

**Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
"Обеспечение качества и тестирование"**

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчик программы
Кулямин В.В, к.ф.-м.н., kuliamin@ispras.ru

Одобрена на заседании департамента программной инженерии «___»_____ 2016 г.
Руководитель департамента Авдошин С.М. _____

Утверждена Академическим советом образовательной программы
«___»_____ 2016 г., № протокола _____

Академический руководитель образовательной программы
Шилов В.В. _____

Москва, 2016

Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.



1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов образовательной программы «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия», изучающих дисциплину «Обеспечение качества и тестирование».

Программа разработана в соответствии с образовательным стандартом Национального исследовательского университета «Высшая школа экономики» по направлению 09.03.04 «Программная инженерия».

2 Цели освоения дисциплины

Цель курса – Целью курса является изучение основных методов тестирования программного обеспечения (ПО) с упором на использование формализуемых моделей поведения и ситуаций в более широком контексте обеспечения качества ПО.

Задачами данного курса являются:

- освоение студентами базовых знаний о методах контроля качества программ;
- приобретение теоретических знаний в области тестирования ПО и практических навыков по использованию различных техник построения тестов и оценки их полноты;
- оказание помощи студентам в проведении собственных исследований и разработок в областях, использующих методы контроля качества ПО;
- приобретение навыков работы с некоторыми инструментами тестирования ПО и оценки тестового покрытия.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

1. Знать:

- основные понятия в области обеспечения качества ПО;
- основные методы моделирования поведения ПО и ситуаций, связанных с его работой;
- основные методы оценки полноты тестирования ПО;
- цели, задачи и основные методы разработки тестов и тестирования ПО;

2. Уметь:

- разрабатывать и реализовывать тесты для программ различных типов;
- оценивать полноту тестирования программ различных типов;
- анализировать результаты выполнения тестов;
- применять формализованные модели при разработке тестов и оценке их полноты;

3. Иметь навыки (приобрести опыт):

- в решении типовых задач разработки тестов для ПО;
- в анализе результатов выполнения тестов;
- в применении формализованных моделей для разработки тестов.

В результате освоения дисциплины студент осваивает следующие компетенции.

Компетенция	Код	Дескрипторы — основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
Способен решать проблемы в профессиональной деятельности	УК-3	Владеет	Лекционные и практические занятия, самостоя-



на основе анализа и синтеза			тельная работа
Способен оценивать потребность в ресурсах и планировать их использование при решении задач в профессиональной деятельности	УК-4	Владеет	Практические занятия, самостоятельная работа
Способен работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения научных и профессиональных задач (в том числе на основе системного подхода)	УК-5	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества	УК-6	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен использовать методы и инструментальные средства исследования объектов профессиональной деятельности	ПК-3	Применяет	Лекционные и практические занятия, самостоятельная работа
Способен обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности	ПК-4	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен проектировать, конструировать и тестировать программные продукты	ПК-10	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен читать, понимать и выделять главную идею прочитанного исходного кода, документации	ПК-11	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен использовать различные технологии разработки программного обеспечения	ПК-16	Владеет	Лекционные и практические занятия, самостоятельная работа
Способен применять основные методы и инструменты разработки программного обеспечения	ПК-17	Применяет	Лекционные и практические занятия, самостоятельная работа

4 Место дисциплины в структуре образовательной программы

Данная дисциплина относится к базовой части (блок Б.3.Б) дисциплин профессионального цикла.

Изучение данной дисциплины базируется на знаниях, полученных студентами при освоении учебных дисциплин:

- «Дискретная математика»,
- «Программирование»,
- «Построение и анализ алгоритмов»,



- «Архитектура вычислительных систем»,

5 Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Качество программного обеспечения и методы его контроля	6	2		2	8
2	Методы верификации ПО	12	4		4	16
3	Цели и задачи тестирования ПО	6	2		2	8
4	Организация тестовых наборов	6	2		2	10
5	Модели поведения ПО	12	4		4	16
6	Модели ситуаций и критерии полноты тестирования	12	4		4	16
7	Основные методы построения тестов. Вероятностные и нацеленные методы	6	4		4	12
8	Комбинаторные методы построения тестов	12	4		4	16
9	Автоматные методы построения тестов	12	4		4	16
10	Интегрированные технологии построения тестов	6	2		2	8
		190	32		32	126

