



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Электронные вычислительные машины»

«СОГЛАСОВАНО»

Декан факультета ВТ
 Д.А. Перепелкин
«__» _____ 2020 г.

Заведующий кафедрой ЭВМ
 Б.В. Костров
«__» _____ 2020 г.

«УТВЕРЖДАЮ»

Проректор РГОИИМД
 А.В. Корячко
«__» _____ 2020 г.


РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.03.01 «Утилиты разработки программного обеспечения»

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль) подготовки

«Вычислительные машины, комплексы, системы и сети»

Уровень подготовки

Академический бакалавриат

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

Рязань 2020 г

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника (уровень бакалавриата), утвержденного приказом Минобрнауки России от 19.09.2017 г. № 929.

Программу составил

к.т.н., доц. кафедры

«Электронные вычислительные машины»



А.И. Ефимов

Программа рассмотрена и одобрена на заседании кафедры ЭВМ

«11» июня 2020 г., протокол № 10

Заведующий кафедрой

«Электронные вычислительные машины»,

д.т.н., проф. кафедры ЭВМ



Б.В. Костров

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы бакалавриата

Рабочая программа по дисциплине «Утилиты разработки программного обеспечения» является составной частью основной профессиональной образовательной программы (ОПОП) академического бакалавриата «Вычислительные машины, комплексы, системы и сети», разработанной в соответствии с ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника (уровень бакалавриата), утвержденным приказом Минобрнауки России от 19.09.2017 г. № 929.

Целью освоения дисциплины «Утилиты разработки программного обеспечения» является выработка базовых знаний в области поддержки программных проектов, навыков индивидуальной и коллективной разработки программного обеспечения на основе современных методов и стандартов версионирования, разработки, отладки и дальнейшего развития программных продуктов.

Задачи дисциплины:

- 1) дать обучающимся представление о принципах построения и архитектуры программного обеспечения;
- 2) дать представление о задаче контроля версий программного обеспечения;
- 3) дать представление об отслеживании ошибок в программных проектах;
- 4) дать представление о роли виртуализации в процессе разработки программного обеспечения;
- 5) дать представление об основных приемах автоматизации тестирования программных продуктов.

Перечень планируемых результатов обучения по дисциплине

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ПК-2	Способен проектировать и разрабатывать программное обеспечение	ПК-2.1. Знает принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения, баз данных и программных интерфейсов. ПК-2.2. Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных и программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами. ПК-2.3. Трудовые действия: разработка, изме-

		нение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения; проектирование структур данных, баз данных и программных интерфейсов; оценка и согласование сроков выполнения поставленных задач.
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Утилиты разработки программного обеспечения» является дисциплиной по выбору, относится к части основной профессиональной образовательной программы академического бакалавриата «Вычислительные машины, комплексы, системы и сети» по направлению подготовки 09.03.01 Информатика и вычислительная техника ФГБОУ ВО «РГРТУ», формируемой участниками образовательных отношений.

Дисциплина изучается по очной форме обучения на 1 курсе во 2 семестре.

Программа курса ориентирована на возможность расширения и углубления знаний, умений и навыков бакалавра для успешной профессиональной деятельности.

Постреквизиты дисциплины. Компетенции, полученные в результате освоения дисциплины необходимы обучающемуся при освоении следующих дисциплин: «Преддипломная практика», подготовки к государственной итоговой аттестации (подготовки и защиты выпускной квалификационной работы).

3. Объем дисциплины и виды учебной работы

Общая трудоемкость (объем) дисциплины составляет 3 зачетные единицы (ЗЕ), 108 часов.

Вид учебной работы	Всего часов
Общая трудоемкость дисциплины	108
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	32,25
лекции	16
практические занятия	16
лабораторные работы	-
консультации	-
иная контактная работа (промежуточная аттестация)	0,25
2. Самостоятельная работа обучающихся (всего), в том числе:	67
курсовой проект (работа)	-
иная самостоятельная работа	67
3. Контроль	8,75
Вид промежуточной аттестации обучающегося	зачет

4. Содержание дисциплины

В структурном отношении программа представлена следующими разделами:

Раздел 1. Основные понятия и цели дисциплины.

Раздел 2. Системы контроля версий.

Раздел 3. Система контроля версий Subversion.

Раздел 4. Система контроля версий GIT.

Раздел 5. Системы отслеживания ошибок, средства автоматизации тестирования.

Раздел 6. Виртуализация.

