

Оборотная сторона титульного листа РПД

I. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от « ____ » _____ 20__ г. № ____
Директор департамента _____
(подпись) (И.О.Фамилия)

II. **Рабочая программа пересмотрена на заседании департамента:**
Протокол от « ____ » _____ 20__ г. № ____
Директор департамента _____
(подпись) (И.О.Фамилия)

Аннотация дисциплины
Методы коллективной разработки и верификации
программного обеспечения

Общая трудоемкость дисциплины 4 зачетных единицы (144 часов). Дисциплина входит в базовую часть блока «Дисциплины (модули)» Б1.В.02.05.

Дисциплина реализуется на 2 курсе в 3, 4 семестрах, завершается экзаменом в 3, 4 семестрах. Учебным планом предусмотрено 18 часов лекций, 40 час. лабораторных работ (из них 30 час. в интерактивной форме), самостоятельная работа студента – 86 часа, в том числе 54 час. на подготовку к экзамену..

Цель дисциплины – обучение студентов методам коллективной разработки сложного программного обеспечения для использования в профессиональной деятельности в различных предметных областях, в том числе и в слабо формализованных. Рассматриваются процессы: выявления участков профессиональной деятельности, которые подлежат автоматизации с использованием компьютера; моделирование соответствующих предметных областей; разработка постановок задач на построенных моделях; формулирования требований к создаваемой программой системе, ее разработки и сопровождения.

Задачи дисциплины:

1. Формирование готовности проявлять качества лидера и организовать работу коллектива разработчиков при разработке проектов информационных систем для автоматизации профессиональной деятельности, владений эффективными технологиями решения профессиональных проблем

2. Овладение умением работать в проектных междисциплинарных командах, в том числе в качестве руководителя

3. Овладение навыками организации промышленного тестирования создаваемого программного обеспечения

Для успешного изучения дисциплины «Методы коллективной разработки и верификации программного обеспечения» у обучающихся должны быть сформированы следующие предварительные компетенции:

готовность анализировать проблемы и направления развития технологий программирования;

способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения;

способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений;

способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения;

способность использовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки.

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют формирование следующих компетенций:

Наименование категории (группы) компетенций	Код и наименование компетенции (результат освоения)	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
Командная работа и лидерство	УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК 3.1 формирует стратегию командной работы на основе совместного обсуждения целей и направлений деятельности для их реализации	<u>Знает</u> принципы формирования команд разработчиков программного обеспечения <u>Умеет</u> исполнять основные роли участников команд разработчиков программного обеспечения <u>Владеет</u> навыками исполнения ролей участников команд разработчиков программного обеспечения
		УК 3.2 организует работу команды с учетом объективных условий (технология, внешние факторы, ограничения), индивидуальных особенностей поведения и возможностей членов команды	<u>Знает</u> принципы взаимодействия участников команд разработчиков программного обеспечения <u>Умеет</u> использовать инструменты взаимодействия участников команд разработчиков программного обеспечения <u>Владеет</u> навыками работы с инструментами взаимодействия участников команд разработчиков программного обеспечения
		УК 3.3 обеспечивает выполнение поставленных задач на основе мониторинга командной работы и своевременного	<u>Знает</u> принципы взаимодействия участников команд разработчиков программного обеспечения <u>Умеет</u> использовать инструменты взаимодействия участников команд

		реагирования на существенные отклонения	разработчиков программного обеспечения. <i>Владеет</i> навыками исполнения ролей участников команд разработчиков программного обеспечения
научно-исследовательский	ПК-5. Способен выполнить постановку новых задач анализа и синтеза новых проектных решений	ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений.	<i>Знает</i> современные технологии проектирования и производства программного обеспечения <i>Умеет</i> правильно выбрать технологию производства программного обеспечения для конкретного проекта <i>Владеет</i> навыками применения технологий производства программного обеспечения
		ПК-5.2 использует методы постановки новых задач анализа и синтеза новых проектных решений	<i>Знает</i> современные технологии проектирования и производства программного обеспечения <i>Умеет</i> правильно выбрать технологию производства программного обеспечения для конкретного проекта <i>Владеет</i> навыками применения технологий производства программного обеспечения
		ПК-5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности	<i>Знает</i> современные технологии реализации, оценки качества и анализа эффективности программного обеспечения <i>Умеет</i> правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения <i>Владеет</i> навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
научно-исследовательский	ПК-6. Способен использовать существующие подходы к верификации моделей программного обеспечения	ПК-6.1 демонстрирует знание методов верификации моделей программного обеспечения	<i>Знает</i> основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Умеет</i> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Владеет</i> навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
		ПК-6.2 использует методы верификации моделей	<i>Знает</i> основные концептуальные положения функционального, логического, объектно-

