



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

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»

Руководитель ОП

 Артемьева И.Л.

«УТВЕРЖДАЮ»

И.о. директора департамента

 Смагин С.В.
«23» марта 2022 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Коллективная разработка распределенных систем

**Направление подготовки 02.03.03 Математическое обеспечение и администрирование
информационных систем
(Технология программирования)
Форма подготовки очная**

курс 4 семестр 7, 8
лекции 30 час.
практические занятия 00 час.
лабораторные работы 30 час.
в том числе с использованием МАО лек. 9 /пр. 0 /лаб. 29 час.
всего часов аудиторной нагрузки 60 час.
в том числе с использованием МАО 29 час.
самостоятельная работа 84 час.
в том числе на подготовку к экзамену 36 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект 8 семестр
зачет 7 семестр
экзамен 8 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем, утвержденного приказом Министерства образования и науки РФ от 23 августа 2017 г. № 809 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, протокол № 3.0 от «23» марта 2022 г.

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.
Составитель: В.М. Гриняк, д.т.н., доцент

Владивосток

2022

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

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

Задачи дисциплины:

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 распределенных систем у обучающихся должны быть сформированы следующие предварительные компетенции: таких нет.

Универсальные компетенции выпускников и индикаторы их достижения:

Наименование категории (группы) универсальных компетенций	Код и наименование универсальной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
---	---	--

Системное и критическое мышление	УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1 Определяет свою роль в социальном взаимодействии и командной работе, исходя из стратегии сотрудничества для достижения поставленной цели УК-3.2 Осуществляет обмен информацией, знаниями и опытом с членами команды УК-3.3 Соблюдает нормы и установленные правила командной работы; несет личную ответственность за результат
----------------------------------	--	---

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
УК-3.1 Определяет свою роль в социальном взаимодействии и командной работе, исходя из стратегии сотрудничества для достижения поставленной цели	Знает принципы формирования команд разработчиков программного обеспечения
	Умеет исполнять основные роли участников команд разработчиков программного обеспечения.
	Владеет навыками исполнения ролей участников команд разработчиков программного обеспечения
УК-3.2 Осуществляет обмен информацией, знаниями и опытом с членами команды	Знает принципы взаимодействия участников команд разработчиков программного обеспечения.
	Умеет использовать инструменты взаимодействия участников команд разработчиков программного обеспечения.
	Владеет навыками работы с инструментами взаимодействия участников команд разработчиков программного обеспечения.
УК-3.3 Соблюдает нормы и установленные правила командной работы; несет личную ответственность за результат	Знает принципы взаимодействия участников команд разработчиков программного обеспечения.
	Умеет использовать инструменты взаимодействия участников команд разработчиков программного обеспечения.
	Владеет навыками исполнения ролей участников команд разработчиков программного обеспечения

Профессиональные компетенции выпускников и индикаторы их достижения:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
	ПК-1. - Способность	ПК-1.1. демонстрирует знания,

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
научно-исследовательский	демонстрировать базовые знания математических и естественных наук,	полученные в области математических и (или) естественных наук, программирования и информационных технологий
	программирования и информационных технологий	ПК-1.2. формулирует стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий.
		ПК-1.3. решает стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий.
производственно-технологический	ПК-6. Способность использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов.	ПК-6.1. демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования
		ПК-6.2. использует современные инструментальные средства в практической деятельности
	ПК-9. Способность использования современных методов разработки и реализации конкретных алгоритмов	ПК-9.1. демонстрирует знание современных методов разработки и реализации алгоритмов математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
	математических моделей на базе языков программирования и пакетов прикладных программ моделирования	ПК-9.2. использует современные методы разработки алгоритмов, математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ
		ПК-9.3. разрабатывает и реализует алгоритмы на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-1.1. демонстрирует знания, полученные в области математических и (или) естественных наук, программирования и информационных технологий	Знает современные технологии проектирования и производства программного обеспечения
	Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
ПК-1.2. формулирует стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий.	Знает современные технологии проектирования и производства программного обеспечения
	Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
ПК-1.3. решает стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий.	Знает современные технологии реализации, оценки качества и анализа эффективности программного обеспечения
	Умеет правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения
	Владеет навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
ПК-6.1. демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программ в рамках