Раздел 7. Контейнеризация.

Раздел 8. Системы управления проектами.

4.1. Содержание дисциплины, структурированное по темам (разделам)

1. Основные понятия и цели дисциплины

Определение, основные задачи. Основные типы программных средств поддержки разработки ПО. Особенности совместной работы над программными проектами. Среда разработки ПО и их основные функции.

2. Системы контроля версий

Обзор. Терминология. Общие принципы работы. Централизованные и распределенные системы контроля версий. Обзор рынка систем контроля версий.

3. Система контроля версий Subversion

История создания. Структура хранилища данных. Дельта-кодирование. Рабочая копия. Протоколы доступа к хранилищу. Правки: создание, фиксация, смешивание. Справочная система. Идентификация правок: числовые идентификаторы, ключевые слова, даты. Создание рабочей копии. Внесение изменений в рабочую копию. Анализ изменений. Обновление рабочей копии. Публикация изменений рабочей копии в хранилище. Решение конфликтов. Ветвление в Subversion. Использование веток. Создание ветки. Работа с веткой. Копирование изменений между ветками. Копирование отдельных изменений. Слияние веток.

4. Система контроля версий GIT

Основы работы в GIT. Ветвление, как основа системы контроля версий, достоинства и недостатки. Слешки. Области хранения файлов. Создание и клонирование репозитория. Игнорирование файлов. Жизненный цикл файлов в GIT. Работа с удаленными репозиториями.

5. Системы отслеживания ошибок, средства автоматизации тестирования

Отслеживание ошибок, как неотъемлемая часть разработки и сопровождения программного обеспечения. Классификация программных ошибок. Основные атрибуты отчета об ошибке. Жизненный цикл ошибки. Рынок систем отслеживания ошибок. Интеграция в среды разработки ПО. Системы отслеживания ошибок.

Основные понятия о тестировании программного обеспечения. Пирамида автоматизации тестирования. Особенности написания unit-тестов. Тестовый фреймворк NestNG.

6. Виртуализация

Принципы виртуализации. Преимущества виртуализации. Типы виртуализации: программная и аппаратная. Виртуализация машин и серверов. Компоненты виртуальной инфраструктуры. Облачные технологии. Обзор рынка виртуальных машин.

7. Контейнеризация

Общие сведения и назначение. Отличие от подходов виртуализации. Преимущества и области применения.

8. Системы управления проектами

Общие принципы построения архитектуры программного обеспечения. Архитектурные стили. Основные функции систем управления проектами. Преимущества использования систем управления проектами. Функции менеджера проекта. Слежение за проектом. Обзор систем управления проектами. Microsoft Project. Jira. Enterprise Architect.

4.2. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

Очная форма обучения

№ п/п	Тема (раздел)	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль (подготовка к зачету)
			Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
1	2	3	4	5	6	7	8	9	10	11
1	Раздел 1. Основные понятия и цели дисциплины	8	2	2	0				6	
2	Раздел 2. Системы контроля версий	8	2	2	0				6	
3	Раздел 3. Система контроля версий Subversion	12	6	2	4				6	
4	Раздел 4. Система контроля версий GIT	13	6	2	4				7	
5	Раздел 5. Системы отслеживания ошибок, средства автоматизации тестирования	17	10	2	8				7	
6	Раздел 6. Виртуализация	13	2	2	0				11	
7	Раздел 7. Контейнеризация	12	2	2	0				10	
8	Раздел 8. Системы управления проектами	16	2	2	0				14	
	Подготовка к промежуточной аттестации	8,75								8,75
	Промежуточная аттестация	0,25	0,25					0,25		
	Итого	108	32,25	16	16			0,25	67	8,75

Виды практических и самостоятельных работ

Очная форма обучения

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Основные понятия и цели дисциплины	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме	6
Системы контроля версий	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме	6
Система контроля версий Subversion	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме Подготовка к практическому занятию	3 3
	Практическое занятие	Изучение Subversion. Основные операции. Создание репозитория, рабочей копии, разрешение конфликтов обновления. Ветвление в Subversion. Разрешение конфликтов слияния	4