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	3 год				Параметры
		1	2	3	4	
Текущий (неделя)	Контрольная работа		8			Письменная работа 45 минут
	Домашнее задание	2,4,6	2,4,6			
Итоговый	Экзамен		8			Письменный экзамен 120 минут

6.1 Критерии оценки знаний, навыков

Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

В рамках выполнения домашних заданий студент показывает освоение компетенций УК-4, УК-5, УК-6, ПК-3, ПК-4, ПК-10, ПК-16, ПК-17. На контрольной работе и экзамене демонстрируется освоение компетенций УК-3, УК-5, ПК-3, ПК-4, ПК-11, ПК-17.

Контрольная работа состоит из 8 заданий, правильное решение которых оценивается от 2 до 6 технических баллов, в зависимости от их сложности. Контрольная работа в целом оценивается в сумме от 0 до 30 технических баллов. Экзамен состоит из 4 задач, правильное решение которых оценивается от 8 до 10 технических баллов. Экзамен в целом оценивается от 0 до 34 технических баллов. Процент суммарно набранных баллов от максимально возможного количества определяет оценку за контрольную работу и за письменный экзамен.



Домашние задания оцениваются по аналогичной схеме.

Оценка	Набранные баллы (% от максимально возможных)
отлично (10)	более 90
отлично (9)	от 80 до 90 включительно
отлично (8)	от 70 до 80 включительно
хорошо (7)	от 60 до 70 включительно
хорошо (6)	от 50 до 60 включительно
удовлетворительно (5)	от 40 до 50 включительно
удовлетворительно (4)	от 30 до 40 включительно
неудовлетворительно (3)	от 20 до 30 включительно
неудовлетворительно (2)	от 10 до 20 включительно
неудовлетворительно (1)	не более 10

6.2 Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на практических занятиях и выполненные ими домашние задания. Оценка за работу студента на практических занятиях и в рамках домашних заданий выставляется в рабочую ведомость. Полученная оценка по 10-ти балльной шкале определяется перед итоговым контролем - $O_{\text{ауд}}$.

Преподаватель оценивает контрольную работу студентов. Оценка за контрольную работу студента выставляется в рабочую ведомость. Оценка по 10-ти балльной шкале за контрольную работу – $O_{\text{контр. работа}}$.

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{\text{накопл.}} = 0,5 * O_{\text{ауд}} + 0,5 * O_{\text{контр. работа}}$$

В диплом выставляет результирующая оценка по учебной дисциплине, которая формируется по следующей формуле:

$$O_{\text{результ}} = 0,5 * O_{\text{накопл}} + 0,5 * O_{\text{ЭКЗ}}$$

7 Содержание дисциплины

№ п/п	Разделы и темы лекционных занятий	Содержание
1	Качество программного обеспечения и методы его контроля	Качество ПО. Характеристики качества, стандарты ISO 9126 и ISO 25010. Задачи обеспечения качества ПО. Методы контроля качества ПО. Верификация и валидация. Методы верификации ПО: экспертиза, статический анализ, динамический анализ, формальные методы, синтетические методы верификации. Требования к ПО. Задачи извлечения и анализа требований. Необходимые свойства требований, стандарты IEEE 830 и IEEE 1233.
2	Методы верификации ПО	Классификация методов верификации ПО. Экспертиза. Статический анализ. Верификационный мониторинг. Формальные методы