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
	функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-6.2. использует современные инструментальные средства в практической деятельности	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программ в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-6.3. применяет современные инструментальные средства при разработке программных продуктов	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования
	Умеет создавать программы в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
	Владеет навыками создания программных систем в рамках функционального, логического, объектно-ориентированного и визуального направлений программирования
ПК-9.1. демонстрирует знание современных методов разработки и реализации алгоритмов математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ	Знает методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.
	Умеет использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.
	Владеет навыками коллективной разработки и использования ПО
ПК-9.2. использует современные методы разработки алгоритмов, математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ.	Знает методы организации работы в профессиональной деятельности.
	Умеет использовать методы организации работы в профессиональной деятельности.
	Владеет навыками организации работы в профессиональной деятельности.
ПК-9.3. разрабатывает и реализует алгоритмы на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ	Знает методы разработки и сопровождения ПО при создании программных средств
	Умеет применять методы разработки и сопровождения ПО при создании программных средств
	Владеет навыками создания программных средств и их сопровождения

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

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекционные занятия
Лаб	Лабораторные работы
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1	Раздел I. Структура и анализ бизнес-процессов разработки программного обеспечения.	7	16	16	-	-	48	36	УО-1, ПР-6, ПР-2, ПР-5
2	Раздел II. Тестирование и отладка программного средства	8	14	14					
Итого:			30	30		-	48	36	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционный материал (30 часов)

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

Тема 1 Введение в коллективную разработку ПО (2 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 (2 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 (2 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 (3 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 (3 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. (3 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.

Раздел 2. Тестирование и отладка программного средства

Тема 7 Модель зрелости организации СММ (2 час.)

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

Тема 8 Тестирование (2 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 (2 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 (3 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. Композициональные операции в сетях Петри. Свойства операций композиции. Примеры построения композиции моделей на исходном наборе моделей. Цветные сети Петри. Композициональные 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 (3 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 (3 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 И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Лабораторные работы (30 час)

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 (2 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» (2 час)

Лабораторная работа №6 Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта» (2 час)

Лабораторная работа №7 Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (4 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 (2 час.)

Лабораторная работа №4 Кодирование программной системы в соответствии с требованиями; подготовка документа «Список задач проекта» (2 час)

Лабораторная работа №5 Тестирование программной системы в соответствии с требованиями; подготовка документа «План тестирования проекта» и «Матрица покрытия требований» (2 час)

Лабораторная работа №6 Разработка списка задач на доработку программной системы по результатам тестирования (2 час.)

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. Разработка студенческих проектов ведётся на базе именно этой системы контроля версий с использованием встроенной системы issue трекинга.

В начале учебного семестра студентам необходимо разбиться на группы по 3-5 человек, в каждой команде должен быть назначен руководитель (team-

leader), кодировщики и технический писатель. Каждая команда должна определиться с программным проектом, который она будет разрабатывать в течение семестра. Если команда не может выбрать себе проект, то тематику проектов предлагает преподаватель.

Задания для самостоятельной работы

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

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

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

Индивидуальные задания на самостоятельную работу студенты получают в ходе процесса разработки программного продукта в соответствии с теми ролями, которые они играют, с этапом разработки проекта, в зависимости от потребности бригады и в соответствии с планом работы бригады. Распределение заданий производит бригадир под контролем преподавателя. Примерные виды индивидуальных заданий: подготовка спецификаций подсистемы, разработка тестовых ситуаций, составление набора тестов для одной тестовой ситуации, программирование одной или нескольких функций подсистемы, составление и редактирование документов разработки, инспекция документов разработки, разработка схемы базы данных продукта, проектирование объекта, доклад об инструментальном средстве и т.п.

