

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.04 Программная инженерия

ФГОС введен в действие приказом №229 от 12.03.2015 г. , дата утверждения: 01.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.04 Программная инженерия

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета автоматики и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Романов Е. Л.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ПК.1 готовность применять основные методы и инструменты разработки программного обеспечения; в части следующих результатов обучения:	
у2. уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения	
у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	
у8. уметь выявлять ошибки в программном коде	
у9. уметь применять выбранные языки программирования для написания программного кода	
Компетенция ФГОС: ПК.11 владение особенностями эволюционной деятельности как с технической точки зрения, так и с точки зрения бизнеса (работа с унаследованными системами, возвратное проектирование, реинженеринг, миграция и рефакторинг; в части следующих результатов обучения:	
з1. знать методы и средства рефакторинга и оптимизации программного кода	
у1. уметь применять методы, средства для рефакторинга и оптимизации	
Компетенция ФГОС: ПК.21 владение навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации; в части следующих результатов обучения:	
з1. знать методы повышения читаемости программного кода	
Компетенция ФГОС: ПК.3 владение навыками использования различных технологий разработки программного обеспечения; в части следующих результатов обучения:	
з1. знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения	
у1. уметь применять методы и приемы отладки дефектного программного кода	
Компетенция ФГОС: ПК.7 владение методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения; в части следующих результатов обучения:	
з4. знать правила, алгоритмы и технологии создания тестовых наборов данных	
з7. знать методы создания и документирования контрольных примеров и тестовых наборов данных	
у10. уметь подготавливать наборы данных, используемые в процессе проверки работоспособности программного обеспечения	
у11. уметь разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками	
у12. уметь разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения	
у13. уметь использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования	
у14. уметь писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования	

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Конструирование и тестирование программного обеспечения</i>	
ПК.1.у4 уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	
1. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	Лабораторные работы; Самостоятельная работа
ПК.1.у8 уметь выявлять ошибки в программном коде	
2. уметь выявлять ошибки в программном коде	Практические занятия
ПК.1.у9 уметь применять выбранные языки программирования для написания программного кода	

3.уметь применять выбранные языки программирования для написания программного кода	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.3.31 знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения	
4.знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.3.y1 уметь применять методы и приемы отладки дефектного программного кода	
5.уметь применять методы и приемы отладки дефектного программного кода	Практические занятия; Самостоятельная работа
ПК.7.34 знать правила, алгоритмы и технологии создания тестовых наборов данных	
6.знать правила, алгоритмы и технологии создания тестовых наборов данных	Практические занятия; Самостоятельная работа
ПК.7.37 знать методы создания и документирования контрольных примеров и тестовых наборов данных	
7.знать методы создания и документирования контрольных примеров и тестовых наборов данных	Практические занятия; Самостоятельная работа
ПК.7.y10 уметь подготавливать наборы данных, используемые в процессе проверки работоспособности программного обеспечения	
8.уметь подготавливать наборы данных, используемые в процессе проверки работоспособности программного обеспечения	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.7.y11 уметь разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками	
9.уметь разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками	Практические занятия; Лабораторные работы; Самостоятельная работа
ПК.7.y12 уметь разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения	
10.уметь разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения	Практические занятия; Самостоятельная работа
ПК.7.y13 уметь использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования	
11.уметь использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования	Лабораторные работы; Самостоятельная работа
ПК.7.y14 уметь писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования	
12.уметь писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования	Практические занятия; Самостоятельная работа
ПК.11.31 знать методы и средства рефакторинга и оптимизации программного кода	
13.знать методы и средства рефакторинга и оптимизации программного кода	Практические занятия; Самостоятельная работа
ПК.11.y1 уметь применять методы, средства для рефакторинга и оптимизации	
14.уметь применять методы, средства для рефакторинга и оптимизации	Лабораторные работы; Самостоятельная работа
ПК.21.31 знать методы повышения читаемости программного кода	
15.знать методы повышения читаемости программного кода	Практические занятия; Самостоятельная работа
ПК.1.y2 уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения	

16. уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Практические занятия; Самостоятельная работа
---	---