		верификации. Дедуктивный анализ. Проверка моделей. Примеры применения различных методов верификации.
3	Цели и задачи тестирования ПО	Различные определения тестирования ПО. Тесты и тестовые наборы. Тестируемая система, тестируемая реализация, тестируемый интерфейс. Точки контроля и наблюдения. Роль требований. Модели поведения и модели ситуаций. Характеристики ситуаций. Цели тестирования в рамках жизненного цикла ПО. Задачи разработки тестов. Виды тестирования. Модульное, интеграционное и системное тестирование.
4	Организация тестовых наборов	Тестовые варианты. Основные паттерны организации тестовых наборов. Квалификаторы тестов. Связи между тестами и другими артефактами разработки. Основные модули тестов. Тестовые заглушки. Тестовые оракулы. Способы построения оракулов. Виды инструментов разработки тестов и тестирования. Семейство инструментов xUnit.
5	Модели поведения ПО	Логико-алгебраические и исполнимые модели. Абстрактные типы данных. Конечные автоматы. Расширенные конечные автоматы. Системы размеченных переходов. Использование разных видов моделей для описания функциональности.
6	Модели ситуаций и критерии полноты тестирования	Общая структура моделей ситуаций. Критерии полноты и критерии покрытия. Доменные, функциональные, структурные и проблемные критерии полноты. Использование графов, грамматик и логических выражений для построения критериев полноты тестирования. Типовые критерии покрытия кода: покрытие инструкций, ветвлений, покрытия потоков данных, покрытие условий, комбинаций условий, модифицированное покрытие условий и ветвлений (MC/DC). Критерии покрытия на интеграционном и системном уровнях.
7	Основные методы построения тестов Вероятностные и нацеленные методы	Вероятностные, нацеленные и комбинаторные методы тестирования. Области применимости различных методов. Вероятностные методы. Распределение рисков. Марковские цепи. Нацеленные методы. Метод функциональных диаграмм. Техника выбора экстремальных значений. Доменное тестирование. Тестирование с помощью сценариев. Часто встречающиеся ошибки для различных типов данных.
8	Комбинаторные методы построения тестов	Основные комбинаторные техники построения тестов. Дерево классификации. Использование грамматик. Покрывающие наборы. Алгоритмы построения покрывающих наборов.
9	Автоматные методы построения тестов	Задача тестирования на соответствие конечному автомату. Методы построения тестов для конечных полных детерминированных автоматов. D-метод. W-метод. Wp-метод. Построение тестов для расширенных и недетерминированных автоматов. Различные виды соответствия реализации и спецификации.
10	Интегрированные технологии построения тестов	Элементы технологии UniTESK. Использование программных контрактов для описания поведения. Редукция автомата по критерию полноты. Описание асинхронного поведения. Тестирование параллелизма и асинхронности на соответствие семантике линейности набора событий.



8 Оценочные средства для текущего контроля и аттестации студента

8.1 Тематика заданий текущего контроля

Текущий контроль проводится в форме выполнения домашних заданий и письменной контрольной работы.

Примерный вариант письменной контрольной работы для текущего контроля успеваемости.

1. К каким из характеристик качества по ISO 25010 относятся описанные требования? Перечислите номера соответствующих характеристик качества в первом столбце строки с описаниями требований в приведенной ниже таблице.

1. Функциональность
2. Надежность
3. Эффективность, производительность
4. Удобство использования
5. Переносимость
6. Удобство сопровождения
7. Совместимость
8. Защищенность

	Система должна работать 24 часа в сутки, 7 дней в неделю. Допустимый простой составляет 10 минут в год
	Код каждого класса должен быть снабжен комментариями, описывающими все задачи, решаемые его объектами
	Относительная погрешность не должна превосходить 10^{-15} при вызове одной функции, и 10^{-8} для всего вычисления с учетом накопления ошибок
	Доступ пользователя к счетам должен быть невозможен без аутентификации
	Пользователь, имеющий специальность «финансовая аналитика», должен осваивать основные возможности системы не более чем за один день
	Web-сайт должен поддерживать одновременную работу с 10000 пользователей

2. К каким из характеристик качества по ISO 25010 относятся описанные ошибки? Перечислите номера соответствующих характеристик качества (из предыдущего задания) в первом столбце строки с описанием ошибки в приведенной ниже таблице.

