

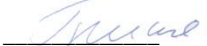


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»

Руководитель ОП
«Математическое моделирование, численные методы и
комплексы программ»

 Гриняк В. М.
(подпись) (Ф.И.О.)
« 09 » _____ июля _____ 2018 г.

«УТВЕРЖДАЮ»

Заведующий кафедрой
Информатики, математического и компьютерного
моделирования

 Чеботарев А.Ю.
(подпись) (Ф.И.О.)
« 09 » _____ июля _____ 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Методы создания программных систем моделирования

Направление подготовки 09.06.01 Информатика и вычислительная техника
Профиль «Математическое моделирование, численные методы и комплексы программ»
Форма подготовки (очная)

курс 2 семестр 4

лекции 18 час.

практические занятия 18 час.

лабораторные работы не предусмотрены.

с использованием МАО лек. 0/пр. 18/лаб. 0 час.

всего часов контактной работы 36 час.

в том числе с использованием МАО 18 час., в электронной форме 0 час.

самостоятельная работа 72 часа.

в том числе на подготовку к экзамену 18 час.

курсовая работа / курсовой проект не предусмотрена

зачет __ семестр

экзамен 4 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (уровень подготовки кадров высшей квалификации), утвержденного приказом Министерства образования и науки РФ от 30.07.14 № 875

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 18 от «09» июля 2018 г.

Заведующий кафедрой Информатики, математического и компьютерного моделирования, д.ф.-м.н., профессор Чеботарев А.Ю.

Составитель: д-р техн. наук, доцент, профессор кафедры прикладной математики, механики, управления и программного обеспечения Гриняк В.М.

Оборотная сторона титульного листа

I. Рабочая программа пересмотрена на заседании кафедры / академического департамента:

Протокол от «11» июня 2019 г. № 11

Заведующий кафедрой Информатики, математического и компьютерного моделирования



(подпись)

____ Чеботарев А.Ю. _____
(И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры (академического департамента):

Протокол от «18» января 2020 г. № 5

Заведующий кафедрой Информатики и компьютерного моделирования



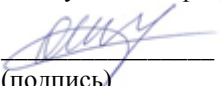
(подпись)

Чеботарев А.Ю.
(И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры (академического департамента):

Протокол от «27» января 2021 г. № 4

Заведующий кафедрой/директор академического департамента



(подпись)

Чеботарев А.Ю.
(И.О. Фамилия)

АННОТАЦИЯ

Рабочая программа дисциплины «Методы создания программных систем моделирования» разработана для аспирантов, обучающихся по направлению 09.06.01 «Информатика и вычислительная техника», профиль «Математическое моделирование, численные методы и комплексы программ».

Трудоемкость дисциплины составляет 3 зачетных единицы (108 часов). Трудоемкость контактной работы (по учебным занятиям) составляет 36 часов, в том числе 18 часов в интерактивной форме. На самостоятельную работу отводится 72 часа. Дисциплина реализуется на втором году обучения в 3, 4 семестрах. Формы контроля – зачет и экзамен.

В 3 семестре трудоемкость дисциплины составляет 1 зачетную единицу (36 часов). Трудоемкость лекций в 3 семестре составляет 9 часов. Трудоемкость лабораторных работ в 3 семестре составляет 0 часов. Трудоемкость практических занятий в 3 семестре составляет 9 часов, в том числе 9 часов в интерактивной форме. На самостоятельную работу в 3 семестре отводится 18 часов.

В 4 семестре трудоемкость дисциплины составляет 2 зачетных единицы (72 часа). Трудоемкость лекций в 4 семестре составляет 9 часов. Трудоемкость лабораторных работ в 4 семестре составляет 0 часов. Трудоемкость практических занятий в 4 семестре составляет 9 часов, в том числе 9 часов в интерактивной форме. На самостоятельную работу в 4 семестре отводится 54 часа, в том числе 18 часов – на подготовку к экзамену.

Курс «Методы создания программных систем моделирования» входит в вариативную часть учебного плана подготовки аспирантов по научной специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ».

