



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Дальневосточный федеральный университет»**  
(ДФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**  
**«СОГЛАСОВАНО»**

Руководитель ОП

Артемова И.Л.

**«УТВЕРЖДАЮ»**

И.о. директора департамента

Смагин С.В.

«2» марта 2023 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

*Технологии коллективной промышленной разработки информационных систем*

***Направление подготовки 09.03.04 Программная инженерия***

*(Программная инженерия)*

*Форма подготовки очная*

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия, утвержденный приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920.

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, от «02» марта 2023 г., протокол № 3.0

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.

Составитель: профессор департамента ПИИИИ В.М. Гриняк, д.т.н., доцент

Владивосток

2023

**Оборотная сторона титульного листа РПУД**

**I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**III. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**IV. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## **Аннотация дисциплины**

### *Технологии коллективной промышленной разработки информационных систем*

Общая трудоемкость дисциплины составляет 5 зачётных единиц /180 академических часов. Является дисциплиной части ОП, формируемой участниками образовательных отношений, изучается на 4 курсе и завершается зачетом в 7 семестре и экзаменом в 8 семестре. Учебным планом предусмотрено проведение лекционных занятий в объеме 28 часов, лабораторных 40 часов, а также выделены часы на самостоятельную работу студента – 112 часов (в том числе на подготовку к экзамену 27 часов).

**Язык реализации:** русский.

**Цель** дисциплины - познакомить студентов с современными приёмами создания программных средств различного целевого назначения, в том числе в рамках проектной работы и различных технологий программирования.

#### **Задачи дисциплины:**

1. Развитие способности анализировать проблемы и направления развития технологий программирования
2. Приобретение способности применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
3. Развитие способности использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения
4. Приобретение способности использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях
5. Приобретение представления о проектном методе разработки программного обеспечения

Для успешного изучения дисциплины Технологии коллективной промышленной разработки информационных систем у обучающихся должны быть сформированы следующие предварительные компетенции: таких нет.

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)
Организационно-управленческий	ПК-1. Способен использовать классические концепции и модели менеджмента в управлении проектами	ПК-1.1 Демонстрирует знание концептуальных моделей менеджмента	<p><i>Знает</i> основные понятия концептуальных моделей: система, элемент, часть, модель, предметная область, состояние, предусловие, постусловие, переход, процесс, триггер, сущность, экземпляр, отношение, атрибут и т. д.</p> <p><i>Умеет</i> различать концептуальные модели (модели потоков данных, сущностных отношений, переходов состояний и т. д.), объяснять их назначения, основные особенности, достоинства и недостатки каждой модели</p> <p><i>Владеет</i> навыками создания концептуальных моделей менеджмента для выбранных предметных областей с использованием языков концептуального моделирования (UML, IDEF, язык прикладной логики и другие)</p>
		ПК-1.2 Использует основные модели менеджмента в управлении	<p><i>Знает</i> модель «Тройственной ограниченности», в которой ограничениями являются Scope, Schedule, Cost + Quality</p> <p><i>Умеет</i> применять модель «Тройственной ограниченности» в управлении проектами в конкретных предметных областях</p> <p><i>Владеет</i> навыками варьирования параметров ограничений в управлении проектами в конкретных предметных</p>

			областях
		ПК-1.3 Применяет модели и методы менеджмента в управлении программным обеспечением	<p><i>Знает</i> методы анализа требований и проектирования программного обеспечения, модели проектирования и модели реализации, стратегии и методы испытаний и метрики оценки качества программного обеспечения и его компонентов</p> <p><i>Умеет</i> оценивать применимость разных методов архитектурного проектирования к особенностям создаваемого программного обеспечения, оценивать архитектурный проект с помощью метрик качества, выполнить планирование испытаний отдельных программных единиц архитектуры и их интеграции, оценивать надежность, сопровождаемость, эффективность и другие свойства качества программного обеспечения</p> <p><i>Владеет</i> навыками архитектурного проектирования программного обеспечения, способами приспособления архитектуры к обеспечению требуемых свойств качества программного обеспечения</p>
Организационно-управленческий	ПК-2. Способен использовать методы контроля проекта и контроля версий	ПК-2.1 Демонстрирует знание основных методов контроля проекта и контроля версий	<p><i>Знает</i> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологических подходы, принципы структурного и</p>