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 7				
Дидактическая единица: Конструирование ПО				
1. Изучение IDE Andtoid Studio	0	4	1, 3, 4	Установка IDE и разработка простого приложения на Android
2. Разработка приложения с динамической разметкой	4	4	1, 14, 3, 4	Разработать приложение под Android с программируемой динамической разметкой и обработкой событий (по вариантам)
3. Ресурсы и межпрограммные коммуникации	4	6	1, 14, 3, 4	Доработать приложение из предыдущей лаб. работы, включив в него взаимодействие с другими приложениями через намерения, либо взаимодействие с ресурсами (аудио, видео, GPS)
Дидактическая единица: Тестирование ПО				
4. Разработка тестового покрытия	0	4	11, 8, 9	Для выбранной программы разработать тестовое покрытие структурного и функционального тестирования. Проверить на обнаружение вкрсимых дефектов.

Таблица 3.2

Темы практических занятий	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 7				
Дидактическая единица: Конструирование ПО				

1. Общие вопросы конструирования ПО. Автоматизация сборки и контроль версий	0	2	13, 15	Жизненный цикл проекта и программного продукта. Модели жизненного цикла. Каскадная, итерационная и спиральная модели. Технологические процессы жизненного цикла. Основные термины и определения. Тестирование на различных фазах жизненного цикла: тестирование требований, аналитических моделей, графического интерфейса, модулей, конфигураций. Свод знаний по программной инженерии SWEBOK о конструировании ПС. Основные характеристики, требования и техники управления программным кодом. Конструирование при разработке и сопровождении. Понимание. Рефакторинг. Реинжиниринг. Обратный инжиниринг. Совместное управление программным кодом. Системы контроля версий. Mercurial Автоматизация сборки. Maven.
2. Архитектура ОС Android.	0	2	4	Архитектура ОС Android. Ядро. Библиотеки. Среда исполнения (Dalvik-машина). Уровень каркаса приложений. Виды приложений (действия, сервисы, источники данных, приемники). Особенности компиляции приложений: файлы dx и apk. Система разработки для Android. Установка Android SDK и ее структура (эмулятор, библиотеки API, отладочный мост - adb, документация). Установка plugin-ов в NetBeans. Структура приложения. Файл AndroidManifest.xml. Описание компонент (действий, служб) и разрешений.

3. Проектирование разметки	0	2	4	Класс Activity. Классы элементов изображения и менеджеров компоновки View, Button, TextView, EditText, LinearLayout, TableLayout, TableRow. Статическое описание разметки в XML-файлах, доступ по идентификаторам ресурсов, назначение обработчиков событий. Динамическая (программная разметка). Диалоги с динамической (программной) и статической XML-разметкой
4. Жизненный цикл Activity и приложения. Управление ресурсами	0	2	4	Особенности выполнения приложений, связанные с ресурсными ограничениями. Класс Activity. Методы onCreate, onDestroy, onStart, onStop. Запуск Activity, передача данных и возврат результата от Activity. Сохранение данных при выгрузке. Меню. Контекстные меню. Всплывающие уведомления (Toast) и уведомления в строке состояния (Notification). Внутренний параллелизм в Android-приложениях. Особенности использования потоков. Класс Handler, использование сообщений для реализации параллелизма. Службы. Жизненный цикл. Взаимодействие с приложениями.
5. Практическое программирование в ОС Android. Примеры работы с ресурсами и сервисами	0	4	16, 3, 4	Обработка звука, изображения, работа с камерой. Намерения, межпрограммные коммуникации. Широковещательные сообщения между Activity и приложениями. GPS-навигатор и акселерометр. 2D-графика.
Дидактическая единица: Тестирование ПО				

6. Программная ошибка. Инспекция и отладка кода	0	1	Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок Обзор и основная классификация распространенных программных ошибок (дефектов), документирование и анализ ошибок. Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования.
--	----------	----------	---

7. Сущность, методология и виды тестирования	0	1	2	Свод знаний по программной инженерии SWEBOK о тестировании ПС. Тестирование на различных этапах жизненного цикла. Тестирование требований, прецедентов, архитектуры, моделей анализа. Сущность тестирования. . Экономическая сторона тестирования. Методологическая сторона тестирования. Невозможность исчерпывающего тестирования. Алгоритмически неразрешимые проблемы тестирования и инспекции. Понятие теста, тестового набора. Критерии выбора тестов: структурный (белый ящик), функциональный (черный ящик), стохастический, мутационный
8. Модульное тестирование. Структурное и функциональное тестирование	0	1	10, 6, 8, 9	Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.
9. Разработка, управляемая тестирование (TDD)	0	1	2, 5	Тестирование, совмещенное с проектированием (TDD-Test Driven Development). Технологический цикл. Паттерны тестирования. Приемы рефакторинга.