		программного обеспечения	ориентированного и визуального направлений программирования <u>Умеет</u> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <u>Владеет</u> навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
		ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности	<u>Знает</u> основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования <u>Умеет</u> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <u>Владеет</u> навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
проектный	ПК-8. Способен организовать промышленное тестирование создаваемого программного обеспечения	ПК-8.1 демонстрирует знание методов организации промышленного тестирования создаваемого программного обеспечения	<u>Знает</u> методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО. <u>Умеет</u> использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО <u>Владеет</u> навыками коллективной разработки и использования ПО
		ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения	<u>Знает</u> методы организации работы в профессиональной деятельности <u>Умеет</u> использовать методы организации работы в профессиональной деятельности <u>Владеет</u> навыками организации работы в профессиональной деятельности.
		ПК-8.3 применяет методы организации тестирования программных средств	<u>Знает</u> методы разработки и сопровождения ПО при создании программных средств <u>Умеет</u> применять методы разработки и сопровождения ПО при создании программных средств <u>Владеет</u> навыками создания программных средств и их сопровождения

I. Цели и задачи освоения дисциплины:

Цель: обучение студентов методам коллективной разработки сложного программного обеспечения для использования в профессиональной деятельности в различных предметных областях, в том числе и в слабо формализованных. Рассматриваются процессы: выявления участков профессиональной деятельности, которые подлежат автоматизации с использованием компьютера; моделирование соответствующих предметных областей; разработка постановок задач на построенных моделях; формулирования требований к создаваемой программой системе, ее разработки и сопровождения.

Задачи:

- Формирование знаний и умений в области устройства, низкоуровневого программирования, построения и комплексирования вычислительных систем.
- Изучение направлений развития современных компьютеров, системных программных средств;
- Изучение особенностей архитектур современных компьютеров, и компьютерных систем, влияющих на их выбор и сопровождение.

Для успешного изучения дисциплины у обучающихся должны быть сформированы следующие предварительные компетенции: знания о системах счисления, представлении данных в ЭВМ, физика в пределах школьного курса, навыки по проектированию и разработке программ, навыки по поиску необходимой информации в интернет.

Общепрофессиональные компетенции студентов, индикаторы их достижения и результаты обучения по дисциплине

Наименование категории (группы) компетенций	Код и наименование компетенции (результат освоения)	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
Командная работа и лидерство	УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК 3.1 формирует стратегию командной работы на основе совместного обсуждения целей и направлений деятельности для их реализации	<u>Знает</u> принципы формирования команд разработчиков программного обеспечения <u>Умеет</u> исполнять основные роли участников команд разработчиков программного обеспечения <u>Владеет</u> навыками исполнения ролей участников команд

			разработчиков программного обеспечения
		УК 3.2 организует работу команды с учетом объективных условий (технология, внешние факторы, ограничения), индивидуальных особенностей поведения и возможностей членов команды	<u>Знает</u> принципы взаимодействия участников команд разработчиков программного обеспечения <u>Умеет</u> использовать инструменты взаимодействия участников команд разработчиков программного обеспечения <u>Владеет</u> навыками работы с инструментами взаимодействия участников команд разработчиков программного обеспечения
		УК 3.3 обеспечивает выполнение поставленных задач на основе мониторинга командной работы и своевременного реагирования на существенные отклонения	<u>Знает</u> принципы взаимодействия участников команд разработчиков программного обеспечения <u>Умеет</u> использовать инструменты взаимодействия участников команд разработчиков программного обеспечения. <u>Владеет</u> навыками исполнения ролей участников команд разработчиков программного обеспечения
научно-исследовательский	ПК-5. Способен выполнить постановку новых задач анализа и синтеза новых проектных решений	ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений.	<u>Знает</u> современные технологии проектирования и производства программного обеспечения <u>Умеет</u> правильно выбрать технологию производства программного обеспечения для конкретного проекта <u>Владеет</u> навыками применения технологий производства программного обеспечения
		ПК-5.2 использует методы постановки новых задач анализа и синтеза новых проектных решений	<u>Знает</u> современные технологии проектирования и производства программного обеспечения <u>Умеет</u> правильно выбрать технологию производства программного обеспечения для конкретного проекта <u>Владеет</u> навыками применения технологий производства программного обеспечения
		ПК-5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности	<u>Знает</u> современные технологии реализации, оценки качества и анализа эффективности программного обеспечения <u>Умеет</u> правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения

			<i>Владеет</i> навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
научно-исследовательский	ПК-6. Способен использовать существующие подходы верификации моделей программного обеспечения	ПК-6.1 демонстрирует знание методов верификации моделей программного обеспечения	<i>Знает</i> основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Умеет</i> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Владеет</i> навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
		ПК-6.2 использует методы верификации моделей программного обеспечения	<i>Знает</i> основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Умеет</i> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Владеет</i> навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
		ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности	<i>Знает</i> основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Умеет</i> создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования <i>Владеет</i> навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
проектный	ПК-8. Способен организовать промышленное тестирование создаваемого	ПК-8.1 демонстрирует знание методов организации промышленного	<i>Знает</i> методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО. <i>Умеет</i> использовать методы организации работы в коллективах

	программного обеспечения	тестирования создаваемого программного обеспечения	разработчиков ПО и методы сопровождения ПО <i>Владеет</i> навыками коллективной разработки и использования ПО
		ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения	<i>Знает</i> методы организации работы в профессиональной деятельности <i>Умеет</i> использовать методы организации работы в профессиональной деятельности <i>Владеет</i> навыками организации работы в профессиональной деятельности.
		ПК-8.3 применяет методы организации тестирования программных средств	<i>Знает</i> методы разработки и сопровождения ПО при создании программных средств <i>Умеет</i> применять методы разработки и сопровождения ПО при создании программных средств <i>Владеет</i> навыками создания программных средств и их сопровождения

II. Трудоемкость дисциплины и виды учебных занятий по дисциплине

Общая трудоемкость дисциплины составляет 4 зачётных единиц (144 академических часа).

