

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

Кафедра компьютерных сетей

**УТВЕРЖДАЮ**

Декан факультета ИВТ



(подпись)

Д. Ю. Чалый

« 18 » \_\_мая\_\_ 2020г.

**Рабочая программа дисциплины**  
**«Основы тестирования программного обеспечения»**

Направление подготовки  
09.03.03 Прикладная информатика

Направленность (профиль)  
«Прикладная информатика в экономике»

Форма обучения  
очная

Программа одобрена  
на заседании кафедры  
от «16» апреля 2020 года, протокол № 8

Программа одобрена НМК  
факультета ИВТ  
протокол № 7 от «17» мая 2020 года

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 образовательной программы

Дисциплина «Основы тестирования программного обеспечения» относится к вариативной части образовательной программы.

Для освоения данной дисциплины студенты должны обладать навыками программирования и использования современных информационных технологий.

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.

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, ООП 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
<b>Профессиональные компетенции</b>		
<b>ПК-2</b> Способен разрабатывать и адаптировать прикладное программное обеспечение	<b>ИД-ПК-2_4</b> Способен осуществлять тестирование компонентов информационных систем по заданным сценариям	<b>Знать:</b> - основные методы тестирования программных продуктов.  <b>Уметь:</b> - разрабатывать и отлаживать программное обеспечение; - разрабатывать тесты для программного обеспечения  <b>Владеть:</b> - практическими навыками работы с инструментами разработки программного обеспечения.



## Содержание разделов дисциплины:

- 1. Понятие и предмет тестирования.**
  - 1.1. Понятие тестирования.
  - 1.2. Тестирование разработчиками и тестировщиками. Идея и процесс тестирования.
  - 1.3. Требования и спецификации. Понятие ошибок, дефектов и сбоев.
  - 1.4. Виды и уровни тестирования.
  - 1.5. Тест-кейсы и наборы тест-кейсов. Понятие об автоматизации тестирования.
- 2. Тестирование методом чёрного ящика.**
  - 2.1. Преимущества и недостатки тестирования методом чёрного ящика.
  - 2.2. Техника тестирования на основе классов эквивалентности.
  - 2.3. Техника тестирования на основе граничных условий.
  - 2.4. Техника тестирования по таблицам решений.
  - 2.5. Техника попарного тестирования.
  - 2.6. Техника тестирования переходов между состояниями.
  - 2.7. Техника тестирования сценариев использования.
- 3. Тестирование методом белого ящика.**
  - 3.1. Преимущества и недостатки тестирования методом белого ящика.
  - 3.2. Техника тестирования потоков управления. Критерии покрытия. Цикломатическая сложность и базовые пути графа потока управления.
  - 3.3. Метод базового пути для покрытия операторов и рёбер графа потока управления.
  - 3.4. Техника тестирования потоков данных.
- 4. Планирование тестирования.**
  - 4.1. Тестирование по сценарию и исследовательское тестирование, их преимущества и недостатки.
  - 4.2. Таксономии дефектов.
  - 4.3. Документы тестирования: план тестирования, спецификация дизайна тестов, спецификации тест-кейсов, спецификация процедуры тестирования, отчёт тестирования, отчёт об инцидентах.
- 5. Автоматизация тестирования.**
  - 5.1. Области применения автоматизации.
  - 5.2. Выгоды и риски автоматизации.
  - 5.3. Метрики тестирования. Инструментальные средства управления тестированием и отчётами о дефектах.
  - 5.4. Непрерывная интеграция и инструментарий для её поддержки.
- 6. Инструментальные средства тестирования.**
  - 6.1. Фреймворки модульного тестирования.
  - 6.2. Техники интеграционного и системного тестирования. Mock-объекты и заглушки, область их применения.
  - 6.3. Организация кода тестов и инструментальные средства её поддержки.
- 7. Методология разработки через тестирование (test-driven development, TDD).**
  - 7.1. Мотивация и область применимости методологии разработки через тестирование.
  - 7.2. Понятие о рефакторинге кода.
  - 7.3. Основной цикл разработки через тестирование «тест-код-рефакторинг».
- 8. Средства и методы тестирования приложений определённых типов.**
  - 8.1. Тестирование настольных приложений.
  - 8.2. Тестирование мобильных приложений.
  - 8.3. Тестирование веб-приложений.

## **5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные 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.

В процессе обучения используются следующие технологии электронного обучения и дистанционные образовательные технологии:

**Электронный учебный курс «Основы тестирования программного обеспечения» в LMS Электронный университет Moodle ЯрГУ**, в котором:

- предоставлены задания для самостоятельной работы обучающихся по темам дисциплины;
- осуществляется проведение отдельных мероприятий текущего контроля успеваемости студентов;
- представлены видеозаписи лекций по отдельным темам дисциплины;
- представлены правила прохождения промежуточной аттестации по дисциплине;
- представлен список учебной литературы, рекомендуемой для освоения дисциплины;
- посредством форума осуществляется взаимодействие между обучающимися и преподавателем в рамках изучения дисциплины.

