



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

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**

«СОГЛАСОВАНО»

Руководитель ОП

 Артемьева И.Л.

«УТВЕРЖДАЮ»

И.о. директора департамента



Смагин С.В.

«15» июля 2021 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Технологии коллективной промышленной разработки информационных систем

**Направление подготовки 09.03.04 Программная инженерия**

(Программная инженерия)

**Форма подготовки очная**

курс 4 семестр 7, 8

лекции 24 час.

практические занятия 0 час.

лабораторные работы 34 час.

в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 34 час.

всего часов аудиторной нагрузки – 58 час.

в том числе с использованием МАО – 34 час.

самостоятельная работа 122 час.

контрольные работы нет

курсовая работа / курсовой проект – 7 семестр

экзамен - 8 семестр

зачёт – 7 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 09.03.04 Программная инженерия утвержденного приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 920 (с изменениями и дополнениями)

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 12.1 от «25» декабря 2019 г.

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

Составители: зав. кафедрой ПММУиПО И.Л.Артемьева д.т.н., профессор  
профессор кафедры ПММУиПО д.т.н., доцент

**Владивосток  
2021**

**Оборотная сторона титульного листа РПД**

**I. Рабочая программа пересмотрена на заседании кафедры прикладной математики, механики, управления и программного обеспечения:**

Протокол от «09» июля 2021 г. № 7.1

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

**II. Рабочая программа пересмотрена на заседании департамента программной инженерии и искусственного интеллекта:**

Протокол от «17» сентября 2021 г. № 9.1

И.о. директора департамента \_\_\_\_\_ Смагин С.В.  
(подпись) (И.О. Фамилия)

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

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

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

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

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

Директор департамента \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**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

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

Универсальные компетенции выпускников и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции выпускника	Код и наименование индикатора достижения универсальной компетенции
Командная работа и лидерство	УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1. Знает различные приемы и способы социализации личности и социального взаимодействия. УК-3.2. Умеет строить отношения с окружающими людьми, с коллегами. УК-3.3. Имеет практический опыт участия в командной работе, в социальных проектах,

		распределения ролей в условиях командного взаимодействия.
--	--	---

Профессиональные компетенции выпускников и индикаторы их достижения:

Задача профессиональной деятельности	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Основание (ПС, анализ иных требований, предъявляемых к выпускникам)
Тип задач профессиональной деятельности: организационно-управленческий				
Участие в проведении переговоров с заказчиком и презентация проектов; участие в организации работ по управлению проектом ИС; участие в организации информационно-телекоммуникационной инфраструктуры и управлении информационной безопасностью ИС; участие в организации и управлении информационными ресурсами и сервисами	Прикладные и информационные процессы. Информационные технологии. Программное обеспечение	ПК-1. Владение Классическими концепциями и моделями менеджмента в управлении проектами	ПК-1.1. Знает концептуальные модели менеджмента ПК-1.2. Умеет использовать основные модели менеджмента в управлении ПК-1.3. Имеет навыки практического применения моделей и методов менеджмента в управлении ПО	06.022 Системный аналитик
		ПК-2. Владение методами контроля проекта и готовностью осуществлять контроль версий	ПК-2.1. Знает основные методы информационной безопасности ИС ПК-2.2. Умеет организовать работы по управлению проектом ИС ПК-2.3. Имеет навыки в проведении переговоров и способен осуществлять контроль версий	
Тип задач профессиональной деятельности: проектный				
Формирование требований	Прикладные и	ПК-8. Способность создавать	ПК-8.1. Знает способы создания	06.028 Системный