Структура дисциплины:

Форма обучения – очная

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1	Раздел I. Структура и анализ бизнес-процессов разработки программного обеспечения.	3	18	18	-	-	32	54	УК-3, ОПК-8, ПК-6, ПК-7, ПК-9
2	Раздел II. Тестирование и отладка программного средства	4		22		-			
Итого:			18	40		-	32	54	

*онлайн курс

III. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА **Трудоемкость теоретической части курса 18 час.** **Семестр 3 (18 часов)**

Тема 1. Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности (3 час).

Профессиональные позиции ИТ специалистов и их особенности. Особенности работы в отечественных и международных ИТ компаниях. Понятие коллективной разработки программного обеспечения. Средства поддержки процесса коллективной разработки ПО.

Тема 2. Модели жизненного цикла разработки программных средств (3 час).

Фазы стандартного процесса разработки ПО. Виды проектной документации, сопровождающей разработку ПО. Модели жизненных циклов разработки ПО: водопад, водопад с перекрытиями, водопад с возвратами, инкрементная модель, итеративная модель, итеративно-инкрементная модель, спиральная модель.

Тема 3. Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов (3 час).

Верификация программных продуктов в процессе их разработки. Принципы проведения инспекций кода, дизайна, тестов, требований. Роли участников инспекций. Процедура организации и проведения формальной инспекции. Метрики по результатам инспекций. Статус и степень серьезности замечаний по инспекции. Анализ результатов инспекции. Корректировка процесса разработки программного обеспечения на основе результатов инспекции.

Тема 4. Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов (3 час).

Workflow системы отслеживания дефектов и сопровождения задач при разработке программных продуктов. Технологический процесс коллективной разработки программ. Классы подзадач. Основные состояния подзадачи. Переход подзадач из состояния в состояние. Специфика систем сопровождения заданий (issue tracking) и отслеживания дефектов (defect

tracking). Специфика управления проектами с использованием метрик сотрудников.

Тема 5. Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников (3 час).

Особенности бизнес-процессов ИТ предприятий. Общие элементы методики регистрации временных затрат. Проектные и непроjektные виды деятельности. Типы действий различных видов деятельности. Логирование времени сотрудником. Отчеты о затратах времени и уведомления. Специфика управления проектами с использованием метрик сотрудников.

Тема 6. Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС (3 час).

Роль и место измерений при производстве программных средств. Виды измерений, связанные с процессом разработки ПО. Метрики эффективности процесса производства. Метрики качества продуктов. Специфика управления проектами с использованием метрик сотрудников. Структура информационных систем, обеспечивающих программу измерений.

IV. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ (0 час.)

Лабораторные работы (40 час.)

Лабораторная работа №1. Организация коллективной работы разработчиков (4 часа)

1. Объединение в проектные группы, распределение ролей
2. Выбор направленности, типа и конкретной темы проекта
3. Выбор и обоснование инструментов разработки

Лабораторная работа №2. Работа с заказчиком программного продукта (4 часа)

1. Сбор и формулировка основных требований к программному продукту
2. Объектный анализ и концептуальное моделирование конкретной предметной области

3. Выбор и обоснование архитектуры проекта

Лабораторная работа №3. Работа с требованиями к программному продукту (4 часа)

1. Разработка пользовательских требований и внешних спецификаций к программной системе, подготовка документа «Пользовательские требования»
2. Разработка системных требований и верхнего уровня проекта программной системы; подготовка документа «Системные требования»
3. Разработка архитектуры программной системы; подготовка документа «Дизайн проекта»

Лабораторная работа №4. Кодирование программной системы (4 часа)

1. Подготовка документа «Список задач проекта»
2. Разработка политики по управлению конфигурацией проекта
3. Разработка политики по верификации рабочих продуктов на проекте

Лабораторная работа №5. Тестирование программной системы (4 часа)

1. Подготовка документа «План тестирования проекта»
2. Подготовка документа «Матрица покрытия требований»
3. Разработка политики по прогону тестов на проекте (модульное, системное тестирование)

Лабораторная работа №6. Доработка программной системы (5 час)

1. Разработка списка задач на доработку программной системы по результатам тестирования
2. Подготовка отчета по эффективности обработки задач

Лабораторная работа №7. Метрики качества разработки программной системы (5 час)

1. Разработка системы метрик для оценки качества программного продукта
2. Сбор метрик

Лабораторная работа №8. Метрики эффективности процесса разработки программной системы (5 часа)

1. Разработка системы метрик для эффективности процесса разработки программного продукта
2. Сбор метрик