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Система контроля версий GIT	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме Подготовка к практическому занятию	4 3
	Практическое занятие	Изучение Git. Основные операции. Создание и настройка локального репозитория, разрешение конфликтов обновления. Ветвление в Git. Разрешение конфликтов слияния	4
Системы отслеживания ошибок, средства автоматизации тестирования	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме Подготовка к практическому занятию	4 3
	Практическое занятие	Изучение систем отслеживания ошибок на примере GitLab. Изучение подходов по автоматизации тестирования и тестового фреймворка TestNG	4 4
Виртуализация	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме.	7
		Изучение концепции и видов виртуализации. Изучение виртуализации ресурсов и виртуализации платформ	4
Контейнеризация	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме.	6
		Изучение подходов контейнеризации, контейнеров Docker	4
Системы управления проектами	Самостоятельная работа	Изучение конспекта лекций и рекомендованной литературы по теме.	6
		Изучение систем управления проектами при коллективной разработке программного обеспечения. Знакомство с системой управления проектами Jira.	8
Подготовка к промежуточной аттестации		Изучение конспекта лекций и рекомендованной литературы	8,75

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

1. Зубкова, Т. М. Технология разработки программного обеспечения [Электронный ресурс]: учебное пособие / Т. М. Зубкова. — Электрон. текстовые данные. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2017. — 469 с. — 978-5-7410-1785-2. — Режим доступа: <http://www.iprbookshop.ru/78846.html>

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств приведен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Инструментальные средства разработки программного обеспечения»).

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная учебная литература

1. Вендров А.М. Современные технологии создания программного обеспечения [Электронный ресурс] / «CitForum». URL: <http://citforum.ru/programming/application/program/> (дата обращения: 01.12.2014).
2. Липаев В.В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс]: учебное пособие/ Липаев В.В.- Электрон. текстовые данные.- М.: МАКС Пресс, 2014.- 309 с.- Режим доступа: <http://www.iprbookshop.ru/27297.-> ЭБС "IPRbooks" (дата обращения: 21.06.2016).
3. Алексеев В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений [Электронный ресурс]/ Алексеев В.Е., Таланов В.А.- Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.- 153 с.- Режим доступа: <http://www.iprbookshop.ru/52186.-> ЭБС "IPRbooks" (дата обращения: 21.06.2016).

Дополнительная учебная литература

1. Волкова Т.В. Разработка систем распределенной обработки данных [Электронный ресурс]: учебно-методическое пособие/ Волкова Т.В., Насейкина Л.Ф.- Электрон. текстовые данные.- Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2012.- 330 с.- Режим доступа: <http://www.iprbookshop.ru/30127.-> ЭБС "IPRbooks", по паролю (дата обращения: 21.06.2019).
2. Алексеев, В. А. Паттерны проектирования программных систем [Электронный ресурс] : методические указания к проведению лабораторных работ по курсу «Архитектура программных систем» / В. А. Алексеев. — Электрон. текстовые данные. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2016. — 33 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/74412.html>

8. Перечень ресурсов информационно–телекоммуникационной сети Интернет, необходимых для освоения дисциплины

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

9. Методические указания для обучающихся по освоению дисциплины

Указания в рамках лекций

Во время лекции студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающимся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Указания в рамках подготовки к промежуточной аттестации

При подготовке к экзамену в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала предусмотренного рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендован-

ной к настоящей рабочей программе. При подготовке к зачету нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по несколько типовых задач из каждой темы (в том случае если тема предусматривает решение задач). При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

Указания в рамках лабораторных работ

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Они направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений.

- Выполнение студентами лабораторных работ направлено на следующие цели:
- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплин;
- формирование необходимых профессиональных умений и навыков.

Выполнению лабораторной работы предшествует проверка знаний студентов – их теоретической готовности к выполнению задания.

Помимо выполнения работы для каждой лабораторной работы предусмотрена процедура защиты, в ходе которой преподаватель проводит устный или письменный опрос студентов для контроля понимания и правильной интерпретации полученных результатов и усвоения ими основных теоретических и практических знаний по теме лабораторной работы.

Указания в рамках самостоятельной работы студентов

Самостоятельная работа студентов по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения самостоятельной работы студентов готовятся преподавателем и выдаются студентам в виде раздаточных материалов или оформляются в виде электронного ресурса используемого в рамках системы дистанционного обучения ФГБОУ ВО «РГРТУ».

Самостоятельное изучение тем учебной дисциплины способствует:

- закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий;
- углублению и расширению знаний по отдельным вопросам и темам дисциплины;
- освоению умений прикладного и практического использования полученных знаний.

Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучается дополнительная рекомендованная литература. Литературу по курсу рекомендуется изучать в библиотеке, с использованием доступной электронной библиотечной системы или с помощью сети Интернет (источники, которые могут быть использованы без нарушения авторских прав).