			<p>объектно-ориентированного программирования  <u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта  <u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу</p>
		<p>ПК-2.2 Организует работы по управлению проектом информационной системы</p>	<p><u>Знает</u> определения понятий project score, project context, заинтересованные лица и роли в команде проекта (Аналитик project/product analyst, руководитель проекта PM, руководитель разработки DM, разработчик D, тестировщик TM, UI/UX специалист, маркетолог MM, технический писатель TW, специалист по анализу данных DS)  <u>Умеет</u> определять project score и project context для конкретной предметной области  <u>Владеет</u> навыками управления масштабом проекта и взаимодействия в команде проекта в зависимости от играемой роли в команде проекта</p>
		<p>ПК-2.3 Проводит переговоры и осуществляет контроль версий</p>	<p><u>Знает</u> методы выявления, сбора и анализа требований заказчика  <u>Умеет</u> осуществлять трассировку требований: от высокоуровневых бизнес-целей через пользовательские</p>

			требования к функциональным требованиям. <i>Владеет</i> навыками управления изменениями требований
Проектный	<b>ПК-8.</b> Способен создавать программные интерфейсы	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<i>Знает</i> способы создания программных интерфейсов <i>Умеет</i> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов <i>Владеет</i> набором операций над структурами данных и операторами языка программирования и методами их использования
		ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<i>Знает</i> определение интуитивно понятного интерфейса <i>Умеет</i> создавать интуитивно понятные интерфейсы <i>Владеет</i> набором средств для создания интуитивно понятного интерфейса
		ПК-8.3 Создает современные программные интерфейсы	<i>Знает</i> свойства современного программного интерфейса <i>Умеет</i> создавать современные программные интерфейсы <i>Владеет</i> методами обеспечения качества создаваемых интерфейсов
Производственно-технологический	<b>ПК-12.</b> Способен применять стандарты и модели жизненного цикла	ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<i>Знает</i> содержание основных стандартов и модели жизненного цикла ПО <i>Умеет</i> работать с документацией, касающейся основных стандартов и моделей жизненного цикла ПО <i>Владеет</i> навыками работы со стандартами и различными моделями жизненного цикла ПО при создании ПО

		<p>ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения</p>	<p><u>Знает</u> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО  <u>Умеет</u> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла  <u>Владеет</u> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области</p>
		<p>ПК-12.3. Применяет стандарты и модели жизненного цикла программного обеспечения</p>	<p><u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы, принципы структурного и объектно-ориентированного программирования, основные понятия СОМ-технологий и SOA-технологий; тенденции развития современных технологий программирования  <u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбирать технологический подход к разработке и соответствующие средства автоматизации разработки  <u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного</p>

			цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов
--	--	--	--

## I. Цели и задачи освоения дисциплины:

**Цель** дисциплины - познакомить студентов с современными приёмами создания программных средств различного целевого назначения, в том числе в рамках проектной работы и различных технологий программирования.

### **Задачи дисциплины:**

6. Развитие способности анализировать проблемы и направления развития технологий программирования
7. Приобретение способности применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
8. Развитие способности использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения
9. Приобретение способности использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях
10. Приобретение представления о проектном методе разработки программного обеспечения

Для успешного изучения дисциплины Технологии коллективной промышленной разработки информационных систем у обучающихся должны быть сформированы следующие предварительные компетенции: таких нет.

Место дисциплины в структуре ОПОП ВО: является дисциплиной части ОП, формируемой участниками образовательных отношений, изучается на 4 курсе и завершается экзаменом.

Профессиональные 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)
Организационно-управленческий	ПК-1. Способен использовать классические концепции и модели менеджмента в управлении	ПК-1.1 Демонстрирует знание концептуальных моделей менеджмента	<i>Знает</i> основные понятия концептуальных моделей: система, элемент, часть, модель, предметная область,