Лабораторная работа №9. Выработка рекомендаций по изменению процесса разработки (5 часа)

1. Анализ метрик для оценки качества программного продукта и эффективности процесса разработки

Формулировка рекомендаций по трансформации процесса разработки по результатам программы измерений

V. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	ПК-5.1 демонстрирует знание методов постановки новых задач анализа и синтеза новых проектных решений.	Знает современные технологии проектирования и производства программного обеспечения	ПР1 лабораторные работы и задания для самостоятельного выполнения	Зачет, вопросы № 1-26
			Умеет правильно выбрать технологию производства программного обеспечения		
			Владеет навыками применения технологий производства программного обеспечения		
		ПК-5.2 использует методы постановки новых задач анализа и синтеза новых	Знает современные технологии проектирования и производства программного обеспечения		
			Умеет правильно выбрать технологию		

проектных решений	производства программного обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
	Знает современные технологии реализации, оценки качества и анализа эффективности программного обеспечения
ПК5.3 применяет методы разработки постановок задач анализа и синтеза новых проектных решений, требуемых в профессиональной деятельности	Умеет правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения
	Владеет навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального программирования

<p>ПК-6.1 демонстрирует знание методов верификации моделей программного обеспечения</p>	<p>Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования</p>
	<p>Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования</p>
	<p>Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования</p>
<p>ПК-6.2 использует методы верификации моделей программного обеспечения.</p>	<p>Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования</p>
	<p>Владеет навыками создания программ в рамках функционального</p>

			о, логического, объектно-ориентированного и визуального направлений программирования		
			Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования		
		ПК-6.3 применяет методы проведения верификации моделей программного обеспечения, требуемых в профессиональной деятельности	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования		
			Владеет навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования		
			Знает современные технологии проектирования и производства программного обеспечения		
2	Раздел 2. Тестирование и отладка		ПК-8.1 демонстрирует знание методов организации	Знает методы организации работы в	ПР1 лабораторн

программного средства	промышленного тестирования создаваемого программного обеспечения	коллективах разработчиков ПО и методы сопровождения ПО.	ые работы и задания для самостоятельного выполнения	№ 1-16
		Умеет использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.		
		Владеет навыками коллективной разработки и использования ПО		
	ПК-8.2 использует методы организации промышленного тестирования создаваемого программного обеспечения	Знает методы организации работы в профессиональной деятельности.		
		Умеет использовать методы организации работы в профессиональной деятельности.		
		Владеет навыками организации работы в профессиональной деятельности.		
	ПК-8.3 применяет методы организации тестирования программных средств	Знает методы разработки и сопровождения ПО при создании программных средств		
		Умеет применять методы разработки и сопровождения ПО при		

			создании программных средств	
			Владеет навыками создания программных средств и их сопровождения	

* Формы оценочных средств:

1) собеседование (УО-1).

2) контрольная работа (ПР-2), лабораторная работа (ПР-6), тест (ПР-1).

VI. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Самостоятельная работа определяется как индивидуальная или коллективная учебная деятельность, осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем. Самостоятельная работа – это познавательная учебная деятельность, когда последовательность мышления студента, его умственных и практических операций и действий зависит и определяется самим студентом.

Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня, что в итоге приводит к развитию навыка самостоятельного планирования и реализации деятельности.

Целью самостоятельной работы студентов является овладение необходимыми компетенциями по своей специальности, опытом творческой и исследовательской деятельности.

Формы самостоятельной работы студентов:

- работа с основной и дополнительной литературой, Интернет ресурсами;
- самостоятельное ознакомление с лекционным материалом, представленным на электронных носителях, в библиотеке образовательного учреждения;

- выполнение лабораторных работ;
- решение задач;
- выполнение тестов;
- подготовка к экзамену;
- другие виды деятельности, организуемые и осуществляемые образовательным учреждением и органами студенческого самоуправления.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Методы коллективной разработки и верификации программного обеспечения» включает в себя план-график выполнения самостоятельной работы по дисциплине.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
3 семестр				
1	1-4 неделя	Software Project Plan (SPP) – календарный план работы над проектом	2 час	Собеседование
2	5-9 неделя	Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера	2 час	Проект
3	10-14 неделя	Software Requirements Specification (SRS) - технические требования к продукту	2 час	Проект
4	15-18 неделя	User Manual - руководство оператора (пользователя).	3 час	Проект
5	18	Подготовка к экзамену	27 часов	экзамен
4 семестр				
5	1-3 неделя	Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.	4 час	Проект
6	4-7 неделя	Code Issue List - список задач, выполняемых на этапе кодирования.	4 час	проект
7	8-11 неделя	Software Test Plan - докумен, включающий в себя спецификации тестов и процедур	5 час	Проект

		тестирования, в том числе список тестов.		
8	12-15 неделя	Requirements Tracaebility Matrix - матрица покрытия тестами требований.	5 час	Проект
9	16-18 неделя	Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.	5 час	Проект
	18 неделя	Подготовка к экзамену	27 час	экзамен
	ИТОГО		86 час	

Самостоятельная работа включает в себя разработку итогового проекта, параллельно с лабораторными работами по дисциплине. В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель).