Дисциплина «Методы создания программных систем моделирования» базируется на дисциплинах, связанных с анализом профессиональной деятельности и построением их моделей, а также проектированием и разработкой программного обеспечения, изучаемых в бакалавриате и магистратуре.

Знания, полученные при изучении курса «Методы создания программных систем моделирования», будут востребованы при подготовке к сдаче кандидатского экзамена по научной специальности 05.13.18 «Математическое моделирование, численные методы и комплексы программ», в научно-исследовательской работе, при подготовке выпускной работы и диссертации на соискание ученой степени кандидата технических наук.

Цель дисциплины - формирование теоретических знаний и практических навыков создания программных систем, решаемых при этом прикладных задач.

Задачи дисциплины:

1. Формирование представлений о специфике разработки классических приложений, интернет-систем, распределённых приложений, клиент-серверных систем, интеллектуальных систем.
2. Изучение методов разработки, обоснования и исследования моделей, методов, алгоритмов и программной инфраструктуры, требуемых для создания средств автоматизации профессиональной деятельности в различных предметных областях
3. Изучение методов проведения системного анализа автоматизируемой профессиональной деятельности, предметных областей, решаемых прикладных задач с целью определения свойств прикладных программных систем.
4. Развитие умения анализировать требования и на их основе выбирать современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения.
5. Приобретение навыка применения методов обоснования моделей профессиональной деятельности и предметных областей, спецификации прикладных задач, методов и алгоритмов решения задач, программной инфраструктуры, требуемой при создании программных систем для автоматизации профессиональной деятельности.
6. Приобретение навыков работы с современными инструментальными средствами, предназначенными для создания прикладных программных систем различного назначения.
7. Овладение методологией теоретических и экспериментальных исследований в области разработки программных систем.
8. Овладение приёмами организации работы исследовательского и производственного коллектива в области разработки программных систем.

Для успешного изучения дисциплины «Методы создания программных систем моделирования» у обучающихся должны быть сформированы следующие *предварительные компетенции*:

- способность к самоорганизации и самообразованию;
- способность к коммуникации в устной и письменных формах на русском и иностранных языках для решения задач межличностного и межкультурного взаимодействия;
- способность работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия;
- способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- способность к определению общих форм и закономерностей отдельной предметной области;
- способность публично представлять собственные и известные научные результаты;

– способность использовать методы математического и алгоритмического моделирования при анализе управленческих задач в научно-технической сфере, в экономике, бизнесе и гуманитарных областях.

В результате изучения дисциплины у аспирантов формируются следующие универсальные / общепрофессиональные / профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-4 Готовность организовать работу исследовательского коллектива в области профессиональной деятельности	Знает	- нормативно-правовые основы по организации коллективов исследователей в области информатики и вычислительной техники; - методологию проведения исследований коллективом разработчиков.
	Умеет	- анализировать, сравнивать и обосновывать альтернативные методы исследования, предлагаемые коллективом разработчиков; - применять современные сетевые технологии для организации работы коллектива в области информатики и вычислительной техники;
	Владеет	методологией организации работы исследовательского коллектива.
ПК-4 Способность к разработке и обоснованию комплексов проблемно-ориентированных программ для компьютерного моделирования предметных областей и проведения вычислительных экспериментов	Знает	- Технологию разработки прикладных систем, используемых для автоматизации профессиональной деятельности в различных областях, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных; - Современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.
	Умеет	Анализировать требования и на их основе выбирать современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.
	Владеет	Методами обоснования выбора инструментальных средств, предназначенных для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.