10. Модульное, интеграционное и системное тестирование. Тестирование при сопровождении. Автоматизация тестирования	0	1	12, 7, 8, 9	Модульное и интеграционное тестирование как фазы процесса конструирования. Средства автоматизации тестирования при конструировании (JUnit). Интеграционное тестирование. Технологии и метрики интеграционного тестирования. Интеграционное тестирование процедурно-ориентированных проектов. Интеграционное тестирование объектно-ориентированных проектов. Примеры. Системное тестирование. Виды: функциональное, ресурсное, стрессовое, тестирование производительности, устойчивости, конфигураций, документации. Тестирование при внесении изменений. Тестирование при сопровождении. Регрессионное тестирование. Виды. Управляемое регрессионное тестирование. Классификация тестов при отборе. Разновидности методов отбора тестов. Автоматизация тестирования. Сценарии тестирования. Скрипты. Индустриальное тестирование.
11. Тестирование GUI, мобильных и web-приложений приложений.	0	1	6, 8, 9	Тестирование GUI, мобильных и web-приложений приложений. Нагрузочное и стресс-тестирование web-приложений. Примеры.

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 7				
1	РГЗ	1, 10, 11, 12, 14, 16, 3, 5, 8, 9	33	2
Найти свободно распространяемую систему тестирования GUI для web- или мобильных приложений, произвести ее установку, разработать сценарии тестирования (контрольную задачу), инструкцию по установке и использованию, произвести аннотированный поиск материалов по практике использования, ошибках и проблемах.: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана.				
2	Подготовка к занятиям	13, 15, 4, 6, 7	20	0

Оформление результатов лаб.работ, самостоятельное изучение материала к практическим занятиям: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана.				
3	Подготовка к аттестации	13, 15, 4, 6, 7	10	5
: Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана.				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Социальные сети:YandexDisk vk.com/cprog_cs; Среда электронного обучения НГТУ:ИДО - дисц."Программная инженерия"
Размещение учебных материалов	Социальные сети:YandexDisk vk.com/cprog_cs; Среда электронного обучения НГТУ:ИДО - дисц."Программная инженерия"

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Максимальный балл
Семестр: 7	
<i>Лабораторная:</i>	40
Контролирующие материалы приводятся в "Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170 . - Загл. с экрана."	
<i>РГЗ:</i>	40
<i>Зачет №4:</i>	20
<i>Зачет №5:</i>	20

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита ЛР	Защита РГЗ	Зачет
ПК.1	у2. уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения			+
	у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	+		
	у8. уметь выявлять ошибки в программном коде	+		

	у9. уметь применять выбранные языки программирования для написания программного кода	+		
ПК.11	з1. знать методы и средства рефакторинга и оптимизации программного кода			+
	у1. уметь применять методы, средства для рефакторинга и оптимизации	+		
ПК.21	з1. знать методы повышения читаемости программного кода			+
ПК.3	з1. знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения			+
	у1. уметь применять методы и приемы отладки дефектного программного кода	+		
ПК.7	з4. знать правила, алгоритмы и технологии создания тестовых наборов данных			+
	з7. знать методы создания и документирования контрольных примеров и тестовых наборов данных			+
	у10. уметь подготавливать наборы данных, используемые в процессе проверки работоспособности программного обеспечения		+	+
	у11. уметь разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками			+
	у12. уметь разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения			+
	у13. уметь использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования		+	+
	у14. уметь писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования			+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Крылов Е. В. Техника разработки программ. В 2 кн.. Кн. 2 : [учебник для вузов по направлениям "Информатика и вычислительная техника" и "Техника и технологии"] / Е. В. Крылов, В. А. Острейковский, Н. Г. Типикин. - М., 2008. - 468, [1] с. : ил., табл.
2. Романов Е. Л. Архитектура и прикладные протоколы клиент-серверных приложений [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000226073. - Загл. с экрана.
3. Липаев В. В. Программная инженерия. Методологические основы : [учебник для вузов по направлению "Бизнес-информатика" (080700)] / В. В. Липаев ; Гос. ун-т - высш. шк. экономики. - М., 2006. - 605, [1] с.
4. Вязовик Н. А. Программирование на Java. Курс лекций / Н. А. Вязовик ; Интернет ун-т информ. технологий. - М., 2003. - 586 с. : ил., схемы
5. Murphy M. L. Beginning Android [electronic resource] // by Mark L. Murphy. - Berkeley, CA :, 2009. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-4302-2420-4>

