



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

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

«СОГЛАСОВАНО»

Руководитель образовательной программы

  
28.08

И.Л. Артемьева

2015 г.

«УТВЕРЖДАЮ»  
Заведующая кафедрой прикладной математики, механики,  
управления и программного обеспечения



И.Л. Артемьева

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Коллективная разработка распределённых систем

**Направление подготовки – 02.03.03 «Математическое обеспечение и администрирование информационных систем»**

профиль «Технология программирования»

**Форма подготовки (очная)**

Курс 4, семестр 7,8

Лекции – 48 час.

практические занятия 0 час.

лабораторные работы 60 час.

в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 42 час.

всего часов аудиторной нагрузки – 108 час.

в том числе с использованием МАО – 42 час.

самостоятельная работа 72 час.

контрольные работы (количество) - нет

курсовая работа / курсовой проект – не предусмотрен

Зачет – 7 семестр

Экзамен – 8 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утвержденного приказом Министерства образования и науки РФ от 15 марта 2015 г. № 222

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 7 от « 4 » июля 2015 г.

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения Артемьева И.Л., д.т.н., профессор

Составитель (ли): доцент кафедры прикладной математики, механики, управления и программного обеспечения Гриняк Виктор Михайлович

**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)

## ABSTRACT

**Bachelor's degree in 02.03.03 – Software and Computer Systems Administration**

**Study profile** “Programming technology”

**Course title:** Team software development

**Basic part of Block 1, 4 credits**

**Instructor:** Victor Grinyak

**At the beginning of the course a student should be able to:** Office programs working, oral presentation.

**Learning outcomes:** This course builds on the skills students gained in Java Fundamentals and helps them advance their Java programming skills using the Java SE7, the latest release of Java. Participants are introduced to the core application programming interfaces used to design objectoriented applications with Java. Hand-on practices and projects figure prominently throughout this course. Those who successfully complete it will possess strong foundational knowledge for Oracle Java SE 7 Programmer I certification.

**Course description:** This course engages students with little or no programming experience to create Java programs. Participants are introduced to objectoriented programming concepts, terminology, and syntax, and the steps required to create basic Java programs using the Alice, Greenfoot, and Eclipse interactive development environments. Hand-on practices figure prominently throughout this course so students can experience firsthand the power of computer programming.

### **Main course literature:**

1. Adams J. Alice 3 in Action: Computing Through Animation. - Cengage Learning. 2014.

2. Kölling M. Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition). - Pearson Education. 2015.

3. Getting Started with Java Using Alice  
[http://ilearningcontent.oracle.com/content/public/oracle\\_acad/SelfStudy/Articulate/Alice/interaction.html](http://ilearningcontent.oracle.com/content/public/oracle_acad/SelfStudy/Articulate/Alice/interaction.html)

4. Creating Java Programs with Greenfoot  
[http://ilearningcontent.oracle.com/content/public/oracle\\_acad/SelfStudy/Articulate/Greenfoot/interaction.html](http://ilearningcontent.oracle.com/content/public/oracle_acad/SelfStudy/Articulate/Greenfoot/interaction.html)

**Form of final control:** exam/pass-fail exam.

## АННОТАЦИЯ

Рабочая программа дисциплины «Коллективная разработка распределённых систем» разработана для студентов 4 курса, обучающихся по направлению 02.03.03 «Математическое обеспечение и администрирование информационных систем, профиль «Технология программирования». Трудоемкость дисциплины 5 зачетных единиц (180 часов). Дисциплина «Коллективная разработка распределённых систем» входит в блок дисциплин по выбору студентов вариативной части профессионального цикла. Дисциплина базируется на дисциплинах «Информатика и программирование», «Компьютерный практикум», «Объектно-ориентированное программирование», «Архитектура вычислительных систем», «Архитектура компьютерных сетей», «Базы данных», «Компьютерная графика», «Теория языков программирования и методы трансляции», «Теория вычислительных процессов и структур», «Технология разработки программного обеспечения», «Языки программирования», «Современные интернет технологии».