## **6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине**

В процессе осуществления образовательного процесса по дисциплине используются:

для разработки документов, презентаций, для работы с электронными таблицами:

- офисный пакет LibreOffice;
- система вёрстки TeX;

для редактирования исходных текстов программного обеспечения:

- среда разработки программных проектов IntelliJ IDEA Community.

## **7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине**

В процессе осуществления образовательного процесса по дисциплине используются:

Автоматизированная библиотечно-информационная система «БУКИ-NEXT»  
[http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)

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

### **а) основная литература**

1. 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, 71 с.

### **б) дополнительная литература**

1. Бубнов А. А. Разработка и анализ требований к программному обеспечению: учебник для вузов. / А. А. Бубнов, С. А. Бубнов, К. А. Майков; Науч.-метод. совет РГРТУ - М.: КУРС, 2018. - 175 с.

Макконнелл, С., Совершенный код. Мастер-класс: пер. с англ / С. Макконнелл, М., Русская редакция, 2013, 868с

## **9. 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 групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- 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.

Автор:

Доцент кафедры  
компьютерных сетей, к.ф.-м.н.

*должность, ученая степень*

*подпись*

И. В. Парамонов

*И.О. Фамилия*

**Приложение №1 к рабочей программе дисциплины  
«Основы тестирования программного обеспечения»**

**Фонд оценочных средств  
для проведения текущего контроля успеваемости  
и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания и иные материалы,  
используемые в процессе текущего контроля успеваемости**

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

**Примеры заданий для лабораторных работ**  
*(проверка сформированности ПК-2, индикатор ИД-ПК-2\_4 в части знания  
методики тестирования программного обеспечения и умения разрабатывать тесты  
для программного обеспечения)*

**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 pull 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:

1. zip-архив, содержащий исправленный проект course-enrollment с разработанными тестами;
2. описание исправлений кода должно размещаться в текстовом файле fixes.txt в корне проекта;
3. описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;
4. файл build.gradle должен размещаться в корне архива без дополнительных подкаталогов;
5. 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 &lt;= 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 <code>@Before</code> 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:

1. zip-архив, содержащий исправленный проект course-enrollment с разработанными тестами;
2. описание исправлений кода должно размещаться в текстовом файле fixes.txt в корне проекта;
3. описание тест-кейсов с обоснованием их выбора в комментариях к соответствующему классу с тестами;
4. 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:

1. Чем отличаются процедуры написания тестов при использовании различных методов?
2. Чем отличаются сами тесты, написанные при использовании различных методов?
3. Какие ошибки легко могут быть найдены каждым из методов, а какие с трудом поддаются нахождению?
4. 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. Ожидается рассмотрение поставленных вопросов с точки зрения проделанной работы по тестированию класса и использование кода 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 joining	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

## 2. Список вопросов и (или) заданий для проведения промежуточной аттестации

### Список заданий к зачету:

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

19. Автоматизация тестирования. Выгоды и риски автоматизации. Области применения автоматизации.
20. Метрики тестирования.
21. Инструментальные средства управления тестированием и отчётами о дефектах.
22. Фреймворки модульного тестирования. Техники интеграционного и системного тестирования. Mock-объекты и заглушки, область их применения. Организация кода тестов и инструментальные средства её поддержки.
23. Методология разработки через тестирование (test-driven development, TDD).
24. Понятие о рефакторинге кода. Основной цикл разработки через тестирование «тест-код-рефакторинг».
25. Тестирование настольных приложений.
26. Тестирование мобильных приложений.
27. Тестирование веб-приложений.

#### Примерные вопросы теста

1. В состав отчёта о дефекте обычно входят следующие поля  
Выберите один или несколько ответов:
  - a. шаги по воспроизведению**
  - b. подробное описание**
  - c. чеклист
  - d. ссылка на номер требования
  - e. фамилия разработчика, внесшего 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
  - c. интеграционное тестирование
  - d. системное тестирование
  - e. 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**
  - c. 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:
  - a. интеграционное тестирование
  - b. модульное тестирование**
  - c. тестирование методом прозрачного ящика
  - d. тестирование методом чёрного ящика
  - e. 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.

### **Правила выставления оценки на зачёте**

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 «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 различных 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, при необходимости, проводятся консультации по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце семестра студенты сдают зачет. 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](http://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](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](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](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.