Дополнительная литература

1. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем : [пер. с англ.] / Борис Бейзер. - СПб., 2004. - 317 с. : ил.

2. Липаев В. В. Тестирование программ / В. В. Липаев. - М., 1986. - 293, [2] с. : ил

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniium.com" : <http://znaniium.com/>
5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496. - Загл. с экрана.
2. Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.

8.2 Специализированное программное обеспечение

- 1 Microsoft Windows
- 2 Microsoft Office

9. Материально-техническое обеспечение

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Компьютеры объединены в локальную сеть с выходом в Internet

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Конструирование и тестирование программного обеспечения

Образовательная программа: 09.03.04 Программная инженерия, профиль: Технологии разработки программного обеспечения

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Конструирование и тестирование программного обеспечения приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ПК.1/ПТ готовность применять основные методы и инструменты разработки программного обеспечения	у2. уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Практическое программирование в ОС Android. Примеры работы с ресурсами и сервисами		Зачет, вопросы 4-9
ПК.1/ПТ	у4. уметь применять современные компиляторы, отладчики и оптимизаторы программного кода	Изучение IDE Andtoid Studio Разработка приложения с динамической разметкой Ресурсы и межпрограммные коммуникации	Отчет по лабораторной работе 1-3	
ПК.1/ПТ	у8. уметь выявлять ошибки в программном коде	Программная ошибка. Инспекция и отладка кода Разработка, управляемая тестирование (TDD) Сущность, методология и виды тестирования	Отчет по лабораторной работе 4	
ПК.1/ПТ	у9. уметь применять выбранные языки программирования для написания программного кода	Изучение IDE Andtoid Studio Практическое программирование в ОС Android. Примеры работы с ресурсами и сервисами Разработка приложения с динамической разметкой Ресурсы и межпрограммные коммуникации	Отчет по лабораторной работе 1-3	
ПК.11/СЭ владение особенностями эволюционной деятельности как с технической точки зрения, так и с точки зрения бизнеса (работа с унаследованными системами, возвратное проектирование, реинженеринг, миграция и рефакторинг	з1. знать методы и средства рефакторинга и оптимизации программного кода	Общие вопросы конструирования ПО. Автоматизация сборки и контроль версий		Зачет, вопросы 2-3
ПК.11/СЭ	у1. уметь применять методы, средства для рефакторинга и оптимизации	Разработка приложения с динамической разметкой Ресурсы и межпрограммные коммуникации	Отчет по лабораторной работе 2-3	

ПК.21/П владение навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	з1. знать методы повышения читаемости программного кода	Общие вопросы конструирования ПО. Автоматизация сборки и контроль версий		Зачет, вопросы 1-3
ПК.3/ПТ владение навыками использования различных технологий разработки программного обеспечения	з1. знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения	Архитектура ОС Android. Жизненный цикл Activity и приложения. Управление ресурсами Изучение IDE Andtoid Studio Практическое программирование в ОС Android. Примеры работы с ресурсами и сервисами Проектирование разметки Разработка приложения с динамической разметкой Ресурсы и межпрограммные коммуникации		Зачет, вопросы 4-9
ПК.3/ПТ	у1. уметь применять методы и приемы отладки дефектного программного кода	Программная ошибка. Инспекция и отладка кода Разработка, управляемая тестирование (TDD)	Отчет по лабораторной работе 4	
ПК.7/ОУ владение методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения	з4. знать правила, алгоритмы и технологии создания тестовых наборов данных	Модульное тестирование. Структурное и функциональное тестирование Тестирование GUI, мобильных и web-приложений приложений.		Зачет, вопросы 13-26
ПК.7/ОУ	з7. знать методы создания и документирования контрольных примеров и тестовых наборов данных	Модульное, интеграционное и системное тестирование. Тестирование при сопровождении. Автоматизация тестирования		Зачет, вопросы 13-26
ПК.7/ОУ	у10. уметь подготавливать наборы данных, используемые в процессе проверки работоспособности программного обеспечения	Модульное, интеграционное и системное тестирование. Тестирование при сопровождении. Автоматизация тестирования Модульное тестирование. Структурное и функциональное тестирование Разработка тестового покрытия Тестирование GUI, мобильных и web-приложений приложений.	РГЗ	Зачет, вопросы 13-26
ПК.7/ОУ	у11. уметь разрабатывать процедуры генерации тестовых наборов данных с заданными характеристиками	Разработка тестового покрытия		Зачет, вопросы 13-26