<p>к информатизации и автоматизации и прикладных процессов, формализация предметной области проекта; технико-экономическое обоснование проектных решений и составление технического задания на разработку программного продукта; проектирование программно-аппаратных средств в соответствии с техническим заданием; применение современных инструментальных средств при разработке программного обеспечения; документирование компонентов информационной системы на стадии жизненного цикла</p>	<p>информационные процессы. Информационные технологии и. Программное обеспечение</p>	<p>программные интерфейсы</p>	<p>программных интерфейсов ПК-8.2. Умеет создавать интуитивно понятные программные интерфейсы ПК-8.3. Имеет навыки в создании современных программных интерфейсов</p>	<p>программист 06.004          Специалист по тестированию в области информационных технологий 06.001          Программист</p>
<p>Тип задач профессиональной деятельности: производственно-технологический</p>				

Проведение работ по инсталляции программного обеспечения автоматизированных систем и загрузки баз данных; настройка параметров ИС и тестирование результатов настройки; ведение технической документации; техническое сопровождение ИС в процессе эксплуатации; применение Web технологий при реализации удаленного доступа в системах клиент – сервер и распределенных вычислений	Программное обеспечение	ПК-12. Владение стандартами и моделями жизненного цикла	ПК-12.1. Знает стандарты и модели жизненного цикла ПО ПК-12.2. Умеет использовать модели жизненного цикла ПО ПК-12.3. Имеет навыки применения стандартов и моделей жизненного цикла ПО	06.028 Системный программист 06.022 Системный аналитик 06.004 Специалист по тестированию в области информационных технологий 06.001 Программист ПО
--	-------------------------	---	--	---

Для формирования вышеуказанных компетенций в рамках дисциплины «Технология коллективной промышленной разработки информационных систем» применяются следующие методы активного/ интерактивного обучения: метод проектов, дискуссия, презентация.

# **I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА**

## **Лекционный материал (24 часов)**

### **Тема 1 Место дисциплины в профессиональной деятельности ИТ специалиста (2 час.)**

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 (2 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 Принципы проведения и организации инспекций рабочих продуктов (2 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.

### **Тема 4 Структура и анализ бизнес-процессов разработки программного обеспечения. Системы отслеживания дефектов. (2 час.)**

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 (2 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.

**Тема 7 Модель зрелости организации CMM (2 час.)**

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

**Тема 8 Стандарты и хорошие практики кодирования на языках высокого уровня (2 час.)**

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

**Тема 9 Тестирование и отладка программного средства (2 час.)**

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 (2 час.)**

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 (2 час.)**

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 (2час.)**

Основные понятия и жаргонизмы, применяемые профессиональным сообществом.

## **II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА**

### **Практические занятия (0 часов)**

Не предусмотрены учебным планом

### **Лабораторные работы (34 час.)**

Лабораторные работы проводятся в виде 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 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.

### **7 семестр**

**Лабораторная работа №1** Организация коллективной работы разработчиков в подгруппах из 9-10 студентов; распределение ролей (2 час.).

**Лабораторная работа №2** Работа с заказчиком программного продукта, объектный анализ и концептуальное моделирование конкретной предметной области (4 час.).

**Лабораторная работа №3** Разработка пользовательских требований и внешних спецификаций к программной системе, подготовка документа «Пользовательские требования» (3 час.)

**Лабораторная работа №4** Разработка системных требований и верхнего уровня проекта программной системы; подготовка документа «Системные требования» (3 час.)

**Лабораторная работа №5** Разработка архитектуры программной системы; подготовка документа «Дизайн проекта» (3 час)

**Лабораторная работа №6** Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта» (3 час)

**Лабораторная работа №7** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (3 час)

## **8 семестр**

**Лабораторная работа №1** Разработка списка задач на доработку программной системы по результатам тестирования (2 час.)

**Лабораторная работа №2** Кодирование программной системы согласно разработанному списку задач; подготовка отчета по эффективности обработки задач (2 час.)

**Лабораторная работа №3** Проектирование следующей версии программной системы; разработка требований и конструирование новой версии (3 час.)

**Лабораторная работа №4** Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта» (3 час)

**Лабораторная работа №5** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (3 час)

**Лабораторная работа №6** Разработка списка задач на доработку программной системы по результатам тестирования (3 час.)

**Лабораторная работа №7** Кодирование программной системы согласно разработанному списку задач; подготовка отчета по эффективности обработки задач (2 час.)