Дисциплина реализуется в 7 и 8 семестре (семестрах). В 7 семестре дисциплина содержит 36 часов лекций, 0 часов практических занятий, 36 часов лабораторных работ, из них 0 часов лекций, 0 часов практических занятий, 18 часов лабораторных работ с использованием методов активного обучения. В 8 семестре дисциплина содержит 12 часов лекций, 0 часов практических занятий, 24 часов лабораторных работ, из них 0 часов лекций, 0 часов практических занятий, 24 часов лабораторных работ с использованием методов активного обучения. На самостоятельную работу студентов отводится 72 часа, в том числе на подготовку к экзамену 27 часов.

**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 «Коллективная разработка распределённых систем» у обучающихся должны быть сформированы следующие предварительные компетенции: способность к самоорганизации и самообразованию; способностью к коммуникации в устной и письменных формах на русском и иностранных языках для решения задач межличностного и межкультурного взаимодействия, способностью работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия; способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности, способностью к определению общих форм и закономерностей отдельной предметной области; способностью публично представлять собственные и известные научные результаты, способностью использовать методы математического и алгоритмического моделирования при анализе управленческих задач в научно-технической сфере, в экономике, бизнесе и гуманитарных областях.

В результате изучения данной дисциплины у обучающихся формируются следующие общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК4 способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения	Знает	основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
	Умеет	применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
	Владеет	навыками использования основных методов и средств автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения

ОПК8 Способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения	Знает	основные средства, поддерживающие создание программного обеспечения
	Умеет	использовать знания методов проектирования и производства программного продукта.
	Владеет	работы с инструментальными средствами, поддерживающими создание программного обеспечения
ОПК9 способность использовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки	Знает	методы организации работы в коллективах разработчиков ПО
	Умеет	использовать знания методов организации работы в коллективах разработчиков ПО.
	Владеет	навыками организации работы в коллективах разработчиков ПО
ОПК11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знает	основные приёмы выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения
	Умеет	использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения.
	Владеет	навыками выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях
ПК7 владение знаниями о содержании, основных этапах и тенденциях развития программирования, математического обеспечения и информационных технологий	Знает	основные этапы развития программирования, математического обеспечения и информационных технологий
	Умеет	анализировать и оценивать программы с точки зрения их отнесения к тому или иному этапу программирования, математического обеспечения и информационных технологий
	Владеет	навыками создания программных средств с использованием современных интегрированных сред разработки визуальных приложений

Для формирования вышеуказанных компетенций в рамках дисциплины «Коллективная разработка распределённых систем» применяются следующие методы активного/ интерактивного обучения: метод проектов, дискуссия, презентация.

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

## **Лекционный материал (48 часов)**

### **Тема 1 Введение в коллективную разработку ПО (4 час.)**

Понятие коллективной разработки программного обеспечения. Роли участников команды, ответственные, заместители. Взаимодействие между командами. Требования к составу и оформлению проектной документации.

### **Тема 2 Управление проектами (4 час.)**

Введение в управление проектами. Важность планирования. Написание плана проекта. Проектные роли. Статусные собрания. Взаимодействие с заказчиком. План конфигурационного управления. План управления качеством. План тестирования. Списки задач на проекте. Сроки и графики. Управление рисками, проблемами и зависимостями. Отчётность

### **Тема 3 Управление конфигурацией (4 час.)**

Введение в конфигурационный менеджмент. Управляемое внесение изменений в рабочие продукты. Комитет контроля изменений. Обзор систем управления изменениями. Redmine, eTraxis, Github. Жизненный цикл запроса на изменение или отчёта об ошибке. Правила работы с тикетами, обязательные поля и состояния, правила перехода между состояниями. Обзор систем контроля версий. ClearCase, SVN, Git, Perforce, Mercury.

### **Тема 4 Анализ предметной области и Разработка требований (4 час.)**

Введение в методики анализа предметных областей. Взаимодействие с заказчиком, Разработка сценариев использования. Маркетинговые и функциональные требования, системные требования. Матрица покрытия требований (RTM). Требования к группе требований. Структура требования. Требования к требованиям. Примеры плохих и хороших требований и наборов требований.

### **Тема 5 Архитектура и дизайн (4 час.)**

Введение в архитектуру и дизайн программных продуктов. Декомпозиция системы на подсистемы и компоненты. Взаимодействие между подсистемами и компонентами. Диаграммы последовательностей. Многопоточная обработка данных. Диаграммы развёртывания системы.