ПК.7/ОУ	у12. уметь разрабатывать и оформлять контрольные примеры для проверки работоспособности программного обеспечения	Модульное тестирование. Структурное и функциональное тестирование		Зачет, вопросы...
ПК.7/ОУ	у13. уметь использовать выбранную среду программирования для разработки процедур проверки работоспособности программного обеспечения на выбранном языке программирования	Разработка тестового покрытия	РГЗ	Зачет, вопросы...
ПК.7/ОУ	у14. уметь писать программный код процедур проверки работоспособности программного обеспечения на выбранном языке программирования	Модульное, интеграционное и системное тестирование. Тестирование при сопровождении. Автоматизация тестирования		Зачет, вопросы...

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 7 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ПК.1/ПТ, ПК.11/СЭ, ПК.21/П, ПК.3/ПТ, ПК.7/ОУ.

Зачет проводится в устной форме, по билетам

Форма проведения экзамена (зачета) описывается разработчиком самостоятельно.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 7 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ПК.1/ПТ, ПК.11/СЭ, ПК.21/П, ПК.3/ПТ, ПК.7/ОУ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований,

теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт зачета

по дисциплине «Конструирование и тестирование программного обеспечения», 7 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-13, второй вопрос из диапазона вопросов 14-26 (список вопросов приведен ниже). В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к зачету по дисциплине «Конструирование и тестирование программного
обеспечения»

Вопрос 1. Внутренний параллелизм в Android-приложениях. Особенности использования потоков. Класс Handler, использование сообщений для реализации параллелизма. Службы. Жизненный цикл. Взаимодействие с приложениями

Вопрос 2. Интеграционное тестирование. Технологии и метрики интеграционного тестирования. Интеграционное тестирование процедурно-ориентированных проектов

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) (дата)

Вопрос 1 Формальное представление задачи
Вопрос 2 Семантические сети, фреймы, сценарии.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) (дата)

2. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за зачет составляет 20 баллов. Соответственно, критерий оценки определяется в процентах к этому баллу:

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет менее 50% базовой
- Ответ засчитывается на **пороговом** уровне, если в теоретических вопросах даны только основные определения - оценка составляет не более 50% базовой
- Ответ засчитывается на **базовом** уровне, если в теоретических вопросах отражена структура вопроса (определения, свойства, правила) - оценка составляет 50-80% базовой
- Ответ засчитывается на **продвинутом** уровне, если дан развернутый ответ на теоретические вопросы и доп. вопросы - оценка составляет 80-100% базовой

3. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к зачету по дисциплине «Конструирование и тестирование программного обеспечения»

Список вопросов

1. Свод знаний по программной инженерии SWEBOK о конструировании ПС. Основные характеристики, требования и техники управления программным кодом. Конструирование при разработке и сопровождении. Понимание. Рефакторинг. Реинжиниринг. Обратный инжиниринг.
2. Совместное управление программным кодом. Системы контроля версий. Mercurial
3. Автоматизация сборки. Maven.
4. Архитектура ОС Android. Ядро. Библиотеки. Среда исполнения (Dalvik-машина). Уровень каркаса приложений. Виды приложений (действия, сервисы, источники данных, приемники). Особенности компиляции приложений: файлы dx и apk. Система разработки для Android.
5. Структура Android- приложения. Файл AndroidManifest.xml. Описание компонент (действий, служб) и разрешений
6. Класс Activity. Классы элементов изображения и менеджеров компоновки View, Button, TextView, EditText, LinearLayout, TableLayout, TableRow. Статическое описание разметки в XML-файлах, доступ по идентификаторам ресурсов, назначение обработчиков событий. Динамическая (программная разметка). Диалоги с динамической (программной) и статической XML-разметкой
7. Особенности выполнения приложений, связанные с ресурсными ограничениями. Класс Activity. Методы onCreate, onDestroy, onRestart, onStop. Запуск Activity, передача данных и возврат результата от Activity. Сохранение данных при выгрузке. Меню. Контекстные меню. Всплывающие уведомления (Toast) и уведомления в строке состояния (Notification).