### Ш. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

На самостоятельную работу студентов отводится 122 час, в том числе на подготовку к экзамену 27 часов.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Технология коллективной разработки информационных систем» представлено в Приложении 1 и включает в себя: план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию; характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению; требования к представлению и оформлению результатов самостоятельной работы; критерии оценки выполнения самостоятельной работы.

### IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	УК3, ПК1, ПК2, ПК8, ПК12	Знает Умеет, владеет	ПР1 лабораторные работы и задания для самостоятельного выполнения	Зачет, вопросы № 1-22
2	Раздел 2. Тестирование и отладка программного средства	УК3, ПК1, ПК2, ПК8, ПК12	Знает Умеет, владеет	ПР1 лабораторные работы и задания для самостоятельного выполнения	

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

## **V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

*(электронные и печатные издания)*

1. Мейер Б. Почувствуй класс. М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2011. 775с. <http://www.iprbookshop.ru/22435>.— ЭБС «IPRbooks»
2. 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>
3. 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>
4. 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>
5. 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>
6. 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. Сеницын, С.В. Верификация программного обеспечения: учебное пособие/ С.В. Сеницын, 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. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004. <http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
3. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002. <http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
4. 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>

5. Майерс Г. Искусство тестирования программ. 3-е изд. Пер. с англ. М: Диалектика, Вильямс. 2012. 272 с.
6. Липаев, В.В. Программная инженерия. Методологические основы [Текст] : Учеб. / В. В. Липаев ; Гос. ун-т — Высшая школа экономики. — М. : ТЕИС, 2006. — 608 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**

Занятия проводятся с использованием проектора и мультимедийного комплекса для демонстрации мультимедийного контента внутренней системы портала ДВФУ. Лабораторные занятия проводятся в специализированном компьютерном классе. Для написания программ используется свободное программное обеспечение:

Система контроля версий Git  
Сервис GitHub

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Дисциплина изучается в следующих организационных формах: лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального задания; выполнение группового проекта (в рамках самостоятельной работы); индивидуальные и групповые консультации.

Основной формой самостоятельной работы студента является изучение теоретического материала, его дополнение рекомендованной литературой,

выполнение индивидуальных заданий и группового 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.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 и аудиовизуальными средствами проектор Panasonic DLPProjectorPT-D2110XE, плазма LG FLATRON M4716SSBAM4716SJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Дальневосточный федеральный университет»  
(ДФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ  
ОБУЧАЮЩИХСЯ**

по дисциплине «Технологии коллективной промышленной разработки  
информационных систем»

**Направление подготовки – 09.03.04 Программная инженерия**

**Форма подготовки (очная)**

## План-график выполнения самостоятельной работы по дисциплине

7 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Форма контроля
1	1 неделя	Создание и настройка репозитория в системе GitHub	Проверка задания
2	2-3 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.	Проверка задания
3	4-5 неделя	Разработка плана проекта Создайте документ "Software Project Plan" согласно образца и разместите его в своём репозитории	Проверка задания
4	6-7 неделя	Разработка требований Создайте и разместите в репозитории документ "Software Requirements Specification". В документе опишите первую версию требований к Вашему продукту.	Проверка задания
5	7-8 неделя	Разработка дизайна проекта Разработайте и разместите в репозитории документ Software Disign Specification. Опишите в нем первую версию дизайна (архитектуры) Вашего проекта.	Проверка задания
6	9-15 неделя	Кодирование проекта Создайте и разместите в репозитории код Вашего проекта. Создайте документ Code Issue List - список задач, выполняемых на этапе кодирования	Проверка задания
7	16-17 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 содержит результаты выполнения тестов	Проверка задания
8	18 неделя	Презентация проекта	Защита проекта



