов с фиксированным содержимым используйте средства библиотеки `Guava` (уже добавлена в проект).

Ожидаемый результат:

- zip-архив, содержащий исправленный проект `average-grade-calculator` с разработанными тестами;
- описание исправлений кода должно размещаться в текстовом файле `fixes.txt` в корне архива;
- описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;
- файл `build.gradle` должен размещаться в корне архива без дополнительных подкаталогов; тесты должны успешно выполняться с помощью вызова `./gradlew test` или `gradlew.bat test` из каталога, в котором был распакован архив.

Критерии оценивания:

Критерий	Баллы
Проект правильно организован. Тесты успешно выполняются.	10
Исправлено вычисление среднего значения	2
Добавлена проверка контейнера на пустоту	2
Исправлено использование <code>joining</code>	2
Добавлено сохранение данных о лучшем студенте	2
Имена методов понятны и верно описывают назначение соответствующих тестов	4
Описания тестов понятны	4

Сравнение вещественных чисел в тестах осуществляется верно (с offset)	4
Присутствует Проверка, что метод calculate вызывает retrieve у DAO	2
Присутствует Проверка, что при повторном вызове используется кэш	2
Присутствуют проверки целого или дробного среднего	2
Присутствует Тест с данными для 2 разных месяцев и 1 студента	2
Присутствует Тест для 2 студентов и 1 месяца	2
Присутствует Тест для кэширования нескольких наборов данных	2
Присутствует Тест для студента без оценок	2
Присутствует Тест на очистку кэша	4
Присутствует Тест на запись лучшего студента в базу данных	2
Присутствует Тест для 0 лучших студентов	2
Присутствует Тест для 1 лучшего студента	2
Присутствует Тест для 2 лучших студентов	2
В проекте используется библиотека Guava	4

Баллы суммируются, максимальная сумма баллов – 60.

Итоговая оценка:

Баллы	Оценка
0-29	Неудовлетворительно
30-39	Удовлетворительно
40-49	Хорошо
50-60	Отлично

Список вопросов к зачету:

9. Понятие и предмет тестирования. Требования и спецификации.
10. Понятие ошибок, дефектов и сбоев.
11. Виды и уровни тестирования.
12. Тест-кейсы и наборы тест-кейсов.
13. Преимущества и недостатки тестирования методом чёрного ящика.
14. Техника тестирования на основе классов эквивалентности.
15. Техника тестирования на основе граничных условий.
16. Техника тестирования по таблицам решений.
17. Техника попарного тестирования.
18. Техника тестирования переходов между состояниями.
19. Техника тестирования сценариев использования.
20. Преимущества и недостатки тестирования методом белого ящика.
21. Техника тестирования потоков управления. Критерии покрытия.
22. Цикломатическая сложность и базовые пути графа потока управления.
23. Метод базового пути для покрытия операторов и рёбер графа потока управления.
24. Техника тестирования потоков данных.
25. Тестирование по сценарию и исследовательское тестирование, их преимущества и недостатки. Таксономии дефектов.
26. Документы тестирования: план тестирования, спецификация дизайна тестов, спецификации тест-кейсов, спецификация процедуры тестирования, отчёт тестирования, отчёт об инцидентах.

27. Автоматизация тестирования. Выгоды и риски автоматизации. Области применения автоматизации.
28. Метрики тестирования.
29. Инструментальные средства управления тестированием и отчётами о дефектах.
30. Фреймворки модульного тестирования. Техники интеграционного и системного тестирования. Mock-объекты и заглушки, область их применения. Организация кода тестов и инструментальные средства её поддержки.
31. Методология разработки через тестирование (test-driven development, TDD).
32. Понятие о рефакторинге кода. Основной цикл разработки через тестирование «тест-код-рефакторинг».
33. Тестирование настольных приложений.
34. Тестирование мобильных приложений.
35. Тестирование веб-приложений.