	Для выполнения наиболее частой операции — получения отчета за некоторый временной отрезок — требуется нажать не менее 8-ми кнопок на 4-х диалогах
	Если в системе работают больше двух пользователя одновременно, время реакции на их действия повышается до 40-60 с
	Система падает при вводе символа '#' в поле, где должно быть введено число
	Внесение первого же изменения, которое было затребовано пользователями, привело к полной перестройке архитектуры системы
	При работе одновременно с СУБД Oracle примерно 50% функций системы не выполняются — возникают исключения, которых не должно быть
	При вычислении логарифмов больших чисел относительная погрешность результатов вырастает до 50%



3. К какому виду относятся методы контроля качества, описанные в приведенных сценариях? Укажите соответствующий номер в первом столбце строки с описанием способа проверки в приведенной ниже таблице.

1. Экспертиза
2. Статический анализ
3. Формальная верификация
4. Тестирование
5. Мониторинг

	Группа, выполняющая проверку, формулирует набор сценариев возможных действий пользователей. Разработчики системы описывают работу системы в рамках этих сценариев и ее результаты на каждом шаге. Каждый проверяющий по ходу работы задает вопросы и составляет список замечаний и предложений по улучшению системы. После этого предложения и замечания совместно обсуждаются и сортируются по важности на три группы: критически важные, важные, несущественные. Замечания из первых двух групп принимаются к исправлению.
	Разработчики системы выделяют набор условий, при различных сочетаниях которых необходимо проверить ее правильность. Группа проверки определяет входные данные, которые необходимо использовать для выполнения каждого сочетания выделенных условий, при необходимости обращаясь с вопросами к разработчикам. Система выполняется с полученным набором данных на входе. Результаты ее работы фиксируются и потом подвергаются анализу на соответствие требованиям.
	Аналитик формулирует набор правил, которым должна подчиняться корректная работа системы. Проверка каждого правила и протоколирование ее результатов оформляются в виде отдельного модуля, который можно включить или отключить во время работы системы. Эксплуатация системы в течение первых 6 месяцев проходит с включенной проверкой. После этого накопленные результаты анализируются.
	Требования к системе оформляются в виде набора утверждений в одной из временных логик. Алгоритм работы системы формализуется в виде конечного автомата, быть может, с большим числом состояний. Этот конечный автомат и формализованные требования подаются на вход инструменту, автоматически ищущему сценарии работы автомата, при которых требования нарушаются. В результате выдается сценарий некорректной работы или сообщение о том, что ошибок в данном алгоритме найти не удалось.

4. Отметьте те из перечисленных пунктов, которые правильно описывают тесты.

	Установить систему, после чего подготовить список обнаруженных во время процедуры установки проблем.
	После запуска программы открыть форму «Реквизиты организации-клиента». Ввести название организации, состоящее из одних пробелов или символов табуляции. В остальные поля ввести данные реальной организации. Нажать кнопку «Сохранить». Должно появиться сообщение о неправильном названии организации.
	Запустить текстовый редактор и набрать 2-3 абзаца произвольного текста. Открыть диалог установки отступов. Задать нулевые отступы от левого и правого краев страницы. Сохранить введенные изменения. Отправить полученный документ на печать.
	Запустить калькулятор Windows. Набрать «4195835», затем «/», «3145727» и Enter. Если выдается ответ близкий к 1.33374, то процессор — Intel Pentium. На других типах процессоров будет ответ близкий к 1.33382.

5. Поставьте в последнем столбце приведенной ниже таблицы числа, соответствующие покрытию данного кода с помощью указанных в первом столбце тестовых данных по критерию, указанному во втором.



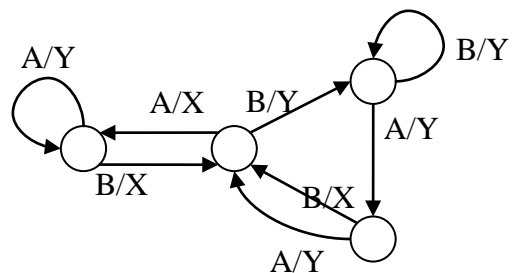
При вычислении покрытия строк пустые строки и строки, содержащие только заголовки метода, do, { или } не учитываются. При вычислении покрытия условий текстуально одинаковые условия считаются как одно, если их значения всегда совпадают, и как разные, если при каком-то сценарии работы метода их значения могут оказаться различны.

```
1  int gcd(int a, int b)
2  {
3      if(a == 0)
4          return b;
5      if(b == 0)
6          return a;
7      if(a == b)
8          return a;
9      if(a > 0 && b < 0 || a < 0 && b > 0)
10         b = -b;
11
12     do
13     {
14         if(b > a && a > 0 || b < a && a < 0)
15         {
16             a = b-a;
17             b = b-a;
18             a = a+b;
19         }
20
21         b = a-b;
22         a = a-b;
23     }
24     while(b != 0);
25
26     return a;
27 }
```