### **Тема 6 Разработка. (4 час.)**

Введение в разработку. Кодирование и отладка отдельных модулей системы. Модульное тестирование. Интеграция модулей в компоненты,

интеграция компонентов в подсистемы, интеграция подсистем в систему. Конфигурирование системы. Развёртывание системы в производство.

**Тема 7 Модель зрелости организации CMM (4 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 Тестирование (4 час.)**

Введение в тестирование программных продуктов. Виды тестирования. Роли отдельных участников команды в ходе тестирования. Интеграционное тестирование. Системное тестирование. Приёмочное тестирование.

**Тема 9 Современные технологии и средства построения распределённых систем (4 час.)**

Архитектуры распределённых систем: клиент-сервер, тонкие клиенты, толстые клиенты, архитектура Peer-To-Peer. Использование протоколов TCP и UDP для построения распределённой системы. Передача по сети простых типов данных и сложных объектов с использованием механизма сериализации. Создание распределённой системы с помощью RMI. Использование технологии CORBA. Технология Web сервисов. Использование технологии JMS. Транзакции и репликация данных.

**Тема 10 Теория сетей Петри (4 час.)**

Простые сети Петри. Граф сети Петри. Понятие маркировки и достижимости. Дерево состояний (или дерево достижимости). Пометка сети Петри и бисимуляционная эквивалентность сетей. Точки доступа в сетях Петри. Композиционные операции в сетях Петри. Свойства операций композиции. Примеры построения композиции моделей на исходном наборе моделей. Цветные сети Петри. Композиционные операции цветных сетей Петри. Моделирование вычислений в терминах сетей Петри. Модели классических задач параллельного программирования в сетях Петри (обедающие философы, проблема читателей и писателей).

**Тема 11 Парадигмы параллельного программирования: передача сообщений (4 час.)**

Архитектура многопроцессорной системы с распределенной архитектурой памяти. Модель взаимодействия открытых систем. Понятие протокола передачи данных. Базовые операции передачи



данных: Send-Receive. Организация передачи данных на прикладном уровне в распределенных и параллельных программах. Библиотека сокетов и реализация клиент-серверной модели взаимодействия. Стандарт интерфейса передачи сообщений (MPI). Парная передача сообщений. Групповая передача сообщений. Топологии процессов.

## **Тема 12 Парадигмы параллельного программирования: синхронизация (4 час.)**

Архитектура многопроцессорной системы с общей памятью. Проблема синхронизации потоков. Проблема условий гонок. Объекты синхронизации: семафоры и мьютексы. Многопоточное программирование и управление потоками в прикладных параллельных программах. Технология программирования OpenMP. Распараллеливание циклов. Группы потоков и распределение нагрузки.

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

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

Не предусмотрены учебным планом

### **Лабораторные работы (60 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 (36 час.)**

**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» (6 час.)

**Лабораторная работа №4** Разработка системных требований и верхнего уровня проекта программной системы; подготовка документа «Системные требования» (6 час.)

**Лабораторная работа №5** Разработка архитектуры программной системы; подготовка документа «Дизайн проекта» (6 час)

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

**Лабораторная работа №7** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (6 час)

## **8 семестр (24 час.)**

**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 (4 час.)

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

**Лабораторная работа №5** Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (4 час)

**Лабораторная работа №6** Разработка списка задач на доработку программной системы по результатам тестирования (4 час.)

**Лабораторная работа №7** Кодирование программной системы согласно разработанному списку задач; подготовка отчета по эффективности обработки задач (4 час.)