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Для выполнения практических занятий требуется рабочее место, оборудованное письменным столом.

Перечень лицензионного программного обеспечения:

- 1) Операционная система Windows XP (не ниже) Professional (лицензия Microsoft DreamSpark Membership ID 700102019);
- 2) Open Office (лицензия Apache License, Version 2.0);
- 3) Git (лицензия GNU GPL 2);
- 4) Subversion (лицензия Apache License 2.0).

Перечень профессиональных баз данных (в том числе международным реферативным базам данных научных изданий) и информационных справочных систем:

1) Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ (дата обращения 02.02.2017).

2) Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни — 24.00, выходные и праздничные дни – круглосуточно) (дата обращения 02.02.2017).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для освоения дисциплины необходимы:

1) для проведения лекционных занятий необходима аудитория с достаточным количеством посадочных мест, соответствующая необходимым противопожарным нормам и санитарно-гигиеническим требованиям, для проведения лекций аудитория должна быть оснащена проекционным оборудованием;

2) для проведения лабораторных работ и практических занятий необходим класс персональных компьютеров с установленными операционными системами Microsoft Windows XP (или выше) и установленным лицензионным программным обеспечением Open Office, Subversion, Git.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Электронные вычислительные машины»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Б1.В.ДВ.03.01 «Утилиты разработки программного обеспечения»

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль) подготовки

«Вычислительные машины, комплексы, системы и сети»

Уровень подготовки

Академический бакалавриат

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

Рязань 2020 г

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, закреплённых за дисциплиной.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета.

Форма проведения зачета – тестирование, письменный опрос по теоретическим вопросам.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (зачет) выносятся тест, два теоретических вопроса. Максимально студент может набрать 6 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено», «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 4 баллов (выполнил одно задание на эталонном уровне, другое – не ниже порогового, либо оба задания выполнит на продвинутом уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических и лабораторных работ заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 4 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические, либо лабораторные работы.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Раздел 1. Основные понятия и цели дисциплины	ПК-2	Зачет
Раздел 2. Системы контроля версий	ПК-2	Зачет
Раздел 3. Система контроля версий Subversion	ПК-2	Зачет
Раздел 4. Система контроля версий GIT	ПК-2	Зачет
Раздел 5. Системы отслеживания ошибок, средства автоматизации тестирования	ПК-2	Зачет
Раздел 6. Виртуализация	ПК-2	Зачет
Раздел 7. Контейнеризация	ПК-2	Зачет
Раздел 8. Системы управления проектами	ПК-2	Зачет

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-2	Способен проектировать и разрабатывать программное обеспечение

1. Что такое система контроля версий? (выберите один вариант ответа)

- а) Программное обеспечение, призванное избавить программиста от необходимости загружать проект к себе на машину
- б) Программное обеспечение, призванное автоматизировать работу с историей файла, и организовать защищенное хранилище проекта
- в) Программное обеспечение, позволяющее версионировать проект

2. Что такое репозиторий? (выберите один вариант ответа)

- а) Место, где система контроля версий хранит все документы вместе с их историей
- б) Директория для вашего проекта
- в) Рабочая копия документов

3. Соедините описание системы контроля версий с её названием (соединить)

Локальные	СКВ, в которых вся работа производится с центральным хранилищем (все действия, так или иначе, зависят от него)
Централизованные	СКВ, главной парадигмой которых является локализация данных на машине каждого разработчика проекта
Децентрализованные	СКВ, хранящиеся только на локальном компьютере

4. Объедините СКВ с её типом (соединить)

GIT	Локальная
SVN	Распределённая и централизованная
Mercurial	Централизованная
CVS	Децентрализованная
Bazaar	Распределённая и локальная

5. К какому виду систем контроля версий относится Subversion? (выберите один вариант ответа)

- а) Централизованному
- б) Распределённому
- в) Локальному
- г) Всем вышеперечисленным

6. Соотнесите состояние хранилища и то, какие действия предпримет Subversion во время коммита (соединить)

Хранилище изменялось локально и не устарело	Система не сделает ничего, вам необходимо внести изменения и сделать коммит
---------------------------------------------	-----------------------------------------------------------------------------

Хранилище не изменялось и устарело	Никаких
Хранилище изменялось локально и устарело	Заберет изменения с сервера
Хранилище не изменялось и не устарело	Система забирает к себе изменения с сервера и пользователю нужно разрешить конфликты, если необходимо