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

Особенностью учебного процесса по дисциплине является его максимальное приближение к реальным условиям работы на проектах по коллективной разработке ПО: как коммерческим, так и фрилансовым. Всем студентам необходимо создать аккаунт на портале github.com. 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.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

7 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-2 неделя	Создание и настройка репозитория в системе GitHub	5	УО-1, ПР-1
2	3-4 неделя	Разработка презентации проекта Создайте маркетинговые требования к проекту (Marketing Requirements)	5	УО-1, ПР-1

		Specification - MRS) в виде презентации рекламного характера. Разместите её в папке REQUIREMENS Вашего репозитория.		
3	5-6 неделя	Разработка плана проекта Создайте документ "Software Project Plan" согласно образца и разместите его в своём репозитории	5	УО-1, ПР-1
4	7-8 неделя	Разработка требований Создайте и разместите в репозитории документ "Software Requirements Specification". В документе опишите первую версию требований к Вашему продукту.	5	УО-1, ПР-1
5	9-10 неделя	Разработка дизайна проекта Разработайте и разместите в репозитории документ Software Disign Specification. Опишите в нем первую версию дизайна (архитектуры) Вашего проекта.	5	УО-1, ПР-1
6	11-12 неделя	Кодирование проекта Создайте и разместите в репозитории код Вашего проекта. Создайте документ Code Issue List - список задач, выполняемых на этапе кодирования	5	УО-1, ПР-1
7	13-14 неделя	Тестирование проекта Создайте и разместите в репозитории Software Test Plan - документ, включающий в себя спецификации тестов и процедур тестирования, в том числе список тестов Создайте и разместите в репозитории документ Requirements Tracaebility Matrix - матрица покрытия тестами требований Создайте и разместите в репозитории Test Logs - документ, который заполняется в ходе тестирования и содержит результаты выполнения тестов	5	УО-1, ПР-1
8	15-18 неделя	Презентация проекта	5	Защита проекта
	Итого:		40	

6 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-2 неделя	Подготовка к семинару «Профессиональный слэнг в области программной инженерии» Ориентируясь на лекцию 12, вспомните и опишите десяток профессиональных жаргонизмов, используемых программистской средой.	1	УО-1, ПР-1
2	3-4 неделя	Подготовка к семинарам «Введение в технологию программирования Agile и разбор Agile текстов», «Agile критика традиционных технологий программирования»	1	УО-1, ПР-1
3	5-6 неделя	Подготовка к семинарам Принципы Agile, Роли Agile.	1	УО-1, ПР-1
4	7-8 неделя	Подготовка к семинарам Артефакты Agile, Методы Agile	1	УО-1, ПР-1
5	9-10 неделя	Подготовка к семинарам Работа с Agile командами. Оценка Agile подхода	1	УО-1, ПР-1
6	11-12 неделя	Подготовка к семинарам Запахи кода. «Раздувальщики», Запахи кода. Нарушители объектного дизайна	1	УО-1, ПР-1
7	13-14 неделя	Подготовка к семинарам Приёмы методов рефакторинга. Перемещение функций между объектами	1	УО-1, ПР-1
8	15-18 неделя	Работа над итоговым проектом	1	Защита проекта
		Подготовка к экзамену	36	экзамен
	Итого:		44	

Требования к представлению и оформлению результатов самостоятельной работы

Структура рабочего продукта по проекту

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

В папке `CODE` размещается код проекта.

В папке `DOCS` - файлы с документацией.

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

`PROJECT PLAN` - для размещения документов с планом проекта

`REQUIREMENTS` - для размещения документов с требованиями

`DESIGN` - для размещения документов дизайна

`CODING` - для размещения документов кодирования

`TESTING` - для размещения документов тестирования

Работа с системой контроля версий

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

Проект должен иметь двухуровневую структуру веток.

Основная ветка (`master`) - содержит стабильную версию, интеграцию в которую осуществляет только (!!) тим-лидер. Все члены команды, в том числе тим-лидер, создают в своём репозитории копии стабильной версии по мере необходимости. С этих рабочих (`work`) веток они создают запросы на интеграцию (`merge`) в основную ветку. Запрос на интеграцию назначается вначале на одного из членов команды, который выполняет роль инспектора. Если инспектор одобряет работу, он переназначает запрос на тим-лидера, который после проверки осуществляет интеграцию изменений. Сам тим-лидер поступает аналогично: после одобрения инспектором запрос на интеграцию возвращается к тим-лидеру обратно.

Документация, создаваемая по проекту

В рамках работы над проектом создается следующая документация:

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

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

`Software Requirements Specification (SRS)` - технические требования к продукту

`User Manual` - руководство оператора (пользователя).

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

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

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

Requirements Tracaability 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.