### **III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Коллективная разработка распределённых 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 -		
			текущий контроль	промежуточная аттестация	
1	Тема 1 Введение в коллективную разработку ПО	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
2	Тема 2. Управление проектами	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
3	Тема 3. Управление конфигурацией продуктов	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
4	Тема 4. Анализ предметной области и Разработка требований	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект	зачёт/экзамен

				(ПР-9)	(ПР-1)
5	Тема 5. Архитектура и дизайн	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
6	Тема 6. Разработка	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
7	Тема 7. Модель зрелости организации СММ	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
8	Тема 8. Тестирование	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
9	Тема 9. Современные технологии и средства построения распределённых систем	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
10	Тема 10. Теория сетей Петри	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
11	Тема 11. Парадигмы параллельного программирования: передача сообщений	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
12	Тема 12. Парадигмы параллельного программирования: синхронизация	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)

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. 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>
7. Кознов Д.В. Введение в программную инженерию - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2015.— 306 с. <http://www.iprbookshop.ru/52146.html?replacement=1>

### **Дополнительная литература**

*(печатные и электронные издания)*

1. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
2. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
3. Канер С., Фолк Д., Нгуен Е. К. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. Пер. с англ. М: ДиаСофт. 2001. 544 с. <http://www.twirpx.com/file/74880/>
4. Майерс Г. Искусство тестирования программ. 3-е изд. Пер. с англ. М: Диалектика, Вильямс. 2012. 272 с.
5. 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. Martin Fowler <http://www.martinfowler.com/>
2. Каталог рефакторинга <https://refactoring.guru/ru/catalog>
3. Catalog of Patterns of Enterprise Application Architecture  
<http://www.martinfowler.com/eaCatalog/index.html>
4. Game Programming Patterns  
<http://gameprogrammingpatterns.com/contents.html>

### **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**

Занятия проводятся с использованием проектора и мультимедийного комплекса для демонстрации мультимедийного контента внутренней системы портала ДВФУ. Лабораторные занятия проводятся в компьютерном классе.



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

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ  
ОБУЧАЮЩИХСЯ**

по дисциплине «Коллективная разработка распределённых систем»

**Направление подготовки – 02.03.03 «Математическое обеспечение и  
администрирование информационных систем»**

профиль «Технология программирования»

**Форма подготовки (очная)**



## План-график выполнения самостоятельной работы по дисциплине

### 1 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

### 2 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 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 - докумен, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов.
- Requirements Tracaebility Matrix - матрица покрытия тестами требований.
- Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов.

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

### **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.

## Рекомендации по самостоятельной работе студентов

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 коллегами. 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 Traceability 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 ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Дальневосточный федеральный университет»  
(ДВФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
по дисциплине «Коллективная разработка распределённых систем»  
**Направление подготовки – 02.03.03 «Математическое обеспечение и  
администрирование информационных систем»**  
профиль «Технология программирования»  
**Форма подготовки (очная)**



## Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК4 способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения	Знает	основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
	Умеет	применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
	Владеет	навыками использования основных методов и средств автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения
ОПК8 Способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения	Знает	основные средства, поддерживающие создание программного обеспечения
	Умеет	использовать знания методов проектирования и производства программного продукта.
	Владеет	работы с инструментальными средствами, поддерживающими создание программного обеспечения
ОПК9 способность использовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки	Знает	методы организации работы в коллективах разработчиков ПО
	Умеет	использовать знания методов организации работы в коллективах разработчиков ПО.
	Владеет	навыками организации работы в коллективах разработчиков ПО
ОПК11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знает	основные приёмы выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения
	Умеет	использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения.
	Владеет	навыками выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях
ПК7 владение знаниями о содержании, основных этапах и тенденциях	Знает	основные этапы развития программирования, математического обеспечения и информационных технологий

развития программирования, математического обеспечения и информационных технологий	Умеет	анализировать и оценивать программы с точки зрения их отнесения к тому или иному этапу программирования, математического обеспечения и информационных технологий
	Владеет	навыками создания программных средств с использованием современных интегрированных сред разработки визуальных приложений

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Тема 1 Введение в коллективную разработку ПО	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
2	Тема 2. Управление проектами	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
3	Тема 3. Управление конфигурацией продуктов	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
4	Тема 4. Анализ предметной области и Разработка требований	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
5	Тема 5. Архитектура и дизайн	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
6	Тема 6. Разработка	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект	зачёт/экзамен