Пример теста на зачете:

1. В состав отчёта о дефекте обычно входят следующие поля Выберите один или несколько ответов:
 - a. шаги по воспроизведению**
 - b. подробное описание**
 - с. чеклист
 - d. ссылка на номер требования
 - е. фамилия разработчика, внёсшего дефект
 - f. идентификатор**
2. Выберите тип тестирования, при котором тестировщик существенно использует структуру исходного текста тестируемой программы
Выберите один ответ:
 - a. тестирование методом прозрачного ящика**
 - b. тестирование методом чёрного ящика
 - с. интеграционное тестирование
 - d. системное тестирование
 - е. модульное тестирование
 - f. регрессионное тестирование
3. Выберите из перечисленного преимущества методики разработки через тестирование
Выберите один или несколько ответов:
 - a. ошибки в требованиях найти легче, чем при использовании классической методики**
 - b. дефекты находятся рано и исправляются без переключения контекста разработчика**
 - с. тестировщик может увидеть особенности реализации, требующие тестирования, но невидимые снаружи
 - d. методика обеспечивает высокий процент покрытия кода
4. Выберите тип тестирования, направленный на проверку правильности работы отдельных компонентов программы (классов, функций и т. п.)
Выберите один ответ:
 - а. интеграционное тестирование
 - b. модульное тестирование**
 - с. тестирование методом прозрачного ящика
 - d. тестирование методом чёрного ящика
 - е. регрессионное тестирование
 - f. системное тестирование
5. Субъектами тестирования программного обеспечения могут быть

Выберите один или несколько ответов:

- a. веб-дизайнеры
- b. тестировщики**
- c. разработчики
- d. заказчики
- e. топ-менеджеры

6. Выберите из перечисленного преимущества тестирования методом белого ящика Выберите один или несколько ответов:

a. тестирование непосредственно проверяет соответствие ПО требованиям пользователя

b. тестировщик может быть уверенным, что тесты выполняются вдоль всех протестированных путей выполнения кода

c. тестировщик не имеет пристрастного отношения к написанному коду

d. тестировщик может увидеть особенности реализации, требующие тестирования, но невидимые снаружи

7. Выберите из перечисленного две наиболее универсальные техники тестирования методом чёрного ящика

Выберите один или несколько ответов:

a. тестирование на основе классов эквивалентности

b. тестирование переходов между состояниями

c. тестирование на основе граничных условий

d. тестирование по таблицам решений

e. попарное тестирование

8. Укажите минимальное количество тестов, необходимое для покрытия всех операторов следующего фрагмента кода

```
if(!var) otherVar = 12;  
std::cout << otherVar << std::endl;
```

Ответ: 1

Правильные ответы выделены жирным шрифтом.

Критерии оценивания:

За каждый выбранный правильный ответ начисляется количество баллов равное единице, делённое на общее количество правильных ответов на вопрос. За каждый выбранный неправильный ответ у студента снимается количество баллов, равное 0.5 балла. Если суммарный балл студента за вопрос оказывается отрицательным, вместо него засчитывается 0.

Набранное количество баллов соответствует оценке за выполнение экзаменационной работы:

Баллы	Оценка	Уровень формирования компетенции ПК-7
0-2	Неудовлетворительно	
3-4	Удовлетворительно	пороговый
5-6	Хорошо	продвинутый
7-8	Отлично	высокий

Методические указания по выставлению зачета

Зачет выставляется по результатам выполнения всех лабораторных работ и зачетного теста на оценку не ниже удовлетворительно.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования,

описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы

формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка «зачтено», «незачтено».

Показатели и критерии, используемые при выставлении оценки подробно описаны в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций».

Высокий уровень формирования компетенций соответствует оценке «отлично» за лабораторные работы и тест.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за лабораторные работы и тест.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за лабораторные работы и тест.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Основы тестирования программного обеспечения»

Методические указания для студентов по освоению дисциплины

Занятия проводятся в интерактивной форме с использованием мультимедиа-технологий. Занятия предполагают наличие дискуссий по поводу тех или иных вопросов разработки программных приложений осуществляемых в результате соответствующего предложения преподавателя.

Применение полученных знаний отрабатывается и во время лабораторных занятий, ориентированных помимо закрепления лекционного материала на разбор различных модельных ситуаций. По большинству тем предусмотрены практические занятия, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам тестирования.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и лабораторных занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – помочь усвоить основы тестирования программного обеспечения. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, лабораторных занятиях или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных лабораторных навыков в течение обучения проводятся мероприятия текущей аттестации в виде лабораторных работ. Также проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце семестра студенты сдают зачет. Зачет по итогам первого семестра выставляется по итогам собеседования по теоретическим вопросам.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Основы тестирования программного обеспечения» самостоятельно студенту крайне сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет по итогам изучения дисциплины студенту практически невозможно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](#) (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](#) доступна в сети университета и через Личный кабинет.