	проектами		<p>состояние, предусловие, постусловие, переход, процесс, триггер, сущность, экземпляр, отношение, атрибут и т. д.</p> <p><u>Умеет</u> различать концептуальные модели (модели потоков данных, сущностных отношений, переходов состояний и т. д.), объяснять их назначения, основные особенности, достоинства и недостатки каждой модели</p> <p><u>Владеет</u> навыками создания концептуальных моделей менеджмента для выбранных предметных областей с использованием языков концептуального моделирования (UML, IDEF, язык прикладной логики и другие)</p>
		ПК-1.2 Использует основные модели менеджмента в управлении	<p><u>Знает</u> модель «Тройственной ограниченности», в которой ограничениями являются Scope, Schedule, Cost + Quality</p> <p><u>Умеет</u> применять модель «Тройственной ограниченности» в управлении проектами в конкретных предметных областях</p> <p><u>Владеет</u> навыками варьирования параметров ограничений в управлении проектами в конкретных предметных областях</p>
		ПК-1.3 Применяет модели и методы менеджмента в управлении программным обеспечением	<p><u>Знает</u> методы анализа требований и проектирования программного обеспечения, модели проектирования и модели реализации, стратегии и методы испытаний и метрики оценки качества программного обеспечения и его компонентов</p> <p><u>Умеет</u> оценивать применимость разных методов архитектурного проектирования к особенностям создаваемого программного обеспечения, оценивать архитектурный</p>

			<p>проект с помощью метрик качества, выполнить планирование испытаний отдельных программных единиц архитектуры и их интеграции, оценивать надежность, сопровождаемость, эффективность и другие свойства качества программного обеспечения</p> <p><u>Владеет</u> навыками архитектурного проектирования программного обеспечения, способами приспособления архитектуры к обеспечению требуемых свойств качества программного обеспечения</p>
Организационно-управленческий	ПК-2. Способен использовать методы контроля проекта и контроля версий	ПК-2.1 Демонстрирует знание основных методов контроля проекта и контроля версий	<p><u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологических подходы, принципы структурного и объектно-ориентированного программирования</p> <p><u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта</p> <p><u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу</p>
		ПК-2.2 Организует работы по управлению проектом информационной системы	<p><u>Знает</u> определения понятий project score, project context, заинтересованные лица и роли в команде проекта (Аналитик project/product analyst, руководитель проекта PM, руководитель разработки DM, разработчик D, тестировщик TM, UI/UX специалист, маркетолог MM, технический писатель TW, специалист по анализу данных DS)</p> <p><u>Умеет</u> определять project score и project context для конкретной предметной области</p>

			<i>Владеет</i> навыками управления масштабом проекта и взаимодействия в команде проекта в зависимости от игровой роли в команде проекта
		ПК-2.3 Проводит переговоры и осуществляет контроль версий	<i>Знает</i> методы выявления, сбора и анализа требований заказчика <i>Умеет</i> осуществлять трассировку требований: от высокоуровневых бизнес-целей через пользовательские требования к функциональным требованиям. <i>Владеет</i> навыками управления изменениями требований
Проектный	ПК-8. Способен создавать программные интерфейсы	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<i>Знает</i> способы создания программных интерфейсов <i>Умеет</i> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов <i>Владеет</i> набором операций над структурами данных и операторами языка программирования и методами их использования
		ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<i>Знает</i> определение интуитивно понятного интерфейса <i>Умеет</i> создавать интуитивно понятные интерфейсы <i>Владеет</i> набором средств для создания интуитивно понятного интерфейса
		ПК-8.3 Создает современные программные интерфейсы	<i>Знает</i> свойства современного программного интерфейса <i>Умеет</i> создавать современные программные интерфейсы <i>Владеет</i> методами обеспечения качества создаваемых интерфейсов
Производственно-технологический	ПК-12. Способен применять стандарты и модели жизненного цикла	ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<i>Знает</i> содержание основных стандартов и модели жизненного цикла ПО <i>Умеет</i> работать с документацией, касающейся основных стандартов и моделей жизненного цикла

		<p>ПО</p> <p><u>Владеет</u> навыками работы со стандартами и различными моделями жизненного цикла ПО при создании ПО</p>
	<p>ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения</p>	<p><u>Знает</u> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО</p> <p><u>Умеет</u> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла</p> <p><u>Владеет</u> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области</p>
	<p>ПК-12.3. Применяет стандарты и модели жизненного цикла программного обеспечения</p>	<p><u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы, принципы структурного и объектно-ориентированного программирования, основные понятия COM-технологий и SOA-технологий; тенденции развития современных технологий программирования</p> <p><u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбирать технологический подход к разработке и соответствующие средства автоматизации разработки</p> <p><u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов</p>

## II. Трудоемкость дисциплины и виды учебных занятий по дисциплине

Общая трудоемкость дисциплины составляет 5 зачётных единиц (180 академических часов).