Критерии оценки выполнения самостоятельной работы

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано навыки подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

60-50 баллов - если структура и содержание задания не соответствуют требуемым

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

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел 1. Структура и анализ бизнес-процессов разработки программного обеспечения.	ПК-1.1. демонстрирует знание концептуальных моделей менеджмента	Знает современные технологии проектирования и производства программного обеспечения Умеет правильно выбрать технологию производства программного обеспечения для конкретного проекта	ПР-1 - тест; ПР-6 - лабораторные работы; и задания для самостоятельного выполнения; УО-1 – устный опрос; ПР-5 – курсовая работа	Зачет, вопросы № 1-26
			Владеет навыками применения технологий производства программного обеспечения		
		ПК-1.2. использует основные модели менеджмента в управлении.	Знает современные технологии проектирования и производства программного обеспечения Умеет правильно выбрать технологию производства программного		

	обеспечения для конкретного проекта
	Владеет навыками применения технологий производства программного обеспечения
	Знает современные технологии реализации, оценки качества и анализа эффективности программного обеспечения
ПК-1.3. применяет модели и методы менеджмента в управлении ПО	Умеет правильно выбрать технологию реализации, оценки качества и анализа эффективности программного обеспечения
	Владеет навыками применения технологий реализации, оценки качества и анализа эффективности программного обеспечения
	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений

	программирован ия
ПК-6.1. демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования	Умеет создавать программы в рамках функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
	Владеет навыками создания программ в рамках функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
	Знает основные концептуальные положения функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
ПК-6.2. использует современные инструментальные средства в практической деятельности.	Умеет создавать программы в рамках функциональног о, логического, объектно- ориентированно го и визуального

	направлений программирован ия
	Владеет навыками создания программ в рамках функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
	Знает основные концептуальные положения функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
ПК-6.3. применяет современные инструментальные средства при разработке программных продуктов	Умеет создавать программы в рамках функциональног о, логического, объектно- ориентированно го и визуального направлений программирован ия
	Владеет навыками создания программных систем в рамках функциональног о, логического, объектно- ориентированно го и визуального

			направлений программирования		
			Знает современные технологии проектирования и производства программного обеспечения		
2	Раздел 2. Тестирование и отладка программного средства	ПК-9.1. демонстрирует знание современных методов разработки и реализации алгоритмов математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ	Знает методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.	ПР-1 - тест; ПР-6 - лабораторные работы; и задания для самостоятельного выполнения; УО-1 – устный опрос; ПР-5 – курсовая работа	Экзамен, вопросы № 1-16
		Умеет использовать методы организации работы в коллективах разработчиков ПО и методы сопровождения ПО.			
		Владеет навыками коллективной разработки и использования ПО			
		ПК-9.2. использует современные методы разработки алгоритмов, математических моделей на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ	Знает методы организации работы в профессиональной деятельности.		
			Умеет использовать методы организации работы в профессиональной деятельности.		

	Владеет навыками организации работы в профессиональной деятельности.
ПК-9.3. разрабатывает и реализует алгоритмы на базе языков и современных инструментальных систем, в том числе пакетов прикладных программ	Знает методы разработки и сопровождения ПО при создании программных средств
	Умеет применять методы разработки и сопровождения ПО при создании программных средств
	Владеет навыками создания программных средств и их сопровождения

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также качественные критерии оценивания, которые описывают уровень сформированности компетенций, представлены в разделе VIII.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

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. Математическое обеспечение и администрирование информационных систем: 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. 1. 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. 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>
4. Майерс Г. Искусство тестирования программ. 3-е изд. Пер. с англ. М.: Диалектика, Вильямс. 2012. 272 с.

5. Липаев, В.В. Математическое обеспечение и администрирование информационных систем. 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

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

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

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

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

Лабораторные занятия проводятся в специализированном компьютерном классе.

Профессиональные базы данных и информационные справочные системы

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

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

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

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

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

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

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

Лабораторные работы по дисциплине проводятся параллельно с самостоятельной работой студентов. Студенты представляют результаты лабораторной работы преподавателю и друг другу, при этом стараются максимально обменяться опытом.