Методические указания к самостоятельной работе студентов

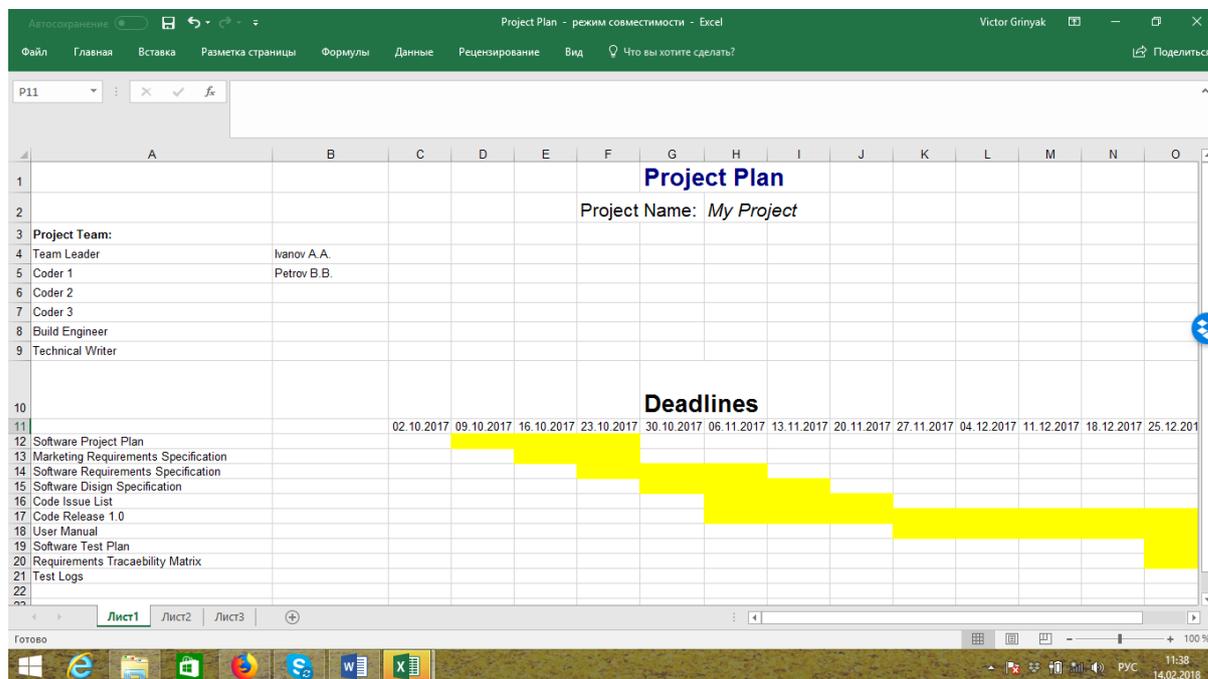
Самостоятельная работа включает в себя разработку итогового проекта, параллельно с лабораторными работами по дисциплине. В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель).

В рамках работы над проектом создается следующая документация:

- Software Project Plan (SPP) - календарный план работы над проектом.
- Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера.
- Software Requirements Specification (SRS) - технические требования к продукту
- User Manual - руководство оператора (пользователя).
- Software Disign Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.
- Code Issue List - список задач, выполняемых на этапе кодирования.
- Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.
- Requirements Tracaebility Matrix - матрица покрытия тестами требований.
- Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.

Результаты самостоятельной работы представляются на итоговом семинаре в виде доклада (презентации).

Примерная форма документа Software Project Plan



Примерная форма документа Software Requirements Specification

Программный продукт *название* предназначен для

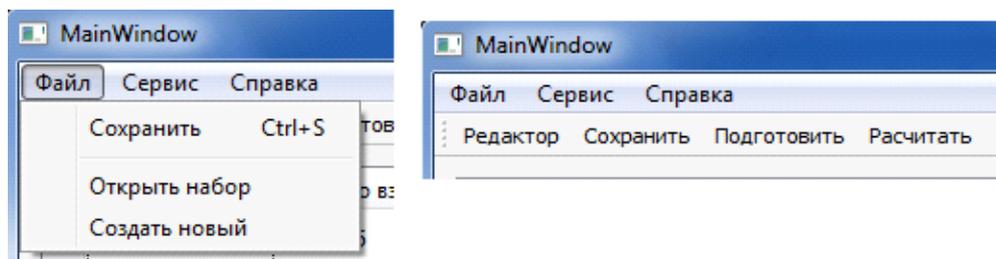
Программный продукт *название* состоит из следующих подсистем:

1. Пользовательский интерфейс
2. Модуль расчета параметров X
3. Модуль расчета параметров Y
4. Подсистема ввода-вывода
5. ...
6. ...

Требования к подсистеме «Пользовательский интерфейс»

Требование REQ_UI_001

Функция сохранения данных должна быть доступна для пользователя тремя способами: кнопка «Сохранить» на панели инструментов, элемент меню «Файл» и нажатием сочетания клавиш «Ctrl + S» (см. Рисунок 10)



Требование REQ_UI_002

Если пользователь ввел данные в строке «+» в одной из ячеек столбцов для отображения обязательной информации, то система должна создать новую строку «+», только если заполнены все ячейки столбцов обязательной информации доступные для редактирования (см. требование REQ_UI_003).