Для формирования вышеуказанных компетенций в рамках дисциплины «Методы создания программных систем моделирования» применяются

следующие методы активного / интерактивного обучения: *учебная дискуссия, решение исследовательской задачи, «мозговой штурм», метод проектов.*

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

(18 час., в том числе 0 час. с использованием методов активного обучения)

Занятие 1. Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности (2 часа)

Профессиональные позиции ИТ специалистов и их особенности. Особенности работы в отечественных и международных ИТ компаниях. Международный стандарт ISO/IEC 12207:1995. Отечественные профессиональные стандарты ИТ специалистов. Особенности исследовательской деятельности в области ИТ. Нормативно-правовые документы, обеспечивающие научно-исследовательскую деятельность и устанавливающие критерии её эффективности. Понятие интеллектуальной собственности в области разработки ПО и её формы. Понятие коллективной разработки программного обеспечения. Средства поддержки процесса коллективной разработки ПО.

Занятие 2. Модели жизненного цикла разработки программных средств (2 часа)

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

Занятие 3. Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов (2 часа)

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

Занятие 4. Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов (2 часа)

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

Занятие 5. Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников (2 часа)

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

Занятие 6. Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС (2 часа)

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

Занятие 7. Модель зрелости организации CMM (2 часа)

Сущность модели CMM и история её создания. Уровни зрелости: Initial, Repeatable, Defined, Managed, Optimizing. Ключевые виды деятельности каждого уровня. Аттестация предприятий по системе CMM. Методы достижений уровня зрелости. Альтернативные подходы к оценке уровня зрелости организации.

Занятие 8. Стандарты и хорошие практики кодирования на языках высокого уровня (2 часа)

Стандарты кодирования и их назначение. Категории правил кодирования: запреты, требования, рекомендации. Примеры запретов, требований и рекомендаций. Примеры оформления кода модулей и заголовочных файлов. Revision History. Назначение и примеры чек-листов для разработчиков. Стандарты кодирования и системы контроля версий. Специфика управления проектами при использовании стандартов кодирования. Паттерны проектирования (design patterns).

Занятие 9. Адаптация классических технологий создания программных систем для разработки интернет-приложений (2 часа)

Разделение frontend/backend разработки, их специфика. Команда проекта. Идеология «вечной беты». Итеративный подход к разработке. Оптимизация нагруженных интернет-проектов. Упаковка стилей и библиотек. Оптимизация работы с БД. Кэширование, виды и стратегии кэширования. Nginx, его преимущества. Интернет-специфика безопасности. Виды атак. [D]DoS, Code/SQL инъекции, XSS-уязвимости, phishing. Типичные ошибки при написании кода, методы защиты. Назначение, основные принципы устройства популярных фреймворков. DRY. Паттерны MVC, ORM. URL dispatching. Шаблонирование и кэширование.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

(18 час., в том числе 18 час. с использованием методов активного обучения)

Практические занятия (18/18 час.)

Занятие 1. Организация коллективной работы разработчиков (2/2 часа)

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

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Беседа.

Занятие 3. Работа с требованиями к программному продукту (2/2 часа)

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

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Учебная дискуссия;

Занятие 4. Кодирование программной системы (2/2 часа)

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

Методы активного обучения:

1. Решение исследовательской задачи.

Занятие 5. Тестирование программной системы (2/2 часа)

1. Подготовка документа «План тестирования проекта»
2. Подготовка документа «Матрица покрытия требований»

Разработка политики по прогону тестов на проекте (модульное, системное тестирование)

Методы активного обучения:

1. Решение исследовательской задачи;
2. Метод проектов.

Занятие 6. Доработка программной системы (2/2 часа)

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

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Решение исследовательской задачи;
3. Метод проектов.

Занятие 7. Метрики качества разработки программной системы (2/2 часа)

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

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Решение исследовательской задачи;
3. Метод проектов.

Занятие 8. Метрики эффективности процесса разработки программной системы (2/2 часа)

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

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Учебная дискуссия;
3. «Мозговой штурм».

Занятие 9. Выработка рекомендаций по изменению процесса разработки (2/2 часа)

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

2. Формулировка рекомендаций по трансформации процесса разработки по результатам программы измерений

Методы активного обучения:

1. Метод анализа конкретных ситуаций;
2. Учебная дискуссия;
3. «Мозговой штурм».