8. Внутренний параллелизм в Android-приложениях. Особенности использования потоков. Класс `Handler`, использование сообщений для реализации параллелизма. Службы. Жизненный цикл. Взаимодействие с приложениями
9. Обработка звука, изображения, работа с камерой. Намерения, межпрограммные коммуникации. Широковещательные сообщения между `Activity` и приложениями. GPS-навигатор и акселерометр. 2D-графика
10. Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок
11. Обзор и основная классификация распространенных программных ошибок (дефектов), документирование и анализ ошибок. Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса.
12. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования.
13. Свод знаний по программной инженерии SWEBOOK о тестировании ПС. Тестирование на различных этапах жизненного цикла. Тестирование требований, прецедентов, архитектуры, моделей анализа.
14. Сущность тестирования. Экономическая сторона тестирования. Методологическая сторона тестирования. Невозможность исчерпывающего тестирования. Алгоритмически неразрешимые проблемы тестирования и инспекции. Понятие теста, тестового набора. Критерии выбора тестов: структурный (белый ящик), функциональный (черный ящик), стохастический, мутационный.
15. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример.
16. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.
17. Стохастическое и мутационное тестирование. Примеры.
18. Оценка уровня тестируемости проекта. Метрики и методики интегральной оценки.
19. Модульное и интеграционное тестирование как фазы процесса конструирования. Средства автоматизации тестирования при конструировании (JUnit).
20. Тестирование, совмещенное с проектированием (TDD-Test Driven Development). Технологический цикл. Паттерны тестирования. Приемы рефакторинга.
21. Интеграционное тестирование. Технологии и метрики интеграционного тестирования. Интеграционное тестирование процедурно-ориентированных проектов.
22. Интеграционное тестирование объектно-ориентированных проектов. Примеры.

23. Системное тестирование. Виды: функциональное, ресурсное, стрессовое, тестирование производительности, устойчивости, конфигураций, документации. Тестирование при внесении изменений.
24. Тестирование при сопровождении. Регрессионное тестирование. Виды. Управляемое регрессионное тестирование. Классификация тестов при отборе. Разновидности методов отбора тестов.
25. Автоматизация тестирования. Сценарии тестирования. Скрипты. Индустриальное тестирование.
26. Тестирование GUI и Useability

Паспорт расчетно-графического задания (работы)

по дисциплине «Конструирование и тестирование программного обеспечения», 7 семестр

1. Методика оценки

РГЗ выполняется в форме реферата по системе тестирования GUI для web- или мобильных приложений. Найти свободно распространяемую систему тестирования GUI для web- или мобильных приложений, произвести ее установку, разработать сценарии тестирования (контрольную задачу), инструкцию по установке и использованию, произвести аннотированный поиск материалов по практике использования, ошибках и проблемах.

2. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за РГР определен в рабочей программе. Соответственно, критерий оценки определяется в процентах к этому баллу:

1. Работа считается **не выполненной**, если выполнены не все части РГЗ(Р), оценка составляет менее 50% базовой
2. РГР засчитывается на **пороговом** уровне, содержание не полностью раскрывает задание, список источников обзора ограничен, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50% базовой
3. РГЗ засчитывается на **базовом** уровне, если содержание работы соответствует заданию, пояснительная записка оформлена в целом грамотно - оценка составляет 50-80% базовой
4. РГЗ засчитывается на **продвинутом** уровне, если содержание раскрыто полностью и разнообразно, количество источников обзора более 30, в пояснительной записке отражены все аспекты тематики, имеется аналитическая часть - оценка составляет 80-100% базовой

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ(Р)

- Фреймворк для тестирования UI Selenium
- Тестирование UI при помощи Espresso Test Recorder
- Tsung – распределённая система нагрузочного тестирования
- Технология тестирования программного обеспечения Фаззинг (Fuzzing)