				(ПР-9)	(ПР-1)
7	Тема 7. Модель зрелости организации СММ	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
8	Тема 8. Тестирование	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
9	Тема 9. Современные технологии и средства построения распределённых систем	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
10	Тема 10. Теория сетей Петри	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
11	Тема 11. Парадигмы параллельного программирования: передача сообщений	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)
12	Тема 12. Парадигмы параллельного программирования: синхронизация	ОПК4, ОПК8, ОПК9, ОПК11, ПК7	знания	опрос (УО-1)	зачёт/экзамен (ПР-1)
			умения	задание (ПР-6)	зачёт/экзамен (ПР-1)
			владения	проект (ПР-9)	зачёт/экзамен (ПР-1)

### Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОПК4 способность применять в профессиональной деятельности	знает (пороговый уровень)	основные методы и средства автоматизации проектирования, производства,	Знание основы жизненного цикла разработки визуальных приложений	Способность сформулировать не менее 4 этапов жизненного цикла

основные методы и средства автоматизации и проектирования, производства, испытаний и оценки качества программного обеспечения		испытаний и оценки качества программного обеспечения		
	умеет (продвинутый)	применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения	Умение разрабатывать архитектуру программных средств и реализовывать программные средства	Способность описать архитектуру программы, состоящую из не менее чем 5 программных модулей
	владеет (высокий)	навыками использования основных методов и средств автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения	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
ОПК8 Способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание	знает (пороговый уровень)	основные средства, поддерживающие создание программного обеспечения	Знание средств, поддерживающих создание программного обеспечения	Способность работать с не менее чем 4 средствами
	умеет (продвинутый)	использовать знания методов проектирования и производства программного продукта.	Умение разрабатывать архитектуру программных средств и реализовывать программные средства	Способность описать архитектуру программы, состоящую из не менее чем 10 программных модулей
	владеет (высокий)	работы с инструментальными средствами, поддерживающими создание программного обеспечения	Умение разрабатывать программные средства	Способность разрабатывать программные средства объемом не менее 1500 строк.

программного обеспечения				
ОПК9 способность использовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки	знает (пороговый уровень)	методы организации работы в коллективах разработчиков ПО	Знание методов организации работы в коллективах разработчиков ПО	Способность организовать работу коллектива разработчиков из не менее чем 4 человек
	умеет (продвинутой)	использовать знания методов организации работы в коллективах разработчиков ПО.	Умение использовать знания методов организации работы в коллективах разработчиков ПО.	Способность организовать работу коллектива разработчиков из не менее чем 4 человек
	владеет (высокий)	навыками организации работы в коллективах разработчиков ПО	Умение организовать работу в коллективах разработчиков ПО	Способность организовать работу коллектива разработчиков из не менее чем 4 человек
ОПК11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности и программного обеспечения для решения задач в различных предметных областях	знает (пороговый уровень)	основные приёмы выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения	Знание основных приёмов выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения	Способность сформулировать не менее 5 этапов инспекций рабочих продуктов при производстве ПО
	умеет (продвинутой)	использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения.	Умение использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения.	Способность разрабатывать программу измерений эффективности процесса разработки ПО не менее чем из 5 метрик
	владеет (высокий)	навыками выбора, проектирования, реализации, оценки качества и анализа эффективности программного	Умение оценивать качество программных средства	Способность разрабатывать программу измерений качества ПО не менее чем из 5 метрик

		обеспечения для решения задач в различных предметных областях		
ПК7 владение знаниями о содержании, основных этапах и тенденциях развития программирования, математического обеспечения и информационных технологий	знает (пороговый уровень)	основные этапы развития программирования, математического обеспечения и информационных технологий	Знание основных этапов	Не менее 4 этапов
	умеет (продвинутый)	анализировать и оценивать программы с точки зрения их отнесения к тому или иному этапу программирования, математического обеспечения и информационных технологий	Умение относить программу к тому или иному этапу	Не менее 4 этапов
	владеет (высокий)	навыками создания программных средств с использованием современных интегрированных сред разработки визуальных приложений	Умение разрабатывать и презентовать программные средства	Способность разрабатывать программные средства объемом не менее 500 строк.