Особенности проведения практических занятий (активные и интерактивные формы)

Практические занятия проводятся в форме деловой игры. Особенностью учебного процесса по дисциплине является его максимальное приближение к реальным условиям работы на проектах по коллективной разработке ПО: как коммерческим, так и фрилансовым. Всем аспирантам необходимо создать аккаунт на портале github.com. Разработка проектов ведётся на базе именно этой системы контроля версий с использованием встроенной системы issue трекинга.

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

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

В обязанности кодировщиков входит текущая работа над проектом, в том числе - написание кода согласно 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
- `REQUIREMENTS` - для размещения документов с требованиями
- `DESIGN` - для размещения документов дизайна
- `CODING` - для размещения документов кодирования
- `TESTING` - для размещения документов тестирования

Работа с системой контроля версий

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

Проект должен иметь двухуровневую структуру веток.

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

Документация, создаваемая по проекту

В рамках работы над проектом создается следующая документация:

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

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

Лабораторные работы (0/0 час.)

Курс не предусматривает лабораторных работ.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Методы создания программных систем моделирования» 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 ЦЕЛИ КУРСА

3 семестр					
№ п/ п	Контролируемые разделы / темы дисциплины	Коды, наименование и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежу точная аттестация
1	Занятие 1 Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 1-4
			Умеет Владеет	ПР-9 Проект	
2	Занятие 2 Модели жизненного цикла разработки программных средств	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 5-9
3	Занятие 3 Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 10-14

			Умеет Владеет	ПР-9 Проект	
4	Занятие 4 Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 14-17
			Умеет Владеет	ПР-9 Проект	
4 семестр					
5	Занятие 5 Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 1-5
			Умеет Владеет	ПР-9 Проект	
6	Занятие 6 Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 6-10
			Умеет Владеет	ПР-9 Проект	
7	Занятие 7 Модель зрелости организации СММ	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 11-15
			Умеет Владеет	ПР-9 Проект	
8	Занятие 8 Стандарты и хорошие практики кодирования на языках высокого уровня	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 15-19
9	Занятие 9 Адаптация классических технологий создания программных систем для разработки интернет-приложений	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 20-22

Фонд оценочных средств по дисциплине представлен в приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В. П. Котляров, Т. В. Коликова. Москва: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2012. – 285 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:668103&theme=FEFU>

2. 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>
3. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с. Режим доступа: <http://znanium.com/go.php?id=389963>
4. Введение в архитектуру программного обеспечения: Учебное пособие / Гагарина Л.Г., Федоров А.Р., Федоров П.А. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 320 с.: 60x90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0649-1 <http://znanium.com/catalog.php?bookinfo=542665>
5. Шопырин, Д.Г. Управление проектами разработки ПО. Дисциплина «Гибкие технологии разработки программного обеспечения». [Электронный ресурс] — Электрон. дан. — СПб. : НИУ ИТМО, 2007. — 131 с. — Режим доступа: <http://e.lanbook.com/book/43554>
6. Дьяконов, В.П. MATLAB. Полный самоучитель [Электронный ресурс] / В.П. Дьяконов. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 768 с. — 978-5-4488-0065-8. — Режим доступа: <http://www.iprbookshop.ru/63590.html>