Требование REQ_UI_003

...

Требования к подсистеме «Модуль расчета параметров X»

Требование REQ_CX_001

.....

Требование REQ_CX_002

....

Примерная форма документа Software Disign Specification

Программный продукт *название* предназначен для

Настоящий документ содержит спецификации дизайна, включающие в себя:

1. Архитектурно-контекстную диаграмму системы в целом
2. Архитектурно-контекстную диаграммы подсистем
3. Диаграммы наследования классов (по подсистемам)
4. Диаграммы вызовов объектов (по подсистемам)

1. Архитектурно-контекстная диаграмма системы

Программный продукт *название* состоит из следующих подсистем:

1. Пользовательский интерфейс

2. Модуль расчета параметров X
3. Модуль расчета параметров Y
4. Подсистема ввода-вывода
5. ...
6. ...

2.1. Архитектурно контекстная диаграмма подсистемы «Пользовательский интерфейс»

...

2.2. Диаграмма наследования классов подсистемы «Пользовательский интерфейс»

...

2.3. Диаграммы вызовов объектов подсистемы «Пользовательский интерфейс»

...

3.1. Архитектурно контекстная диаграмма подсистемы «Модуль расчета параметров X»

...

3.2. Диаграмма наследования классов подсистемы «Модуль расчета параметров X»

...

3.3. Диаграммы вызовов объектов подсистемы «Модуль расчета параметров X»

...

Примерная форма документа Requirements Traceability Matrix

The screenshot shows an Excel spreadsheet with the following content:

- Row 1: Title "Requirements Traceability Matrix" in blue.
- Row 2: "Project Name: My Project"
- Row 3: "Project Test Team:"
- Row 4: "Tester 1 Ivanov A.A."
- Row 5: "Tester 2 Petrov B.B."
- Row 6: "Tester 3"
- Row 7: "Tester 4"
- Row 8: "Tester 5"
- Row 10: "Test ID" (header for a table)
- Row 11: "Req ID" (header for a table)
- Row 12: Table with columns: TEST_UI_001, TEST_UI_002, TEST_CX_001, TEST_CX_002, TEST_SYS_001, TEST_SYS_002
- Row 13: REQ_UI_001 (yellow cell under TEST_UI_001, yellow cell under TEST_SYS_001)
- Row 14: REQ_UI_002 (yellow cell under TEST_UI_002, yellow cell under TEST_CX_002)
- Row 15: REQ_CX_001 (yellow cell under TEST_CX_001, yellow cell under TEST_SYS_002)
- Row 16: REQ_CX_002 (yellow cell under TEST_CX_002)

Примерная форма документа Software Test Plan

Тесты для тестирования подсистемы «Пользовательский интерфейс»

Тест TEST_UI_001

Тестируемые требования: REQ_UI_001
Описание теста...

Тест TEST_UI_002

Тестируемые требования: REQ_UI_002
Описание теста...

Тесты для тестирования подсистемы «Модуль расчета параметров X»

Тест TEST_CX_001

Тестируемые требования: REQ_CX_001
Описание теста...

Тест TEST_CX_002

Тестируемые требования: REQ_CX_002
Описание теста...

Тесты для тестирования системы в целом (System Test)

Тест TEST_SYS_001

Тестируемые требования: REQ_CX_001, REQ_CX_002, REQ_UI_002
Описание теста...

Тест TEST_SYS_002

Тестируемые требования: REQ_CX_002
Описание теста...

Примерная форма документа Software Test Logs

Тесты для тестирования подсистемы «Пользовательский интерфейс»

Тест TEST_UI_001

Тестируемые требования: REQ_UI_001
Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*
Ожидаемый результат: Показано 5 опций меню
Видимый результат: Показано 5 опций меню
Резюме: **Тест пройден**

Тест TEST_UI_002

Тестируемые требования: REQ_UI_002
Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*
Ожидаемый результат: Показано 2 всплывающих окна
Видимый результат: Показано 1 всплывающее окно
Резюме: **Тест НЕ пройден**

Тесты для тестирования подсистемы «Модуль расчета параметров X»

Тест TEST_CX_001

Тестируемые требования: REQ_CX_001
Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*
Ожидаемый результат:
Видимый результат:
Резюме: **Тест пройден**

Тест TEST_CX_002

Тестируемые требования: REQ_CX_002
Описание теста...

Тестируемая версия продукта: *номер версии из GitHub*
Ожидаемый результат:
Видимый результат:
Резюме: **Тест НЕ пройден**

Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ (<http://www.dvfu.ru/library/>) и других ведущих вузов страны, а также доступных для использования научно-библиотечных систем.