## 8 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Форма контроля
1	1 неделя	Подготовка к семинару «Профессиональный слэнг в области программной инженерии» Ориентируясь на лекцию 12, вспомните и опишите десяток профессиональных жаргонизмов, используемых программистской средой.	Выступление на семинаре
2	2 неделя	Подготовка к семинарам «Логирование приложений Java»	Выступление на семинаре
3	3 неделя	Подготовка к семинарам «Автоматизация модульного тестирования и контроля версий приложений Java»	Выступление на семинаре
4	4 неделя	Подготовка к семинарам «Введение в технологию программирования Agile и разбор Agile текстов», «Agile критика традиционных технологий программирования»	Выступление на семинаре
5	5 неделя	Подготовка к семинарам Принципы Agile, Роли Agile.	Выступление на семинаре
6	6 неделя	Подготовка к семинарам Практики Agile: производственные процессы, Практики Agile: технические	Выступление на семинаре
7	7 неделя	Подготовка к семинарам Артефакты Agile, Методы Agile	Выступление на семинаре
8	8 неделя	Подготовка к семинарам Работа с Agile командами. Оценка Agile подхода,	Выступление на семинаре
9	9 неделя	Подготовка к семинарам Запахи кода. «Раздувальщики», Запахи кода. Нарушители объектного дизайна	Выступление на семинаре
10	10 неделя	Подготовка к семинарам Запахи кода. Утяжелители изменений, Запахи кода. Замусориватели	Выступление на семинаре
11	11 неделя	Подготовка к семинарам Запахи кода. Опутыватели связями,	Выступление на семинаре
12	12 неделя	Подготовка к семинарам Приёмы методов рефакторинга. Перемещение функций между объектами.	Выступление на семинаре
13	13 неделя	Подготовка к семинарам Приёмы методов рефакторинга. Организация данных	Выступление на семинаре
14	14 неделя	Подготовка к семинарам Приёмы методов рефакторинга. Упрощение условных выражений	Выступление на семинаре

15	15 неделя	Подготовка к семинарам Приёмы рефакторинга. Решение задач обобщения	Выступление на семинаре
16	16 неделя	Подготовка к семинарам Возможности рефакторинга в интегрированных средах Составление методов.	Выступление на семинаре
17	17-18 неделя	Проектирование и определение метрик проектов.	Проверка задания

### **Рекомендации по самостоятельной работе студентов**

На самостоятельную работу студентов отводится 99 час, в том числе на подготовку к экзамену 27 часов.

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.

## **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

`REQUIREMENS` - 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.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Дальневосточный федеральный университет»  
(ДФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине «Технологии коллективной промышленной разработки информационных  
систем»

**Направление подготовки – 09.03.04 Программная инженерия**

**Форма подготовки (очная)**

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	УК3, ПК1, ПК2, ПК8, ПК12	Знает	ПР1	Зачет, вопросы № 1-22
			Умеет, владеет	лабораторные работы и задания для самостоятельного выполнения	
2	Раздел 2. Тестирование и отладка программного средства	УК3, ПК1, ПК2, ПК8, ПК12	Знает	ПР1	Экзамен, вопросы № 1-11
			Умеет, владеет	лабораторные работы и задания для самостоятельного выполнения	

### Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
УК-3 способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	знает (пороговый уровень)	Знает различные приемы и способы социализации личности и социального взаимодействия	Знание методов организации работы в коллективах разработчиков ПО	Способность организовать работу коллектива разработчиков из не менее чем 4 человек
	умеет (продвинутой)	Умеет строить отношения с окружающими людьми, с коллегами.	Умение использовать знания методов организации работы в коллективах разработчиков ПО.	Способность организовать работу коллектива разработчиков из не менее чем 4 человек
	владеет (высокий)	Имеет практический опыт участия в командной работе, в социальных проектах, распределения ролей в	Умение организовать работу в коллективах разработчиков ПО	Способность организовать работу коллектива разработчиков из не менее чем 4 человек