(1 зачетная единица соответствует 36 академическим часам)

## III. 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.	7	16	16	-	-	76	-	зачет
2	Раздел II. Тестирование и отладка программного средства	8	12	24			9	27	экзамен
	Итого:		28	40		-	85	27	Зачет/ экзамен

## IV. СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

### Лекционный материал

**Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.**

**Тема 1 Место дисциплины в профессиональной деятельности ИТ специалиста**

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

## **Тема 2 Модели жизненного цикла разработки программных средств**

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 Принципы проведения и организации инспекций рабочих продуктов**

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.

## **Тема 4 Структура и анализ бизнес-процессов разработки программного обеспечения. Системы отслеживания дефектов**

Workflow системы отслеживания дефектов и сопровождения задач при разработке программных продуктов. Технологический процесс коллективной разработки программ. Основные состояния подзадачи. Система отслеживания дефектов ClearDDTS.

## **Тема 5 Структура бизнес-процессов 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.

## **Тема 6 Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС. (2 час.)**

Роль и место измерений при производстве программных средств. Виды измерений, связанные с процессом разработки ПО. Метрики эффективности процесса производства. Метрики качества продуктов.

## **Раздел 2. Тестирование и отладка программного средства**

### **Тема 7 Модель зрелости организации СММ**

Сущность модели СММ и история её создания. Уровни зрелости: Initial, Repeatable, Defined, Managed, Optimizing. Ключевые виды деятельности каждого уровня. Аттестация предприятий по системе СММ.

## **Тема 8 Стандарты и хорошие практики кодирования на языках высокого уровня**

Стандарты кодирования и их назначение. Категории правил кодирования: запреты, требования, рекомендации. Примеры запретов требований и рекомендаций языка С++. Примеры оформления кода модулей и заголовочных файлов. Revision History. Назначение и примеры чек-листов для разработчиков.

## **Тема 9 Тестирование и отладка программного средства**

Основные понятия, связанные с тестированием. Принципы и виды отладки программного средства. Основные принципы тестирования программного средства. Основные аспекты организации автономного тестирования программных средств. Комплексная отладка программного средства.

## **Тема 10 Основные понятия тестирования программного обеспечения**

Концепция тестирования ПО: формальный подход и доказательство, интерпретационный подход. Основная терминология, применяемая в тестировании ПО. Поиск и исправление ошибок. Организация тестирования: логирование, пошаговое выполнение программы, выполнение с заказными остановками, реверсивное выполнение программы. Три фазы тестирования. Управляющий граф программы. Основные проблемы тестирования.

## **Тема 11 Критерии выбора тестов**

Требования к идеальному критерию тестирования. Классы критериев. Структурные критерии: критерий команд, критерий ветвей и критерий путей. Функциональные критерии: тестирование пунктов спецификации, тестирование классов входных данных, тестирование правил, тестирование классов выходных данных, тестирование функций. Стохастические критерии. Мутационный критерий.

## **Тема 12 Профессиональный слэнг в области программной инженерии**

Основные понятия и жаргонизмы, применяемые профессиональным сообществом.

## V. СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### Лабораторные работы

#### 7 семестр

- Лабораторная работа №1** Организация коллективной работы разработчиков в подгруппах из 9-10 студентов; распределение ролей
- Лабораторная работа №2** Работа с заказчиком программного продукта, объектный анализ и концептуальное моделирование конкретной предметной области
- Лабораторная работа №3** Разработка пользовательских требований и внешних спецификаций к программной системе, подготовка документа «Пользовательские требования»
- Лабораторная работа №4** Разработка системных требований и верхнего уровня проекта программной системы; подготовка документа «Системные требования»
- Лабораторная работа №5** Разработка архитектуры программной системы; подготовка документа «Дизайн проекта»
- Лабораторная работа №6** Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта»
- Лабораторная работа №7** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований»

#### 8 семестр

- Лабораторная работа №1** Разработка списка задач на доработку программной системы по результатам тестирования
- Лабораторная работа №2** Кодирование программной системы согласно разработанному списку задач; подготовка отчета по эффективности обработки задач
- Лабораторная работа №3** Проектирование следующей версии программной системы; разработка требований и конструирование новой версии (4 час.)
- Лабораторная работа №4** Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта»

**Лабораторная работа №5** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований»

**Лабораторная работа №6** Разработка списка задач на доработку программной системы по результатам тестирования

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 студентам необходимо создать аккаунт на портале [github.com](https://github.com). Разработка студенческих проектов ведётся на базе именно этой системы контроля версий с использованием встроенной системы issue трекинга.