Рекомендуется использовать различные возможности работы с

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

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

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

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

Материально-техническое и программное обеспечение дисциплины

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
<p>690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус D, ауд. D 733,733а. Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</p>	<p>Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA – 1 шт. Доска аудиторная, Моноблок Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013(13 шт.) и аудиовизуальным и средствами проектор Panasonic DLPPjectorPT-</p>	<p>1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12,Alice 3, Anaconda3,Autodesk,CodeBlocks,CorelDRAW X7,Dia,Directum4.8,DosBox-0.74,Farmanager,Firebird 2.5,FlameRobin,Foxit Reader,Free Pascal,Geany,Ghostsript,Git,Greenfoot,gsview,Inscape0.91,Java,Java development Kit,Kaspersky,Lazarus,LibreOffice4.4,MatLab R2017b,Maxima 5.37.2,Microsoft Expression,Microsoft Office 2013,Microsoft Silverlight,Microsoft Silverlight 5SDK-русский,MicrosoftSistem Center,Microsoft Visial Studio 2012,MikTeX2.9,MySQL,NetBeans,Notepad++,Oracle VM VirtualBox,PascalABC.NET,PostgreSQL 9.4,PTC Mathcad,Putty,PyQt GPL v5.4.1 for Pythonv 3.4,Python2.7(3.4,3.6),QGIS Brighton,RStudio,SAM CoDeC Pack,SharePoint,Strawberry Perl,Tecnomatix,TeXnicCenter,TortoiseSVN,Unity2017.3.1f1,Veusz,Vim8.1,Visual Paradigm CE,Visual Studio2013,Windows Kits,Windows Phone SDK8.1,Xilinx Design ToolsAcrobat ReaderDC,AdobeBridge CS3,AdobeDeviceCentralCS3,Adobe ExtendScript Toolkit 2,Adobe Photoshpe CS3,DVD-студия Windows,GoogleChrome,Internet Explorer,ITMOproctor,Mozilla Firefox, Visual Studio Installer,Windows Media Center, WinSCP,</p>

Для освоения дисциплины требуется наличие проектора, аудиторная доска, компьютер.

Для проведения учебных занятий по дисциплине, а также для организации самостоятельной работы студентам доступно оборудование и специализированные кабинеты, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно-производственных работ.

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

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Для дисциплины «Коллективная промышленная разработка распределенных систем» используются следующие оценочные средства:

Письменные работы:

1. Тест (ПР-1)
2. Контрольная работа (ПР-2)
3. Лабораторная работа (ПР-6)
4. Курсовая работа (ПР-5)

Письменные работы

Письменный ответ приучает к точности, лаконичности, связности изложения мысли. Письменная проверка используется во всех видах контроля и осуществляется как в аудиторной, так и во внеаудиторной работе.

Тест (ПР-1) - система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.

Контрольная работа (ПР-2) - средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу.

Курсовая работа (ПР-5) - средство проверки владения компетенциями;

Лабораторная работа (ПР-6) – средство комплексной проверки знаний, умений и навыков.

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Оценочные средства для промежуточной аттестации

Промежуточная аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной. Форма отчётности по дисциплине – экзамен. Экзамен проводится в тестовой форме.

Методические указания по сдаче экзамена

Экзамен принимается ведущим преподавателем. При большом количестве групп у одного преподавателя или при большой численности потока по распоряжению заведующего кафедрой (заместителя директора по учебной и воспитательной работе) допускается привлечение в помощь ведущему преподавателю других преподавателей. В первую очередь привлекаются преподаватели, которые проводили лабораторные занятия по дисциплине в группах.

В исключительных случаях, по согласованию с заместителем директора Школы по учебной и воспитательной работе, заведующий кафедрой имеет право принять экзамен в отсутствие ведущего преподавателя.

Форма проведения экзамена утверждается на заседании кафедры по согласованию с руководителем в соответствии с рабочей программой дисциплины.

Во время проведения зачета студенты могут пользоваться рабочей программой дисциплины, а также с разрешения преподавателя, справочной литературой и другими пособиями (учебниками, учебными пособиями, рекомендованной литературой и т.п.).

Время, предоставляемое студенту на прохождение теста на экзамене, должно составлять не более 60 минут.