		условия командного взаимодействия		
ПК-1 владение классическим и концепциями и моделями менеджмента в управлении проектами	знает (пороговый уровень)	Знает концептуальные модели менеджмента	Знание основы жизненного цикла разработки приложений	Способность сформулировать не менее 4 этапов жизненного цикла
	умеет (продвинутой)	Умеет использовать основные модели менеджмента в управлении	Умение разрабатывать архитектуру программных средств и реализовывать программные средства	Способность описать архитектуру программы, состоящую из не менее чем 5 программных модулей
	владеет (высокий)	Имеет навыки практического применения моделей и методов менеджмента в управлении ПО	Умение разрабатывать и презентовать программные средства	Способность разрабатывать программные средства объемом не менее 1500 строк. Способность презентовать работу не менее чем на 20 слайдах в течение не менее чем 15 минут
ПК-2 Владение методами контроля проекта и готовностью осуществлять контроль версий	знает (пороговый уровень)	Знает основные методы информационной безопасности ИС	Знание методов	Не менее 2
	умеет (продвинутой)	Умеет организовать работы по управлению проектом ИС	Умение сформулировать техническое задание	Не менее чем 1 страницу
	владеет (высокий)	Имеет навыки в проведении переговоров и способен осуществлять контроль версий	Умение разрабатывать и презентовать программные средства	Способность разрабатывать программные средства объемом не менее 1500 строк. Способность презентовать работу не менее чем на 20

				слайдах в течение не менее чем 15 минут
ПК-8 Способность создавать программные интерфейсы	знает (пороговый уровень)	Знает способы создания программных интерфейсов	Знание средств, поддерживающих создание программного обеспечения	Способность работать с не менее чем 4 средствами
	умеет (продвинутой)	Умеет создавать интуитивно понятные программные интерфейсы	Умение разрабатывать архитектуру программных средств и реализовывать программные средства	Способность описать архитектуру программы, состоящую из не менее чем 10 программных модулей
	владеет (высокий)	Имеет навыки в создании современных программных интерфейсов	Умение разрабатывать программные средства	Способность разрабатывать программные средства объемом не менее 1500 строк.
ПК-12 Владение стандартами и моделями жизненного цикла	знает (пороговый уровень)	Знает стандарты и модели жизненного цикла ПО	Знание основ групповой динамики, психологии и профессионального поведения, специфичных для программной инженерии	Знание не менее двух принципов групповой динамики
	умеет (продвинутой)	Умеет использовать модели жизненного цикла ПО	Умение использовать знания основ групповой динамики, психологии и профессионального поведения, специфичных для программной инженерии	Опыт работы в команде при разработке не менее чем одного коллективного проекта
	владеет (высокий)	Имеет навыки применения стандартов и моделей жизненного цикла ПО	навыками применения знаний основ групповой динамики, психологии и профессиональн	Опыт работы в команде при разработке не менее чем одного коллективного проекта



			ого поведения, специфичных для программной инженерии	
--	--	--	--	--

## **Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины**

### **Текущий контроль**

Состоит в проверке правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

### **Критерии оценки проектов**

- 100-86 баллов выставляется, если студент/группа точно определили содержание и составляющие части задания, умеют аргументированно отвечать на вопросы, связанные с заданием. Продемонстрировано знание и владение навыками самостоятельной исследовательской работы по теме. Фактических ошибок, связанных с пониманием проблемы, нет.

- 85-76 - баллов - работа студента/группы характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет.