7. *Соотнесите модель с её описанием (соединить)*

Блокирование - Изменение - Разблокирование	Оба пользователя копируют файл и редактируют его на своём компьютере. Затем один из пользователей отправляет изменения на сервер. Второй пользователь заканчивает свои изменения и тоже хочет отправить файлы на сервер, но не может, т.к. там уже находится обновленная версия. Он должен сначала взять изменения с сервера, объединить изменения, разрешить конфликты, если они есть, и затем отправить свои изменения на сервер.
Копирование — Изменение - Слияние	Пользователь, решивший редактировать файл, блокирует его, в то время другой пользователь не может его редактировать. При разблокировке файла следующим участником другой пользователь может забрать изменения предыдущего пользователя и редактировать файл дальше.

8. *Отметьте, какие команды вы можете использовать для получения подсказок в Subversion (выберите несколько вариантов ответа)*

- а) svn help
- б) svn resolve
- в) svn commit
- г) svn update
- д) svn log
- е) svn info
- ж) svn revert
- з) svn delete

9. *Какие команды вы можете использовать для создания/получения репозитория? (выберите несколько вариантов ответа)*

- а) svnadmin create /path/to/rep

б) `svn import mytree file:///usr/local/svn/newrepos/some/project \`
`-m "Initial import"`

в) `svn checkout http://svn.example.com/repos/calc`

10. Отметьте команды для внесения изменений в рабочую копию (выберите несколько вариантов ответа)

а) `svn add`

б) `svn delete`

в) `svn copy`

г) `svn move`

д) `svn mkdir`

е) `svn update`

11. Объедините способы разрешения конфликтов с их описанием (соединить)

1) Объединить конфликтный файл:

Конфликтный файл будет выглядеть так:

<<<<<<< имя файла

ваши изменения

=====
результат автоматического слияния с ре-

позиторием

>>>>>>> ревизия

2) Выполнить "`svn resolved`"

Отмена ваших изменений

1) Выполнить "`svn revert file`"

Объединение вручную

Скопировать `filename.rOLDREV` или
`filename.rNEWREV`

Копирование одного из файлов (своего
или чужого)

12. Вопрос: верно ли, что в *Git* и *Subversion* используются принципиально разные подходы к хранению файлов репозитория? (выберите один вариант ответа)

а) да

б) нет

13. Вопрос: каковы были основные цели, преследуемые при создании *Git*? (выберите один вариант ответа)

а)

1. Простая архитектура.
2. Полная децентрализация.
3. Хорошая поддержка нелинейной разработки.

б)

1. Высокая скорость работы.
2. Применение подхода CVS.
3. Поддержка нелинейной разработки.

в)

1. Простая архитектура.
2. Централизованный подход.
3. Поддержка нелинейной разработки.

14. Под какой лицензией выпущена системы контроля версий Git? (краткий ответ)

15. Какой алгоритм использует Git для вычисления хэш-сумм? (выберите один вариант ответа)

- а) CRC32
- б) MD6
- в) SHA-1
- г) SHA-2

16. Что такое ветка в Subversion? (выберите один вариант ответа)

- а) Направление разработки, которое существует независимо от другого направления
- б) Направление разработки, которое существует независимо от другого направления, однако имеющие с ним общую историю
- в) Копия репозитория у любого из разработчиков

17. В каких случаях есть необходимость создать ветку? (выберите несколько вариантов ответа)

- а) Изменения, которые вы хотите внести, могут повредить работающему коду
- б) Вы хотите написать улучшение/оптимизировать существующий код
- в) Вы хотите зафиксировать изменения

18. Выберите верные сведения о создании ветвей в Subversion (выберите несколько вариантов ответа)

- а) При использовании команды `svn copy <удаленный url> <удаленный url> -m "Сообщение коммита"`, вы создаёте ветку на удалённом сервере, а не на локальной машине

- б) При использовании команды `svn copy <удаленный url> <удаленный url> -m "Сообщение коммита"`, вы создаёте ветку на локальной машине
- в) При использовании команды `svn copy <удаленный url> <удаленный url> -m "Сообщение коммита"`, вам не обязательно иметь рабочую копию
- г) При использовании команды `svn copy <локальная директория> <локальная директория>` вы создаёте ветку на локальной машине
- д) При использовании команды `svn copy <локальная директория> <локальная директория>` вы создаёте ветку на удаленном сервере, а не на локальной машине

19. При создании ветки в Git командой `git branch <branchname>`, происходит ли автоматический переход на эту ветку? (выберите один вариант ответа)