Дополнительная литература

1. Соммервилл И. Инженерия программного обеспечения. 6-е издание. М.: Изд. дом Вильямс, 2002. – 624 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:735&theme=FEFU>
2. Браудэ Э. Технология разработки программного обеспечения, Издательский дом «Питер», 2004. - 656 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:232388&theme=FEFU>
3. Иванова Г.С. Технология программирования. М.: Изд-во МГУ. 2002. <http://lib.dvfu.ru:8080/lib/item?id=chamo:398553&theme=FEFU>
4. Жоголев Е.А. Технология программирования. М.: Научный мир. 2004. – 215 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:7891&theme=FEFU>
5. Бенкен Е.С. PHP, MySQL, XML программирование для Интернета. СПб: БХВ-Петербург. -2008. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:382736&theme=FEFU>
6. Скляр Д., Трахтенберг А. PHP. Рецепты программирования. – СПб: Русская редакция БХВ-Петербург. -2007. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:381965&theme=FEFU>
7. Липаев, В.В. Программная инженерия. Методологические основы [Текст]: Учеб. / В. В. Липаев; Гос. ун-т — Высшая школа экономики. — М. : ТЕИС, 2006. — 608 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:248067&theme=FEFU>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <http://window.edu.ru/resource/711/79711> Липаев В.В. Проектирование и производство сложных заказных программных продуктов. - М.: СИНТЕГ, 2011. - 408 с
2. <http://novtex.ru/> Журналы издательства Новые технологии
3. <https://github.com/> Система контроля версий Git
4. <http://matlab.exponenta.ru/> Матлаб и Симулинк – сообщество пользователей
5. <https://matlab.ru/> Матлаб и Симулинк – центр компетенций

Перечень информационных технологий и программного обеспечения

Интегрированные среды разработчика Eclipse, NetBeans, Visual Studio.
Система контроля версий Git. Офисное ПО, пакет Matlab.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина изучается в следующих организационных формах: лекционные, практические занятия и самостоятельная работа аспиранта. Аспирант должен планировать график самостоятельной работы по дисциплине и придерживаться его.

Основной формой самостоятельной работы аспиранта является выполнение проекта, а также подготовка докладов для практических занятий.

К практическим занятиям следует готовиться. Для этого необходимо знать программу курса и рекомендованную литературу. Необходимо повторить основные разделы таких курсов, как «Технология разработки программного обеспечения», «Методы системного анализа и моделирования» и «Математическая логика», чтобы осваивать новый материал более эффективно. Рекомендуется также владение хотя бы одним из функциональных и логических языков программирования для успешного освоения дисциплины. Аспиранту необходимо активно участвовать в дискуссиях, не бояться задавать вопросы преподавателю и другим участникам.

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

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Практические занятия проводятся в специализированном компьютерном

классе. Необходимо оборудование для демонстрации презентаций: компьютер, проектор, монитор. Компьютер должен быть оснащен следующим программным обеспечением: LibreOffice или Microsoft Word, а также Microsoft PowerPoint.



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

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине «Методы создания программных систем моделирования»

Направление подготовки *09.06.01 Информатика и вычислительная техника*
Профиль «*Математическое моделирование, численные методы и комплексы программ*»
Форма подготовки (очная)

**Владивосток
2018**

План-график выполнения самостоятельной работы по дисциплине

3 семестр

№ п/п	Дата/сроки выполнения	Виды СРС	Всего часов	Форма контроля
1.	1-4 неделя обучения	Изучение теоретического материала к ПЗ-1 по литературным источникам	1	Собеседование
		Выполнение группового проекта	1	Проверка проекта
2.	5-8 неделя обучения	Изучение теоретического материала к ПЗ-2 по литературным источникам	1	Собеседование
		Выполнение группового проекта	1	Проверка доклада
3.	9-12 неделя обучения	Изучение теоретического материала к ПЗ-3 по литературным источникам	2	Собеседование
		Выполнение группового проекта	2	Проверка проекта
4.	13-16 неделя обучения	Изучение теоретического материала к ПЗ-4 по литературным источникам	2	Собеседование
		Выполнение группового проекта	2	Проверка проекта
5.	17-18 неделя обучения	Изучение теоретического материала к ПЗ-5 по литературным источникам	2	Собеседование
		Выполнение группового проекта	2	Проверка проекта
6.	17 -18 неделя обучения	Подготовка к промежуточной аттестации.	2	зачет
		ВСЕГО	18	

4 семестр

№ п/п	Дата/сроки выполнения	Виды СРС	Всего часов	Форма контроля
1.	1-4 неделя обучения	Изучение теоретического материала к ПЗ-6 по литературным источникам	4	Собеседование
		Выполнение группового проекта	4	Проверка проекта
2.	5-8 неделя обучения	Изучение теоретического материала к ПЗ-7 по литературным источникам	4	Собеседование
		Выполнение группового проекта	4	Проверка проекта
3.	9-12 неделя обучения	Изучение теоретического материала к ПЗ-8 по литературным источникам	4	Собеседование
		Выполнение группового проекта	4	Проверка