- 75-61 балл – проведен достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимание базовых основ и теоретического обоснования выбранной темы. Привлечены основные источники по рассматриваемой теме. Допущено не более 2 ошибок в смысле или содержании проблемы

- 60-50 баллов - если работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Допущено три или более трех ошибок смыслового содержания раскрываемой проблемы

### **Шкала оценивания**

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно

От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

**Оценочные средства для промежуточной аттестации**  
**Вопросы к зачету (7 семестр)**

1. В каких условиях лучше всего применять итеративно-инкрементную модель жизненного цикла ПО и почему.
2. Какой проектной документацией принято сопровождать процесс разработки программного обеспечения.
3. Опишите функции участников инспекции на каждом этапе.
4. Зачем необходимо собирать метрики по результатам инспекций и какие метрики существуют.
5. Опишите состояния, в которых может пребывать подзадача в процессе коллективной разработки программного продукта.
6. Для чего служит система CLEARDDTS.
7. Какие существуют проектные виды деятельности.
8. Назовите этапы регистрации затрат рабочего времени.
9. В чем состоит методика GQ(I)M.
10. Приведите примеры и описание метрик качества продуктов.
11. Опишите сущность модели СММ.
12. Какие ключевые виды деятельности должны иметь место в компании третьего уровня.
13. Зачем необходимы стандарты кодирования.
14. Приведите пример правил и рекомендаций по оформлению заголовков файлов и функций.
15. Опишите основные принципы тестирования программного средства.
16. Основная идея восходящего тестирования, достоинства и недостатки.
17. Что включает в себя стандартный процесс разработки ПО согласно современным парадигмам?
18. Назовите модели жизненных циклов разработки ПО, опишите подробно (основной принцип, достоинства и недостатки) любой из них.

19. Перечислите роли участников инспекции рабочих продуктов и опишите их, напишите функции одного из них на каждом этапе инспекции.
20. Опишите способ оценивания инспекций и приведите пример оценивания.
21. Опишите систему ClearDdts в 5-8 предложениях.
22. Какие понятия лежат в основе технологии Workflow.

### **Вопросы к экзамену (8 семестр)**

Тестовые вопросы размещены на портале BlackBoard.

[https://bb.dvfu.ru/webapps/blackboard/content/listContentEditable.jsp?content\\_id=172372\\_1&course\\_id=5025\\_1&mode=reset](https://bb.dvfu.ru/webapps/blackboard/content/listContentEditable.jsp?content_id=172372_1&course_id=5025_1&mode=reset)

Ниже приведены примеры вопросов к экзамену.

1. Программой измерений компании называется

- 1) \*комплекс мероприятий, направленных на количественную оценку эффективности работы компании;
- 2) перечень метрик, применяемых в компании при разработке программных продуктов;
- 3) перечень метрик, применяемых в компании при сопровождении программных продуктов.

2. С измерениями связано понятие индикатора – формы представления набора значений метрики, удобной для анализа. При этом тип индикатора, называемый «индикатор-цель» (Success) характеризуется тем, что

- 1) \* должно быть достигнуто некоторое целевое значение метрики (например, успешно пройдено 95% тестов);
- 2) отслеживается развитие процесса во времени (например, число найденных ошибок увеличивается или уменьшается);
- 3) используется непосредственный набор значений метрики с целью изучения объекта измерений;

3. С измерениями связано понятие индикатора – формы представления набора значений метрики, удобной для анализа. При этом тип индикатора, называемый «индикатор-прогресс» (Progress) характеризуется тем, что

- 1) должно быть достигнуто некоторое целевое значение метрики (например, успешно пройдено 95% тестов);
- 2) \*отслеживается развитие процесса во времени (например, число найденных ошибок увеличивается или уменьшается);

3) используется непосредственный набор значений метрики с целью изучения объекта измерений;

4. С измерениями связано понятие индикатора – формы представления набора значений метрики, удобной для анализа. При этом тип индикатора, называемый «индикатор-анализ» (Analysis) характеризуется тем, что

1) должно быть достигнуто некоторое целевое значение метрики (например, успешно пройдено 95% тестов);

2) отслеживается развитие процесса во времени (например, число найденных ошибок увеличивается или уменьшается);

3) \*используется непосредственный набор значений метрики с целью изучения объекта измерений;

5. Метрика On project % time (OPPT) (процент времени, затрачиваемый на работу по проектам) рассчитывается по формуле