**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.

### **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**

- Классификация Флинна. Основные законы параллельных вычислений.
- Векторные расширения современных процессоров: принципы и практическое использование.
- Параллельная обработка информации: языки, методы и средства.
- Многопоточное программирование: понятие потока, проблемы синхронизации и условий гонок.
- Взаимодействие потоков на общей памяти, основные объекты синхронизации.
- Основы работы с технологией OpenMP: директивы препроцессора, переменные среды.
- Взаимодействие параллельных процессов через передачу сообщений: принципы, базовые операции.
- Основы MPI: парная передача сообщений, синхронный и асинхронный режимы передачи сообщений.
- Основы MPI: групповая передача сообщений.
- Основы MPI: топологии процессов.
- Основы технологии CUDA.
- Муравьиные алгоритмы.
- Амортизационный анализ.
- Схема программы и ее интерпретация. Свободные интерпретации. Логико-термальная и функциональная эквивалентности.
- Основные проблемы схематологии (тотальность, пустота, свобода). Разрешимость и неразрешимость проблем.



- Преобразования схема программ.
- Просты сети Петри. Маркировка. Дерево достижимости.
- Пометка сети Петри и бисимуляционная эквивалентность сетей.
- Точки доступа к сетям Петри. Базовые операции композиции сетей Петри.
- Модель последовательной программы в терминах сетей Петри.
- Модель параллельной программы в терминах сетей Петри.
- Цветные сети Петри. Композиция цветных сетей Петри.
- Описание процессов в терминах CCS.
- Описание процессов в терминах CSP.
- Основные операции композиции процессов в терминах CSP.
- Темпоральная логика и ветвление времени в программах.

### Вопросы к экзамену

Экзамен проводится в форме тестирования

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)

1. Одним из основных подходов к организации процесса создания и использования программных средств является конвейерный (или водопадный) подход. Он характеризуется тем, что
  - 1) предполагается быстрая (насколько это возможно) реализация рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции;
  - 2) \*разработка ПС состоит из цепочки этапов, на каждом этапе создаются рабочие продукты, используемые на последующем этапе;
  - 3) создаются рабочие версии программ, предназначенные для проведения экспериментов с целью установить или уточнить требования к ПС.
  
2. Одним из основных подходов к организации процесса создания и использования программных средств является исследовательское программирование. Оно характеризуется тем, что
  - 1) \*предполагается быстрая (насколько это возможно) реализация рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции, а после производится их модификация с целью сделать их более полезными для пользователей;
  - 2) разработка ПС состоит из цепочки этапов, на каждом этапе создаются рабочие продукты, используемые на последующем этапе;

3) создаются рабочие версии программ, предназначенные для проведения экспериментов с целью установить или уточнить требования к ПС.

3. Одним из основных подходов к организации процесса создания и использования программных средств является прототипирование. Оно характеризуется тем, что

1) предполагается быстрая (насколько это возможно) реализация рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции;

2) разработка ПС состоит из цепочки этапов, на каждом этапе создаются рабочие продукты, используемые на последующем этапе;

3) \*создаются рабочие версии программ, предназначенные для проведения экспериментов с целью установить или уточнить требования к ПС.

4. Одним из основных подходов к организации процесса создания и использования программных средств являются формальные преобразования. Они характеризуются тем, что

1) предполагается быстрая (насколько это возможно) реализация рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции;

2) \*разрабатываются формальные спецификации ПС, которые затем превращаются в программы путем корректных преобразований;

3) создаются рабочие версии программ, предназначенные для проведения экспериментов с целью установить или уточнить требования к ПС.

5. Одним из основных подходов к организации процесса создания и использования программных средств является сборочное программирование. Оно характеризуется тем, что

1) предполагается быстрая (насколько это возможно) реализация рабочих версий программ ПС, выполняющих лишь в первом приближении требуемые функции;

2) разрабатываются формальные спецификации ПС, которые затем превращаются в программы путем корректных преобразований;

3) \*ПС конструируется, главным образом, из компонент, которые уже существуют.