(0, 5)	Покрывает из 16 строк	
(0, 5)	Покрывает из 12 ветвлений	
(0, 5)	Покрывает из 20 условий	
(-2, 4)	Покрывает из 16 строк	
(-2, 4)	Покрывает из 12 ветвлений	
(-2, 4)	Покрывает из 20 условий	
(0, 5), (-2, 4)	Покрывает из 16 строк	
(0, 5), (-2, 4)	Покрывает из 12 ветвлений	
(0, 5), (-2, 4)	Покрывает из 20 условий	

6. Отметьте в первом столбце таблицы те последовательности, которые являются различающимися для изображенного автомата.

<input type="checkbox"/>	АВА
<input type="checkbox"/>	ААВ
<input type="checkbox"/>	ВВА
<input type="checkbox"/>	ВАВ





По окончании учебного периода проводится письменный экзамен.

Содержание возможных заданий на экзамене

1. Критерии покрытия кода: покрытие ветвлений, условий, комбинаций условий, модифицированное покрытие условий и ветвлений (MC/DC). Составление полного по критерию MC/DC тестового набора для метода.
2. Доменное тестирование с линейными ограничениями. Выбор тестовых данных для метода с помощью техники доменного тестирования.
3. Покрывающие наборы и методы их построения. Использование покрывающих наборов при создании тестов на основе грамматик.
4. Нахождение диагностирующей последовательности и характеризующего множества для конечного автомата. Построение полного набора тестов для конечно-автоматной модели с помощью W-метода и Wp-метода.

9 Учебно-методическое и информационное обеспечение дисциплины

9.1 Базовый учебник

9.2 Основная литература

1. Материалы курса
<http://mbt-course.narod.ru>
2. B. Beizer. Software Testing Techniques. 2-nd edition. Int. Thomson Publishing, 1990.
3. R. Binder. Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison-Wesley, 1999.
4. M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, A. Pretschner (eds.) Model Based Testing of Reactive Systems. LNCS 3472, Springer, 2005.
5. B. Marick. The Craft of Software Testing, Prentice Hall, 1995.
6. M. Utting, B. Legear. Practical Model-Based Testing: A Tools Approach, M-K, 2006.
7. Д. Месарош. Шаблоны тестирования xUnit. М.: Вильямс, 2008.
8. C. Kaner, R. L. Fiedler. Foundations of Software Testing. Context-Driven Press, 2013.

9.3 Дополнительная литература

1. Б. Бейзер. Тестирование черного ящика. СПб: Питер, 2004.
2. В. В. Липаев. Тестирование программ. М., Радио и связь, 1986.
3. В. В. Липаев. Методы обеспечения качества крупномасштабных программных средств. М., Синтег, 2003.
4. Г. Майерс. Искусство тестирования программ. М.: Финансы и статистика, 1982.
5. Г. Майерс. Надежность программного обеспечения. М.: Мир, 1980.
6. С. Канер, Дж. Фолк, Е. К. Нгуен. Тестирование программного обеспечения. М.: Диасофт, 2000.
7. A. P. Mathur. Foundations of Software Testing. Copymat Services, 2006.
8. Software Engineering Body of Knowledge, 2005.
http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf.
9. А. Коберн. Современные методы описания требований к системам. М.: Лори, 2002.
10. Д. Леффингуэлл, Д. Уидриг. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. М.: Вильямс, 2002.
11. В.В.Кулямин, Н.В.Пакулин, О.Л.Петренко, А.А.Сортов, А.В.Хорошилов. Формализация требований на практике. Препринт Института системного программирования РАН, №13, 2006.