1) \*  $OPPT = (\text{Рабочее время, затраченное на проект} / \text{Общее рабочее время}) * 100\%$

2)  $OPPT = \text{Размер продукта} / \text{Общее время инспектирования};$

3)  $OPPT = (\text{Количество найденных ошибок} / \text{Размер рабочего продукта}).$

6. Метрика Inspection Fault Density (IFD) (плотность обнаруженных ошибок) рассчитывается по формуле

1)  $IFD = (\text{Рабочее время, затраченное на проект} / \text{Общее рабочее время}) * 100\%$

2) \*  $IFD = (\text{Количество найденных ошибок} / \text{Размер рабочего продукта})$

3)  $IFD = \text{Размер продукта} / \text{Общее время инспектирования}$

7. Метрика Inspection Preparation Rate (IPR) (производительность подготовки к инспекциям) рассчитывается по формуле

1) \*  $IPR = (\text{Количество инспекторов} * \text{Размер продукта}) / \text{Общее время подготовки};$

2)  $IPR = (\text{Рабочее время, затраченное на проект} / \text{Общее рабочее время}) * 100\%$

3)  $IPR = \text{Размер продукта} / \text{Общее время инспектирования}$

8. Существует метрика эффективности производственного процесса компании Phase Containment Effectiveness (PCE) (эффективность обнаружения ошибок). Целью компании является

1) уменьшение этой метрики;

- 2) \*увеличение этой метрики;
- 3) неизменность этой метрики.

9. Существует метрика эффективности производственного процесса компании Problem Resolution Rate (PRR) (число отработанных задач за единицу времени). Целью компании является

- 1) уменьшение этой метрики;
- 2) \*увеличение этой метрики;
- 3) неизменность этой метрики.

10. Существует метрика качества программного продукта In Process Faults (IPF) (плотность ошибок в продукте). Целью компании является

- 1) \*уменьшение этой метрики;
- 2) увеличение этой метрики;
- 3) неизменность этой метрики.

11. Существует метрика качества программного продукта Product Fault Density (PFD) (плотность ошибок, внесённых на каком-либо этапе проекта). Целью компании является

- 1) \*уменьшение этой метрики;
- 2) увеличение этой метрики;
- 3) неизменность этой метрики.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Дальневосточный федеральный университет»  
(ДФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по дисциплине «Технологии коллективной промышленной разработки информационных  
систем»

**Направление подготовки – 09.03.04 Программная инженерия**

**Форма подготовки (очная)**

Владивосток  
2019

## **Рекомендации по планированию и организации времени, отведенного на изучение дисциплины**

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступить к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине.

## **Рекомендации по работе с литературой**

1. Всю учебную литературу желательно изучать «под конспект». Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

2. При работе над литературой обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

3. При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте.

## **Требования к представлению и оформлению результатов самостоятельной работы**

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 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.

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

### **Критерии оценки выполнения самостоятельной работы**

Критерии оценки выполнения самостоятельной работы – достижение правильного результата при разработке программного средства, качество оформления документации и представления работы на семинаре.

### **Подготовка к промежуточной аттестации по дисциплине: экзамену (зачету)**

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали



уверенные знания в ходе выполнении лабораторных работ и итогового проекта.

Непосредственная подготовка к аттестации осуществляется по вопросам, представленным в рабочей учебной программе. Тщательно изучите формулировку каждого вопроса, вникните в его суть, составьте план ответа.

### **Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины**

**Текущая аттестация студентов.** Текущая аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме защиты проекта, выполняемого в рамках самостоятельной работы параллельно с лабораторными работами и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме собеседования;
- уровень овладения практическими умениями и навыками – оценивается в форме защиты проекта.

**Промежуточная аттестация студентов.** Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрены зачет и экзамен, который проводится в устной и тестовой форме.

## Критерии выставления оценки студенту на зачете (экзамене)

Баллы (рейтингов ой оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.