				проекта
4.	13-16 неделя обучения	Изучение теоретического материала к ПЗ-9 по литературным источникам	4	Собеседование
		Выполнение группового проекта	4	Проверка проекта
5.	17-18 неделя обучения	Выполнение группового проекта	4	Проверка проекта
6.	17 -18 неделя обучения	Подготовка к промежуточной аттестации.	18	Экзамен
		ВСЕГО	54	

Рекомендации по самостоятельной работе студентов **Рекомендации по работе с литературой**

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения практического занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;
- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

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

Методические указания по подготовке к практическим занятиям

Подготовку к каждому практическому занятию каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на

теоретические вопросы по теме задания, правильном выполнении лабораторной работы.

В процессе практического занятия студент должен создать требуемый документ с помощью предлагаемого программного средства и выполнить требуемые в задании операции, либо подготовить к дискуссии теоретический материал по предложенной теме.

Критерии оценки лабораторных(практических) работ

– 100-86 - выполнены все задания практической (лабораторной) работы, студент четко и без ошибок ответил на все контрольные вопросы.

– 85-76 - выполнены все задания практической (лабораторной) работы; студент ответил на все контрольные вопросы с замечаниями.

– 75-61 выполнены все задания практической (лабораторной) работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

- 60-50 баллов - студент не выполнил или выполнил неправильно задания практической (лабораторной) работы; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.



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 «Методы создания программных систем моделирования»
Направление подготовки *09.06.01 Информатика и вычислительная техника*
Профиль *«Математическое моделирование, численные методы и комплексы программ»*

Форма подготовки (очная/заочная)

Владивосток
2018

Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-4 Готовность организовать работу исследовательского коллектива в области профессиональной деятельности	Знает	- нормативно-правовые основы по организации коллективов исследователей в области информатики и вычислительной техники; - методологию проведения исследований коллективом разработчиков.
	Умеет	- анализировать, сравнивать и обосновывать альтернативные методы исследования, предлагаемые коллективом разработчиков; - применять современные сетевые технологии для организации работы коллектива в области информатики и вычислительной техники;
	Владеет	методологией организации работы исследовательского коллектива.
ПК-4 Способность к разработке и обоснованию комплексов проблемно-ориентированных программ для компьютерного моделирования предметных областей и проведения вычислительных экспериментов	Знает	- Технологию разработки прикладных систем, используемых для автоматизации профессиональной деятельности в различных областях, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных; - Современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.
	Умеет	Анализировать требования и на их основе выбирать современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.
	Владеет	Методами обоснования выбора инструментальных средств, предназначенных для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.

3 семестр				
№ п/ п	Контролируемые разделы / темы дисциплины	Коды, наименование и этапы формирования компетенций	Оценочные средства	
			текущий контроль	промежуточная аттестация

1	Занятие 1 Профессиональная деятельность ИТ специалистов с точки зрения разработки программного обеспечения как отрасли промышленности	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 1-4
			Умеет Владеет	ПР-9 Проект	
2	Занятие 2 Модели жизненного цикла разработки программных средств	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 5-9
			Умеет Владеет	ПР-9 Проект	
3	Занятие 3 Специфика верификации рабочих продуктов при коллективной разработке. Принципы проведения и организации инспекций рабочих продуктов	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 10-14
			Умеет Владеет	ПР-9 Проект	
4	Занятие 4 Структура и анализ бизнес-процессов коллективной разработки программного обеспечения. Системы сопровождения заданий. Системы отслеживания дефектов	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Зачет, вопросы 14-17
			Умеет Владеет	ПР-9 Проект	
4 семестр					
5	Занятие 5 Структура бизнес-процессов коллективной разработки программного обеспечения: средства и методы сбора метрик сотрудников	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 1-5
			Умеет Владеет	ПР-9 Проект	
6	Занятие 6 Измерения при разработке и сопровождении программного продукта. Основные метрики эффективности процесса разработки и метрики качества ПС	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 6-10
			Умеет Владеет	ПР-9 Проект	
7	Занятие 7 Модель зрелости организации СММ	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 11-15
			Умеет Владеет	ПР-9 Проект	