В начале учебного семестра студентам необходимо разбиться на группы по 3-5 человек, в каждой команде должен быть назначен руководитель (team-leader), кодировщики и технический писатель. Каждая команда должна определиться с программным проектом, который она будет разрабатывать в течение семестра. Если команда не может выбрать себе проект, то тематику проектов предлагает преподаватель.

### **Задания для самостоятельной работы**

Самостоятельная работа студентов состоит в изучении презентационных материалов по каждой теме, выполнении заданий для самостоятельной работы и работе над итоговым проектом.

Самостоятельная работа первого семестра состоит в разработке программной системы и документации к ней.

Каждый студент отвечает за ту часть документации, которая соответствует его роли, которую он играет в бригаде разработчиков. Результатом работы являются соответствующие документы, написанные в ходе работы по проекту – «План проекта», «Пользовательские требования», «Системные требования», «Архитектура и дизайн», «Тестовые сценарии». Курсовые работы защищаются на собрании бригады разработчиков.

Индивидуальные задания на самостоятельную работу студенты получают в ходе процесса разработки программного продукта в соответствии с теми ролями, которые они играют, с этапом разработки проекта, в зависимости от потребности бригады и в соответствии с планом работы бригады. Распределение заданий производит бригадир под контролем преподавателя.

Примерные виды индивидуальных заданий: подготовка спецификаций подсистемы, разработка тестовых ситуаций, составление набора тестов для одной тестовой ситуации, программирование одной или нескольких функций подсистемы, составление и редактирование документов разработки, инспекция документов разработки, разработка схемы базы данных продукта, проектирование объекта, доклад об инструментальном средстве и т.п.

Контроль самостоятельной работы осуществляется не только преподавателем, но и всей бригадой разработчиков.

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 студентам необходимо создать аккаунт на портале [github.com](https://github.com). Разработка студенческих 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 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 требований и инспектирование кода других кодировщиков.

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

## 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 уровней, что в итоге приводит к развитию навыка самостоятельного планирования и реализации деятельности.

Целью самостоятельной работы студентов является овладение необходимыми компетенциями по своему направлению подготовки, 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;
- подготовка реферативных обзоров источников периодической печати, опорных конспектов, заранее определенных преподавателем;
- поиск информации по теме с последующим ее представлением в аудитории в форме доклада, презентаций;
- подготовка к выполнению аудиторных контрольных работ;
- выполнение домашних контрольных работ;
- выполнение тестовых заданий, решение задач;
- составление кроссвордов, схем;
- подготовка сообщений к выступлению на семинаре, конференции;
- заполнение рабочей тетради;
- написание эссе, курсовой работы;
- подготовка к деловым и ролевым играм;
- составление резюме;
- подготовка к зачетам и экзаменам;
- другие виды деятельности, организуемые и осуществляемые образовательным учреждением и органами студенческого самоуправления.

### **План-график выполнения самостоятельной работы по дисциплине**

7 семестр

<b>№ п/п</b>	<b>Дата/сроки выполнения</b>	<b>Вид самостоятельной работы</b>	<b>Примерные нормы времени на</b>	<b>Форма контроля</b>
--------------	------------------------------	-----------------------------------	-----------------------------------	-----------------------

			<b>выполнение</b>	
1	1-2 неделя	Создание и настройка репозитория в системе GitHub	5	ПР-1
2	3-4 неделя	Разработка презентации проекта Создайте маркетинговые требования к проекту (Marketing Requirements Specification - MRS) в виде презентации рекламного характера. Разместите её в папке REQUIREMENS Вашего репозитория.	5	ПР-1
3	5-6 неделя	Разработка плана проекта Создайте документ "Software Project Plan" согласно образца и разместите его в своём репозитории	5	ПР-1
4	7-8 неделя	Разработка требований Создайте и разместите в репозитории документ "Software Requirements Specification". В документе опишите первую версию требований к Вашему продукту.	5	ПР-1
5	9-10 неделя	Разработка дизайна проекта Разработайте и разместите в репозитории документ Software Disign Specification. Опишите в нем первую версию дизайна (архитектуры) Вашего проекта.	5	ПР-1
6	11-12 неделя	Кодирование проекта Создайте и разместите в репозитории код Вашего проекта. Создайте документ Code Issue List - список задач, выполняемых на этапе кодирования	5	ПР-1
7	13-14 неделя	Тестирование проекта Создайте и разместите в репозитории Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов Создайте и разместите в репозитории документ Requirements Tracaebility Matrix - матрица покрытия тестами требований Создайте и разместите в репозитории Test Logs - документ, который заполняется в ходе тестирования и	5	ПР-1