- а) да
- б) нет

20. Что происходит при команде `git checkout <branchname>` (выберите один вариант ответа)

- а) Мы переходим в директорию `branchname`
- б) Указатель HEAD перемещается на ветку `branchname`
- в) Мы создаём ветку `branchname` и перемещаемся на неё

21. По каким протоколам можно настроить работу Git на сервере? (выберите несколько вариантов ответа)

- а) HTTP
- б) FTP
- в) SSH
- г) Telnet
- д) Git
- е) SMTP

22. Что вам нужно сделать с проектом в случае «вы хотите внести изменения в репозиторий, в который у вас нет доступа»? (выберите один вариант ответа)

- а) fork
- б) pull request

23. Выберите случаи, когда вам может потребоваться использовать GitHub (выберите несколько вариантов ответа)

- а) создать свой проект с открытым исходным кодом / использовать как хранилище кода, который не хотелось бы потерять
- б) воспользоваться сторонней библиотекой (не входящей в список стандартных)
- в) внести вклад в уже существующий проект на GitHub

24. *Что такое система отслеживания ошибок?* (выберите один вариант ответа)

- а) Прикладная программа, которая ищет ошибки в вашем коде, и следит за этими участками
- б) Прикладная программа, разработанная с целью помочь разработчикам программного обеспечения учитывать и контролировать ошибки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнением или невыполнением пожеланий

25. *Что такое «баг» (в программировании)?* (выберите один вариант ответа)

- а) Жук, который забрался в системный блок компьютера
- б) Ошибка в программе, которая выдаёт неожиданный или неправильный результат
- в) Программа, которая пишется, чтобы найти ошибки в программе

26. На каком этапе программы (или кем) могут быть обнаружены «баги»? (выберите несколько вариантов ответа)

- а) В процессе тестирования программы
- б) В процессе отладки программы
- в) На этапе проектирования программы
- г) Сторонними пользователями приложения

27. Расставьте в нужном порядке последовательность действий при обнаружении ошибки технического характера (порядок)

- а) Сообщение об обнаруженной ошибке технической поддержке приложения
- б) Взятие ошибки в обработку членом команды разработки
- в) Обнаружение ошибки пользователем
- г) Исправление ошибки (закончено)
- д) Выпуск патча (исправления) к существующему приложению
- е) Занесение ошибки в систему отслеживания ошибок
- ж) Работа над исправлением ошибки (члена команды разработки)
- з) Смена статуса ошибки в системе отслеживания ошибок на «в работе»
- и) Смена статуса ошибки в системе отслеживания ошибок на «закрыта»

28. Какие сведения об дефекте могут храниться в системе отслеживания ошибок? (выберите несколько вариантов ответа)

- а) Номер (идентификатор) дефекта
- б) Кто сообщил о дефекте
- в) Версия продукта, в которой обнаружен дефект
- г) Обсуждение того, кто возьмет задачу по устранению
- д) Серьёзность (критичность) дефекта и приоритет решения
- е) Описание шагов для выявления дефекта (воспроизведения неправильного поведения программы)
- ж) Кто ответственен за устранение дефекта
- з) Обсуждение возможных решений и их последствий

- и) Текущее состояние (статус) дефекта
- к) Версия продукта, в которой дефект исправлен

29. В каком состоянии может быть дефект? (выберите несколько вариантов ответа)

- а) Открыт
- б) Назначен
- в) Некому назначить
- г) Проверен
- д) Отклонен
- е) Закрыт

30. Что такое тестирование? (выберите несколько вариантов ответа)

- а) Одна из техник контроля качества, включающая в себя активности по планированию работ, проектированию тестов, выполнению тестирования и анализу полученных результатов
- б) Работа тестировщика
- в) Проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом

31. Выберите виды тестов, входящие в пирамиду тестирования (Майк Кон) (выберите несколько вариантов ответа)

- а) юнит-тесты
- б) интеграционные тесты
- в) тесты корректности работы операционной системы, на которой будет установлено приложение
- г) тесты пользовательского интерфейса
- д) тесты правильности сборки аппаратной платформы компьютера

32. Выберите положения, правильные для юнит тестирования (выберите несколько вариантов ответа)

- а) Должны не зависеть от окружения, на котором они выполняются
- б) Запускаться регулярно в автоматическом режиме
- в) Должны выполняться под специально настроенным окружением
- г) Должны запускаться вручную для контроля за ними