VII. СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с. Режим доступа: <http://znanium.com/go.php?id=389963>

2. Введение в архитектуру программного обеспечения: Учебное пособие / Гагарина Л.Г., Федоров А.Р., Федоров П.А. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 320 с.: 60x90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0649-1
<http://znanium.com/catalog.php?bookinfo=542665>

Дополнительная литература

1. Синицын, С.В. Верификация программного обеспечения: учебное пособие / С.В. Синицын, Н.Ю. Налютин. – М.: Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2008. – 367 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:274428&theme=FEFU>
2. Соммервилл И. Инженерия программного обеспечения. 6-е издание. М.: Изд. дом Вильямс, 2002. – 624 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:735&theme=FEFU>
3. Браудэ Э. Технология разработки программного обеспечения, Издательский дом «Питер», 2004. - 656 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:232388&theme=FEFU>
4. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002. <http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
5. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004. – 215 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
6. Бенкен Е.С. PHP, MySQL, XML программирование для Интернета. СПб: БХВ-Петербург. -2008. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:382736&theme=FEFU>
7. Скляр Д., Трахтенберг А. PHP. Рецепты программирования. – СПб: Русская редакция БХВ-Петербург. -2007. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:381965&theme=FEFU>
8. Липаев, В.В. Программная инженерия. Методологические основы [Текст]: Учеб. / В. В. Липаев; Гос. ун-т — Высшая школа экономики. — М. : ТЕИС, 2006. — 608 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:248067&theme=FEFU>
9. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В. П. Котляров, Т. В. Коликова. Москва: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2012. – 285 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:668103&theme=FEFU>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронная библиотечная система ZNANIUM.COM
<http://znanium.com/>
2. Студенческая электронная библиотека «Консультант студента»
<http://www.studentlibrary.ru/cgi-bin/mb4x>
3. Научная библиотека ДВФУ. Электронный каталог
<http://lib.dvfu.ru:8080/>
4. <http://e.lanbook.com/book/43554> Шопырин, Д.Г. Управление проектами разработки ПО. Дисциплина «Гибкие технологии разработки программного обеспечения». [Электронный ресурс] — Электрон. дан. — СПб.: НИУ ИТМО, 2007. — 131 с.
5. <http://novtex.ru/> Журналы издательства Новые технологии
6. <https://github.com/> Система контроля версий Git

Профессиональные базы данных и информационные справочные системы

1. База данных Scopus <http://www.scopus.com/home.url>
2. База данных Web of Science <http://apps.webofknowledge.com/>
3. Научная электронная библиотека <https://www.elibrary.ru/>
4. Общероссийский математический портал Math-Net.Ru
<http://www.mathnet.ru>
5. Электронная библиотека диссертаций Российской государственной библиотеки <http://diss.rsl.ru/>
6. Электронная библиотека Европейского математического общества
<https://www.emis.de/>
7. Электронные базы данных EBSCO <http://search.ebscohost.com/>

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office (Access, Excel, PowerPoint, Word и т. д).
2. Open Office.
3. Интегрированные среды разработчика Eclipse, NetBeans, Visual Studio.

4. Система контроля версий Git.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VIII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина изучается в следующих организационных формах: лекционное занятие; лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального проекта; индивидуальные и групповые консультации.

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение рекомендованной литературой, выполнение проекта, а также активная работа на лабораторных занятиях.

К прослушиванию лекции следует готовиться, для этого необходимо знать программу курса и рекомендованную литературу. Тогда в процессе лекции легче отделить главное от второстепенного, легче сориентироваться: что записать, что самостоятельно проработать, что является трудным для понимания, а что легко усвоить.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в документации и защиты проекта.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

Особенности проведения лабораторных работ по дисциплине (активные и интерактивные формы)

Лабораторные работы по дисциплине проводятся параллельно с самостоятельной работой студентов в форме деловой игры.

Особенностью учебного процесса по дисциплине является его

максимальное приближение к реальным условиям работы на проектах по коллективной разработке ПО: как коммерческим, так и фрилансовым. Всем аспирантам необходимо создать аккаунт на портале github.com. Разработка проектов ведётся на базе именно этой системы контроля версий с использованием встроенной системы issue трекинга.

В начале семестра студенты разбиваются на команды (по 3-5 человек в каждой). Каждая команда придумывает название своего проекта и определяется с составом (тим-лидер, кодировщики, технический писатель). В том случае, если группа аспирантов немногочисленна (1-2 человека), работа проводится по упрощённой схеме.

В обязанности тим-лидера входит общее руководство проектом и поддержание целостности программного кода путем контроля интеграции всех изменений программного кода.

В обязанности кодировщиков входит текущая работа над проектом, в том числе - написание кода согласно требованиям и инспектирование кода других кодировщиков.

В обязанности технического писателя входит разработка документации и поддержание её целостности.

Структура рабочего продукта по проекту

Тим-лидер каждой команды создаёт репозиторий в системе контроля версий, называя его согласно названию проекта в целом. В корне репозитория должен находиться файл `readme`, содержащий краткое описание проекта. Кроме того, в корне находятся папки `CODE` и `DOCS`.

В папке `CODE` размещается код проекта.

В папке `DOCS` - файлы с документацией.

Папка `DOCS` содержит, в свою очередь, папки:

`PROJECT PLAN` - для размещения документов с планом проекта

`REQUIREMENTS` - для размещения документов с требованиями

`DESIGN` - для размещения документов дизайна

`CODING` - для размещения документов кодирования

`TESTING` - для размещения документов тестирования

Работа с системой контроля версий

После того, как тим-лидер создал репозиторий он указывает остальных членов команды его коллабораторами. Таким образом они получают максимум прав для работы с проектом.