		содержит результаты выполнения тестов		
8	15-18 неделя	Презентация проекта	5	Защита проекта
	Итого:		40	

## 8 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-2 неделя	Подготовка к семинару «Профессиональный слэнг в области программной инженерии» Ориентируясь на лекцию 12, вспомните и опишите десяток профессиональных жаргонизмов, используемых программистской средой.	5	ПР-1
2	3-4 неделя	Подготовка к семинарам «Введение в технологию программирования Agile и разбор Agile текстов», «Agile критика традиционных технологий программирования»	5	ПР-1
3	5-6 неделя	Подготовка к семинарам Принципы Agile, Роли Agile.	5	ПР-1
4	7-8 неделя	Подготовка к семинарам Артефакты Agile, Методы Agile	5	ПР-1
5	9-10 неделя	Подготовка к семинарам Работа с Agile командами. Оценка Agile подхода	5	ПР-1
6	11-12 неделя	Подготовка к семинарам Запахи кода. «Раздувальщики», Запахи кода. Нарушители объектного дизайна	5	ПР-1
7	13-14 неделя	Подготовка к семинарам Приёмы методов рефакторинга. Перемещение функций между объектами	2	ПР-1
8	15-18 неделя	Подготовка к экзамену	36	Экзамен
	Итого:		68	

## Требования к представлению и оформлению результатов самостоятельной работы

### 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 (см. образец).

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 - докумен, 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.

Самостоятельная работа второго семестра состоит в подготовке к семинарам.

## VII. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код и наименование индикатора достижения	Результаты обучения	Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	ПК-1.1 Демонстрирует знание концептуальных моделей менеджмента	<p><u>Знает</u> основные понятия концептуальных моделей: система, элемент, часть, модель, предметная область, состояние, предусловие, постусловие, переход, процесс, триггер, сущность, экземпляр, отношение, атрибут и т. д.</p> <p><u>Умеет</u> различать концептуальные модели (модели потоков данных, сущностных отношений, переходов состояний и т. д.), объяснять их назначения, основные особенности, достоинства и недостатки каждой модели</p> <p><u>Владеет</u> навыками создания концептуальных моделей менеджмента для выбранных предметных областей с использованием языков концептуального моделирования</p>	ПР1 лабораторные работы и задания для самостоятельного выполнения	Зачет, вопросы № 1-22

			(UML, IDEF, язык прикладной логики и другие)		
		ПК-1.2 Использует основные модели менеджмента в управлении	<u>Знает</u> модель «Тройственной ограниченности», в которой ограничениями являются Score, Schedule, Cost + Quality		
			<u>Умеет</u> применять модель «Тройственной ограниченности» в управлении проектами в конкретных предметных областях		
			<u>Владеет</u> навыками варьирования параметров ограничений в управлении проектами в конкретных предметных областях		
		ПК-1.3 Применяет модели и методы менеджмента в управлении программным обеспечением	<u>Знает</u> методы анализа требований и проектирования программного обеспечения, модели проектирования и модели реализации, стратегии и методы испытаний и метрики оценки качества программного обеспечения и его компонентов		

			<p><u>Умеет</u> оценивать применимость разных методов архитектурного проектирования к особенностям создаваемого программного обеспечения, оценивать архитектурный проект с помощью метрик качества, выполнить планирование испытаний отдельных программных единиц архитектуры и их интеграции, оценивать надежность, сопровождаемость, эффективность и другие свойства качества программного обеспечения</p>		
			<p><u>Владеет</u> навыками архитектурного проектирования программного обеспечения, способами приспособления архитектуры к обеспечению требуемых свойств качества программного обеспечения</p>		
		<p>ПК-2.1 Демонстрирует знание основных методов контроля проекта и контроля версий</p>	<p><u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологических подходы, принципы структурного и объектно-</p>		

			<p>ориентированного программирования</p>		
			<p><i>Умеет</i> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта</p>		
			<p><i>Владеет</i> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу</p>		
		<p>ПК-2.2 Организует работы по управлению проектом информационной системы</p>	<p><i>Знает</i> определения понятий project score, project context, заинтересованные лица и роли в команде проекта (Аналитик project/product analyst, руководитель проекта PM, руководитель разработки DM, разработчик D, тестировщик TM, UI/UX специалист, маркетолог MM, технический писатель TW, специалист по анализу данных DS)</p>		
			<p><i>Умеет</i> определять project score и project context для конкретной предметной области</p>		