33. Соедините методологию тестирования и ее описание (соединить)

Тестирование черного ящика	метод тестирования программного обеспечения, который предполагает, что внутренняя устройство системы известны тестировщику
Тестирование методом серого ящика	метод тестирования, базируется только лишь на тестировании по функциональной спецификации и требованиям, при этом не имея доступа во внутреннюю структуру кода и базу данных.
Тестирование методом белого ящика	Метод тестирования ПО, который предполагает, что внутреннее устройство программы нам известно лишь частично

34. Расставьте в нужном порядке последовательность действий при разработке через тестирование (порядок)

- а) Пишется тест, покрывающие желаемое изменение
- б) Проводится рефакторинг нового кода к соответствующим стандартам
- в) Пишется код, который позволит пройти тест

35. Соедините вид теста и то, что он проверяет (соединить)

Интеграционные тесты	Тестирование каждой нетривиальной функции или метода
Юнит-тесты	Тестирование корректного взаимодействия с другими приложениями
Тесты пользовательского интерфейса	Тестирование правильности работы пользовательского интерфейса приложения

36. Какие функции должна выполнять служба CI? (выберите несколько вариантов ответа)

- а) получение исходного кода из репозитория
 - б) сборка проекта (компиляция)
 - в) выполнение тестов
 - г) развёртывание готового проекта
 - д) отправка отчетов (на электронную почту)
 - е) последовательное выведение релиза в промышленную среду
-

37. Как вы помните, CD включает в себя и CI. Напишите кратко, какую главную функцию выполняет CD, но не выполняет CI (краткий ответ)

38. Какие задачи входят в обязанности Dev-Ops инженера? (выберите несколько вариантов ответа)

- а) Системное администрирование
- б) Тестирование кода разработчиков
- в) Программирование
- г) Использование облачных технологий
- д) Автоматизация крупной инфраструктуры

39. Соедините подкатегорию виртуализации с ее описанием (соединить)

виртуализация платформ

Данный вид виртуализации преследует своей целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей и т.п

виртуализация ресурсов

Продуктом этого вида виртуализации являются виртуальные машины - некие программные абстракции, запускаемые на платформе реальных аппаратно-программных систем

40. Какие подвиды включает виртуализация платформ? (выберите несколько вариантов ответа)

- а) Разделение ресурсов
- б) Инкапсуляция
- в) Частичная виртуализация
- г) Паравиртуализация
- д) Виртуализация уровня операционной системы
- е) Полная эмуляция (симуляция).
- ж) Частичная эмуляция (нативная виртуализация).
- з) Виртуализация уровня приложений

41. Какие подвиды включает виртуализация ресурсов? (выберите несколько вариантов ответа)

- а) Разделение ресурсов
- б) Инкапсуляция
- в) Частичная виртуализация
- г) Паравиртуализация
- д) Виртуализация уровня операционной системы
- е) Кластеризация компьютеров и распределенные вычисления (grid computing)
- ж) Частичная эмуляция (нативная виртуализация).
- з) Виртуализация уровня приложений
- и) Объединение, агрегация и концентрация компонентов

42. Что такое контейнеризация? (выберите один вариант ответа)

- а) Подход к разработке программного обеспечения, при котором приложение или служба, их зависимости и конфигурация упаковываются вместе в образ контейнера
- б) Абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию
- в) Набор изолированных приложений, не взаимодействующих друг с другом

43. Преимущества контейнеризации перед виртуализацией (выберите несколько вариантов ответа)

- а) Более простая настройка контейнеров
- б) Контейнеры требуют гораздо меньше ресурсов
- в) Контейнеры легко развертывать
- г) Контейнеры быстрее запускаются
- д) Приложение в контейнере выполняется в любой среде

44. Соедините термин с его определением (соединить)

Контейнер	пакет со всеми зависимостями и сведениями, необходимыми для создания контейнера
Образ контейнера	действие по созданию образа контейнера на основе сведений и контекста, предоставленных файлом Dockerfile, а также дополнительных файлов в папке, где создается образ

Сборка	текстовый файл, содержащий инструкции по сборке образа Docker
Dockerfile	экземпляр образа Docker

45. Какие компоненты входят в каждый контейнер Docker? (выберите несколько вариантов ответа)

- а) Выбранная операционная система (например, дистрибутив Linux, Windows Nano Server или Windows Server Core)
- б) Файлы, добавленные разработчиком (двоичные файлы приложения и т. п.)
- в) Сведения о конфигурации (параметры среды и зависимости)
- г) Тесты для приложения