Проект должен иметь двухуровневую структуру веток.

Основная ветка (`master`) - содержит стабильную версию, интеграцию в которую осуществляет только (!!) тим-лидер. Все члены команды, в том числе тим-лидер, создают в своём репозитории копии стабильной версии по

мере необходимости. С этих рабочих (work) веток они создают запросы на интеграцию (merge) в основную ветку. Запрос на интеграцию назначается вначале на одного из членов команды, который выполняет роль инспектора. Если инспектор одобряет работу, он переназначает запрос на тим-лидера, который после проверки осуществляет интеграцию изменений. Сам тим-лидер поступает аналогично: после одобрения инспектором запрос на интеграцию возвращается к тим-лидеру обратно.

Документация, создаваемая по проекту

В рамках работы над проектом создается следующая документация:

Software Project Plan (SPP) - календарный план работы над проектом.

Marketing Requirements Specification (MRS) - маркетинговые требования к проекту в виде презентации рекламного характера.

Software Requirements Specification (SRS) - технические требования к продукту

User Manual - руководство оператора (пользователя).

Software Design Specification (SDS) - документы дизайна, раскрывающие архитектуру программного продукта и то КАК именно будут реализованы те или иные требования.

Code Issue List - список задач, выполняемых на этапе кодирования.

Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.

Requirements Tracability Matrix - матрица покрытия тестами требований.

Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.

Вся документация размещается в соответствующих папках проекта. Запрос на интеграцию для документации выполняется аналогично файлам с кодом.

Работа с литературой. Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ и электронные библиотеки (<http://www.dvfu.ru/library/>), а также доступные для использования другие научно-библиотечные системы.

Подготовка к экзамену/зачету. К сдаче экзамена/зачета допускаются обучающиеся, выполнившие все задания (лабораторные, самостоятельные), предусмотренные учебной программой дисциплины, посетившие не менее 85% аудиторных занятий.

IX. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень программного обеспечения:

Лицензионное программное обеспечение:

AutoCAD;
Autodesk 3DS Max;
Microsoft Visio;
SPSS Statistics Premium Campus Edition;
MathCad Education University Edition;
Microsoft Office 365;
Office Professional Plus 2019;
Photoshop CC for teams All Apps AL;
SolidWorks Campus 500;
Windows Edu Per Device 10 Education;
КОМПАС 3D;
Microsoft Teams

Свободно распространяемое программное обеспечение:

Adobe Reader DC 2015.020 - пакет программ для просмотра электронных публикаций в формате PDF:

http://www.images.adobe.com/content/dam/acom/en/legal/licenses-terms/pdf/PlatformClients_PC_WWEULA-en_US-20150407_1357.pdf ;

ArgoUML - программный инструмент моделирования UML:
<http://argouml.tigris.org> ;

Dia - пакет программ для создания диаграмм в виде блок-схем алгоритмов программ, древовидных схем, статических структур UML, баз данных, диаграмм сущность-связь и др. диаграмм:
https://portableapps.com/support/portable_app#using ;

DiagramDesigner - пакет программ для создания потоковых диаграмм, диаграмм классов UML, иллюстраций и др. диаграмм:

<https://www.foosshub.com/Diagram-Designer.html#clickToStartDownload> ;

IrfanView - пакет программ для просмотра (воспроизведения) графических, видео- и аудиофайлов: <http://www.irfanview.com/eula.htm> ;

LibreOffice - офисный пакет:
<http://www.libreoffice.org/about-us/licenses/> ;

Maxima – система для работы с символьными и численными выражениями: <http://maxima.sourceforge.net/maximalist.html> ;

Project Libre - аналог программной системы управления проектами Microsoft Project для стационарного компьютера:
<https://континентсвободы.рф:/офис/проекты/projectlibre-система-управления-проектами.html> ;

Python - система программирования - динамический интерактивный объектно-ориентированный язык программирования:

<https://python.ru.uptodown.com/windows/download> ;

Ramus Educational - пакет программ для разработки и моделирования бизнес-процессов в виде диаграмм IDEF0 и DFD:

<https://www.obnovisoft.ru/ramus-educational> ;

Scilab –система - язык программирования высокого уровня, рассчитанный на научные расчеты: <http://www.scilab.org/scilab/license> ;

WhiteStarUML –программный инструмент моделирования UML, полученный из StarUML, совместимый с Windows 7-10:

<https://github.com/StevenTCramer/WhiteStarUml/blob/master/staruml/deploy/License.txt/> ;

WinDjView – программа для просмотра электронных публикаций в формате DJV и DjVu: <https://windjview.sourceforge.io/ru/> .

Сведения о материально-техническом обеспечении ОПОП, включая информацию о наличии оборудованных учебных кабинетов, объектов для проведения практических занятий и самостоятельной работы обучающихся с перечнем основного оборудования, объектов физической культуры и спорта, программного обеспечения представлены в виде таблицы в Справке об МТО.