			<i>Владеет</i> навыками управления масштабом проекта и взаимодействия в команде проекта в зависимости от играемой роли в команде проекта		
		ПК-2.3 Проводит переговоры и осуществляет контроль версий	<i>Знает</i> методы выявления, сбора и анализа требований заказчика		
			<i>Умеет</i> осуществлять трассировку требований: от высокоуровневых бизнес-целей через пользовательские требования к функциональным требованиям.		
			<i>Владеет</i> навыками управления изменениями требований		
2	Раздел 2. Тестирование и отладка программного средства	ПК-8.1 Демонстрирует знание способов создания программных интерфейсов	<i>Знает</i> способы создания программных интерфейсов	ПР1 лабораторные работы и задания для самостоятельного выполнения	Экзамен, вопросы № 1-11
			<i>Умеет</i> использовать структуры данных и операторы языка программирования для разработки программных интерфейсов		
			<i>Владеет</i> набором операций над структурами данных и операторами языка программирования и методами их использования		

		ПК-8.2 Использует методы создания интуитивно понятных программных интерфейсов	<u>Знает</u> определение интуитивно понятного интерфейса		
	<u>Умеет</u> создавать интуитивно понятные интерфейсы				
	<u>Владеет</u> набором средств для создания интуитивно понятного интерфейса				
		ПК-8.3 Создает современные программные интерфейсы	<u>Знает</u> свойства современного программного интерфейса		
	<u>Умеет</u> создавать современные программные интерфейсы				
	<u>Владеет</u> методами обеспечения качества создаваемых интерфейсов				
		ПК-12.1 Демонстрирует знание стандартов и моделей жизненного цикла программного обеспечения	<u>Знает</u> содержание основных стандартов и модели жизненного цикла ПО	ПР1 лабораторные работы и задания для самостоятельного выполнения	
	<u>Умеет</u> работать с документацией, касающейся основных стандартов и моделей жизненного цикла ПО				
	<u>Владеет</u> навыками работы со стандартами и				

			различными моделями жизненного цикла ПО при создании ПО		
		ПК-12.2 Использует модели жизненного цикла при создании программного обеспечения	<u>Знает</u> терминологию, нотацию различных групп моделей жизненного цикла при создании ПО		
			<u>Умеет</u> планировать деятельность по созданию ПО в рамках выбранной модели жизненного цикла		
			<u>Владеет</u> навыками создания ПО в рамках выбранной модели жизненного цикла для определенной предметной области		
		ПК-12.3. Применяет стандарты и модели жизненного цикла программного обеспечения	<u>Знает</u> основные группы моделей жизненного цикла к созданию программного обеспечения, их процессы, виды деятельности и задачи, основные технологические подходы, принципы структурного и объектно-ориентированного программирования, основные понятия СОМ-технологий и SOA-технологий; тенденции развития современных технологий программирования		

			<p><u>Умеет</u> выбирать модель жизненного цикла для разработки программного обеспечения, в зависимости от масштаба проекта, стабильности требований, сроков получения стабильных версий программного обеспечения, выбирать технологический подход к разработке и соответствующие средства автоматизации разработки</p> <p><u>Владеет</u> навыками адаптации видов деятельности выбранной модели жизненного цикла разработки программного обеспечения к выбранному технологическому подходу и конкретизации задач по созданию соответствующих технологических артефактов</p>		
--	--	--	--	--	--

\*Рекомендуемые формы оценочных средств:

1) собеседование (УО-1), коллоквиум (УО-2); доклад, сообщение (УО-3); круглый стол, дискуссия, полемика, диспут, дебаты (УО-4); и т.д.

2) тесты (ПР-1); контрольные работы (ПР-2), эссе (ПР-3), рефераты (ПР-4), курсовые работы (ПР-5), научно-учебные отчеты по практикам (ПР-6); лабораторная работа (ПР-7); портфолио (ПР-8); проект (ПР-9); деловая и/или ролевая игра (ПР-10); кейс-задача (ПР-11); рабочая тетрадь (ПР-12) и т.д.

3) тренажер (ТС-1) и т.д.