Типовые теоретические вопросы:

1. Системы управления версиями. Основные понятия.
2. Системы управления версиями. Сущность, назначение, преимущества.
3. Системы управления версиями. Виды и особенности.
4. Системы управления версиями. Основные представители на рынке.
5. Системы управления версиями. Модели версионирования. Достоинства и недостатки.
6. Subversion. Общие сведения, история создания.
7. Subversion. Основные понятия.
8. Subversion. Особенности хранения изменения и организации хранилища.
9. Subversion. Жизненный цикл проекта.
10. Subversion. Создание репозитория, импорт существующего. Создание рабочей копии.
11. Subversion. Создание рабочей копии. Обновление, внесение изменений в рабочую копию.
12. Subversion. Исправление конфликтов слияния.
13. Subversion. Анализ изменений, фиксация изменений. Просмотр истории изменений.
14. Subversion. Ветвление. Понятие ветки, суть работы в ветках.
15. Subversion. Способы создания веток. Работа с ветками.
16. Subversion. Копирование изменений между ветками.
17. Subversion. Ключевые понятия, стоящие за слиянием.
18. Subversion. Конфликты при объединении.
19. Git. Общие сведения, история создания.
20. Git. Основные понятия и определения.
21. Git. Особенности хранения изменения и организации хранилища. Целостность в Git.
22. Git. Жизненный цикл файла.
23. Git. Локальное выполнение операций.

24. Git. Первоначальная настройка репозитория.
25. Git. Жизненный цикл коммитов в Git.
26. Git. Создание репозитория, клонирование репозитория.
27. Git. Изменение файлов, запись изменений в репозиторий, просмотр изменений, отправка изменений на удаленный сервер.
28. Git. Механизм меток. Виды меток.
29. Git. Игнорирование файлов. Составление файла игнорирования.
30. Git. Ветвление. Понятие ветки, суть работы в ветках.
31. Git. Способы создания веток. Работа с ветками.
32. Git. Копирование изменений между ветками.
33. Git. Конфликты при слиянии. Способы разрешения конфликтов.
34. Git. Совместная серверная работа.
35. Git. Модель ветвления git flow.
36. Git. Поиск ломающего коммита методом половинного деления.
37. Системы отслеживания ошибок. Назначение, основные преимущества.
38. Системы отслеживания ошибок. Состав информации о дефекте.
39. Системы отслеживания ошибок. Жизненный цикл дефекта.
40. Системы отслеживания ошибок. Основные представители на рынке.
41. Тестирование программного обеспечения. Основные понятия, история тестирования.
42. Тестирование программного обеспечения. Пирамида тестирования.
43. Тестирование программного обеспечения. Модульное тестирование.
44. Тестирование программного обеспечения. Интеграционное тестирование.
45. Тестирование программного обеспечения. Тестирование UI.
46. Тестирование программного обеспечения. Виды тестирования.
47. Тестирование программного обеспечения. Тестирование методом чёрного ящика.
48. Тестирование программного обеспечения. Тестирование методом белого ящика.
49. Тестирование программного обеспечения. Тестирование методом серого ящика.
50. Тестирование программного обеспечения. Разработка через тестирование.
51. Тестовый фреймворк Test NG. Общие сведения и назначение.
52. Тестовый фреймворк Test NG. Автоматическая сборка. Особенности, преимущества.
53. Система автоматической сборки Gradle.
54. Тестовый фреймворк Test NG. Команды группы Assert.
55. Тестовый фреймворк Test NG. Аннотации.
56. Непрерывная интеграция и развертывание (CI/CD). Понятие и назначение.
57. CI/CD. Требования к проекту, организация CI. Преимущества и недостатки.
58. CI/CD. Непрерывная доставка. Общая схема CI/CD.
59. CI/CD. Основные принципы.
60. Devops. Понятие, области применения. Цели Devops.
61. Взаимодействие CI/CD и Devops.
62. Devops. Преимущества.
63. Виртуализация. Понятие, назначение.
64. Виртуализация. Виды виртуализации.

65. Виртуализация. Виды виртуализации платформ.
66. Виртуализация. Виды виртуализации ресурсов.
67. Виртуализация. Области применения, современное состояние.
68. Контейнеризация. Понятие, назначение, преимущества.
69. Контейнеризация. Docker. Назначение, терминология.
70. Контейнеризация. Рабочий процесс разработки приложений.