6. Одним из критериев качества программных средств является его функциональность. При этом под функциональностью программного средства понимается

- 1) \* способность ПС выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей;
- 2) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) набор характеристик ПС, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПС и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

7. Одним из критериев качества программных средств является его надёжность. При этом под надёжностью программного средства понимается

- 1) способность ПС выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей;
- 2) \*способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) набор характеристик ПС, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПС и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

8. Одним из критериев качества программных средств является его лёгкость применения. При этом под лёгкостью применения программного средства понимается

- 1) способность ПС выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей;
- 2) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) \*набор характеристик ПС, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПС и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

9. Одним из критериев качества программных средств является его эффективность. При этом под эффективностью программного средства понимается

- 1) \*отношение уровня услуг, предоставляемых ПС пользователю при заданных условиях, к объему используемых ресурсов;
- 2) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) набор характеристик ПС, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению ПС и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

10. Одним из критериев качества программных средств является его сопровождаемость. При этом под сопровождаемостью программного средства понимается

- 1) отношение уровня услуг, предоставляемых ПС пользователю при заданных условиях, к объему используемых ресурсов;
- 2) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) \*набор характеристик ПС, которые позволяют минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей

11. Одним из критериев качества программных средств является его мобильность. При этом под мобильностью программного средства понимается

- 1) \*способность ПС быть перенесенным из одной среды (окружения) в другую, в частности, с одного компьютера на другой;
- 2) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.
- 3) набор характеристик ПС, которые позволяют минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей

12. Обязательными критериями качества программных средств являются

- 1) \* функциональность и надёжность;
- 2) сопровождаемость и мобильность;
- 3) эффективность и лёгкость применения.

13. Одним из подходов к обеспечению надёжности программных средств является предупреждение ошибок. Под ним понимается

- 1) \*реализация комплекса мер, цель которых – минимизировать число ошибок в готовых программных продуктах;
- 2) реализация средств обнаружения отказа в программе в процессе её выполнения;
- 3) реализация средств обнаружения отказа в программе в процессе её выполнения и исправление последствий этого отказа.

14. Одним из подходов к обеспечению надёжности программных средств является самообнаружение ошибок. Под ним понимается

- 1) реализация комплекса мер, цель которых – минимизировать число ошибок в готовых программных продуктах;
- 2) \*реализация средств обнаружения отказа в программе в процессе её выполнения;
- 3) реализация средств обнаружения отказа в программе в процессе её выполнения и исправление последствий этого отказа.

15. Одним из подходов к обеспечению надёжности программных средств является самоисправление ошибок. Под ним понимается

- 1) реализация комплекса мер, цель которых – минимизировать число ошибок в готовых программных продуктах;
- 2) реализация средств обнаружения отказа в программе в процессе её выполнения;
- 3) \*реализация средств обнаружения отказа в программе в процессе её выполнения и исправление последствий этого отказа.

16. Элементом предупреждения ошибок в программных средствах является смежный контроль рабочих продуктов. Под ним понимается

- 1) контроль не только рабочих продуктов как таковых, но и проверка, какой процесс обработки данных они реализуют;
- 2) независимое обеспечение проверки точности перевода требований;
- 3) \*проверка рабочего продукта лицами, не участвующими в его разработке, с двух сторон: во-первых, со стороны автора исходного для контролируемого продукта документа, и, во-вторых, лицами, которые будут использовать полученный рабочий продукт в качестве исходного в последующих технологических процессах.

## 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.

### Вопросы для собеседования

1. Классификация Флинна. Основные законы параллельных вычислений.
2. Векторные расширения современных процессоров: принципы и практическое использование.
3. Параллельная обработка информации: языки, методы и средства.
4. Многопоточное программирование: понятие потока, проблемы синхронизации и условий гонок.
5. Взаимодействие потоков на общей памяти, основные объекты синхронизации.

6. Взаимодействие параллельных процессов через передачу сообщений: принципы, базовые операции.
7. Схема программы и ее интерпретация. Свободные интерпретации. Логико-термальная и функциональная эквивалентности.
8. Основные проблемы схематологии (тотальность, пустота, свобода). Разрешимость и неразрешимость проблем.
9. конвейерный (или водопадный) подход.
- 10.исследовательское программирование.
- 11.прототипирование.
- 12.формальные преобразования.
- 13.сборочное программирование.
- 14.функциональность программных средств.
- 15.надёжность программных средств.
- 16.лёгкость применения программных средств.
- 17.эффективность программных средств.
- 18.сопровождаемость программных средств.
- 19.мобильность программных средств.
- 20.критерии качества программных средств
- 21.смежный контроль рабочих продуктов.