## VIII. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Основная литература

1. Гагарина Л.Г., Виснадул Б.Д., Кокорева Е.В. Технология разработки программного обеспечения. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с.  
<http://znanium.com/go.php?id=389963>
2. Терехов А.Н. Технология программирования: учебное пособие. Москва : Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2014. 148 с.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:797542&theme=FEFU>
3. Программная инженерия: учебник для вузов / [В.А. Антипов, А.А. Бубнов, А.Н. Пылькин и др.]; под ред. Б.Г. Трусова. – М.: Академия, 2014. – 282 с.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:790423&theme=FEFU>
4. Кознов Д.В. Введение в программную инженерию - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 306 с.  
<http://www.iprbookshop.ru/52146.html?replacement=1>

### Дополнительная литература

*(печатные и электронные издания)*

1. Котляров В.П., Коликова Т.В. Основы тестирования программного обеспечения. М.: Лаборатория знаний. 2012.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:668103&theme=FEFU>
2. Плаксин, М. А. Тестирование и отладка программ для профессионалов будущих и настоящих [Электронный ресурс] / М. А. Плаксин. - 2-е изд. (эл.). - М.: БИНОМ. Лаборатория знаний, 2013. - 167 с.: ил. - ISBN 978-5-9963-0946-7. <http://znanium.com/catalog.php?bookinfo=476321>
3. Майерс Г., Сандлер К., Баджет Т. Искусство тестирования программ. 3-е изд. Пер. с англ. М: Диалектика, Вильямс. 2020. 272 с.

## **Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. <http://www.martinfowler.com/> Martin Fowler
2. <https://refactoring.guru/ru/catalog> Каталог рефакторинга
3. <http://www.martinfowler.com/eaCatalog/index.html> Catalog of Patterns of Enterprise Application Architecture
4. <http://gameprogrammingpatterns.com/contents.html> Game Programming Patterns

### **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.

### **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/>

## IX. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Успешное освоение дисциплины предполагает активную работу студентов на всех занятиях аудиторной формы: лекциях и практиках, выполнение аттестационных мероприятий. В процессе изучения дисциплины студенту необходимо ориентироваться на проработку лекционного материала, подготовку к практическим занятиям, выполнение контрольных и творческих работ.

Освоение дисциплины « » предполагает рейтинговую систему оценки знаний студентов и предусматривает со стороны преподавателя текущий контроль за посещением студентами лекций, подготовкой и выполнением всех практических заданий, выполнением всех видов самостоятельной работы.

Промежуточной аттестацией по дисциплине *«Технологии коллективной разработки информационных систем»* является зачет/экзамен.

Студент считается аттестованным по дисциплине при условии выполнения всех видов текущего контроля и самостоятельной работы, предусмотренных учебной программой.

Шкала оценивания сформированности образовательных результатов по дисциплине представлена в фонде оценочных средств (ФОС).

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.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в портале BlackBoard 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. Студенты представляют результаты лабораторной работы преподавателю и друг другу, при этом стараются максимально обмениваться опытом.

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.

К промежуточной аттестации допускаются обучающиеся, выполнившие все задания (контрольные, самостоятельные), предусмотренные учебной программой дисциплины, посетившие не менее 85% аудиторных занятий.

## **МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

### **Материально-техническое и программное обеспечение дисциплины**

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус D,	Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716	1С Предприятие (8.2), 7-Zip, ABBYY Lingvo 12, Alice 3, Anaconda 3, Autodesk, CodeBlocks, CorelDRAW X7, Dia, Directum 4.8, DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git, Greenfoot, gsview, Inscapе 0.91, Java, Java development Kit, Kaspersky, Lazarus, LibreOffice 4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight Microsoft System Center, Microsoft

<p>ауд. D 733,733а. Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</p>	<p>ССВА – 1 шт. Доска аудиторная, Моноблок Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 (13 шт.) и аудиовизуальными средствами проектор Panasonic DLPPjectorPT-D2110XE</p>	<p>Visual Studio 2012, MikTeX 2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Python 3.4, Python2.7(3.4, 3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity 2017.3.1f1, Veusz, Vim 8.1, Visual Paradigm CE, Windows Kits, Windows Phone SDK 8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central CS3, Adobe Extend Script Toolkit 2, Adobe Photoshop CS3, DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox, Windows Media Center, WinSCP</p>
--	--	--

### **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](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](https://portableapps.com/support/portable_app#using) ;

DiagramDesigner - пакет программ для создания потоковых диаграмм, диаграмм классов UML, иллюстраций и др. диаграмм: <https://www.fosshub.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/> .

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.