Присутствие на зачете посторонних лиц (кроме лиц, осуществляющих проверку) без разрешения соответствующих лиц (ректора либо проректора по учебной и воспитательной работе, директора Школы, руководителя ОПОП или заведующего кафедрой), не допускается. Инвалиды и лица с ограниченными возможностями здоровья, не имеющие возможности самостоятельного передвижения, допускаются зачет с сопровождающими.

Вопросы к зачету (7 семестр)

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. Основы работы с технологией OpenMP: директивы препроцессора, переменные среды.
7. Взаимодействие параллельных процессов через передачу сообщений: принципы, базовые операции.
8. Основы MPI: парная передача сообщений, синхронный и асинхронный режимы передачи сообщений.
9. Основы MPI: групповая передача сообщений.
10. Основы MPI: топологии процессов.
11. Основы технологии CUDA.
12. Муравьиные алгоритмы.
13. Амортизационный анализ.
14. Схема программы и ее интерпретация. Свободные интерпретации. Логико-термальная и функциональная эквивалентности.
15. Основные проблемы схематологии (тотальность, пустота, свобода). Разрешимость и неразрешимость проблем.
16. Преобразования схема программ.
17. Просты сети Петри. Маркировка. Дерево достижимости.
18. Пометка сети Петри и бисимуляционная эквивалентность сетей.
19. Точки доступа к сетям Петри. Базовые операции композиции сетей Петри.
20. Модель последовательной программы в терминах сетей Петри.
21. Модель параллельной программы в терминах сетей Петри.
22. Цветные сети Петри. Композиция цветных сетей Петри.
23. Описание процессов в терминах CCS.
24. Описание процессов в терминах CSP.
25. Основные операции композиции процессов в терминах CSP.

26. Темпоральная логика и ветвление времени в программах.

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

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 экзамене

К экзамену допускаются обучающиеся, выполнившие программу обучения по дисциплине, прошедшие все этапы текущей аттестации.

Баллы	Оценка зачета/ экзамена	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.

Оценочные средства для текущей аттестации

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

Текущая аттестация проводится в форме собеседования (УО-1) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы;
- результаты самостоятельной работы.

Составляется календарный план контрольных мероприятий по дисциплине. Оценка посещаемости, активности обучающихся на занятиях, своевременность выполнения различных видов заданий ведётся на основе журнала, который ведёт преподаватель в течение учебного семестра.

Студенты получают индивидуальные задания. В процессе их выполнения должны быть разработаны: формальная постановка задачи, алгоритмы ее решения, написана программа на любом языке программирования, для которой созданы тесты. Правильность работы программы демонстрируется с помощью созданных тестов. Преподаватель вправе задать свои значения входных данных.

Вопросы к устному опросу

1. Для автоматизации создания баз данных на основе их проекта используются
 - а. CASE средства
 - б. средства автоматизации проектирования пользовательского интерфейса
 - в. средства логирования программ
2. В основе мутационного способа тестирования программ лежит

- а. написание как можно большего числа тестов, перебирающих все возможные диапазоны входных данных
- б. тестирование всех возможных путей выполнения программ
- в. создание программного кода со специально вносимыми незначительными ошибками**

3. Оценка эффективности процесса разработки и качества программ реализуется

- а. программой измерений**
- б. выборочными аудитами
- в. Инспекциями

4. Коллективная разработка программ, как правило, требует **использования систем контроля версий**

- б. предварительного создания требований к ПО
- в. проведения формальных инспекций

5. Изменение внутренней архитектуры программ без изменения их функциональности называется

- а. рефакторинг**
- б. тестирование
- в. логирование

6. Модель разработки ПО, не требующая предварительной разработки полных и детальных требований, называется

- а. водопадной
- б. Agile**
- в. объектно-ориентированной

7. Ежедневные короткие встречи коллектива разработчиков для обмена мнениями и проблемами характерны для

- а. технологии разработки Agile**
- б. водопадной модели разработки ПО
- в. инкрементной модели разработки ПО

8. Внутренней верификации в коллективе разработчиков (инспекции) подвергаются
- а. все создаваемые рабочие продукты
 - б. только код программ
 - в. только требования

Критерии оценивания

Баллы (рейтингов ой оценки)	Оценка зачета (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «зачтено» / «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 «зачтено» / «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 «зачтено» / «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 «не зачтено» / «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.