8	Занятие 8 Стандарты и хорошие практики кодирования на языках высокого уровня	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 15-19
			Умеет Владеет	ПР-9 Проект	
9	Занятие 9 Адаптация классических технологий создания программных систем для разработки интернет-приложений	ОПК-4, ПК-4	Знает	УО-1 Собеседование	Экзамен вопросы 20-22
			Умеет Владеет	ПР-9 Проект	

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОПК-4 Готовность организовать работу исследовательского коллектива в области профессиональной деятельности	знает (пороговый уровень)	- нормативно-правовые основы по организации коллективов исследователей в области информатики и вычислительной техники; - методологию проведения исследований коллективом разработчиков.	Представления об основах и методах организации коллективов разработчиков	Способность дать ответы на вопросы о существующих методах
	умеет (продвинутый)	- анализировать, сравнивать и обосновывать альтернативные методы исследования, предлагаемые коллективом разработчиков; - применять современные сетевые технологии для организации работы коллектива в области информатики и вычислительной	Умение организовать работу коллектива разработчиков	Не менее 5 ролей разработчиков

		техники;		
	владеет (высокий)	методологией организации работы исследовательского коллектива.	владеет методологией разработки новых методов исследований и методологией их применения при решении задач в области информатики и вычислительной техники	Наличие методов в выполненных проектах
ПК-4 Способность к разработке и обоснованию комплексов проблемно-ориентированных программ для компьютерного моделирования предметных областей и проведения вычислительных экспериментов	знает (пороговый уровень)	- Технологию разработки прикладных систем, используемых для автоматизации профессиональной деятельности в различных областях, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных; - Современные инструментальные средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.	Сформированные представления о методах обоснования полученных результатов исследований с учетом специфики области информатики и вычислительной техники; Сформированные представления об основных особенностях и закономерностях развития области информатики и вычислительной техники	Способность дать ответы на вопросы
	умеет (продвинутый)	Анализировать требования и на их основе выбирать современные инструментальные	Анализ требований и выбор наиболее подходящих для решения научных проблем в области	Система из не менее чем 15 требований

		средства, предназначенные для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.	исследования	
	владеет (высокий)	Методами обоснования выбора инструментальных средств, предназначенных для создания прикладных программных систем различного назначения, в том числе интернет-систем, распределенных, клиент-серверных, интеллектуальных.	Формирование системы критериев	Не менее 5 критериев

Оценочные средства для промежуточной аттестации

3 семестр

1. В каких условиях лучше всего применять итеративно-инкрементную модель жизненного цикла ПО и почему.
2. Какой проектной документацией принято сопровождать процесс разработки программного обеспечения.
3. Опишите функции участников инспекции на каждом этапе.
4. Зачем необходимо собирать метрики по результатам инспекций и какие метрики существуют.
5. Опишите состояния, в которых может пребывать подзадача в процессе коллективной разработки программного продукта.
6. Для чего служит система CLEARDDTS.
7. Какие существуют проектные виды деятельности.
8. Назовите этапы регистрации затрат рабочего времени.
9. В чем состоит методика GQ(I)M.
10. Приведите примеры и описание метрик качества продуктов.
11. Опишите сущность модели CMM.

12. Какие ключевые виды деятельности должны иметь место в компании третьего уровня.
13. Зачем необходимы стандарты кодирования.
14. Приведите пример правил и рекомендаций по оформлению заголовков файлов и функций.
15. Опишите основные принципы тестирования программного средства.
16. Основная идея восходящего тестирования, достоинства и недостатки.
17. Что включает в себя стандартный процесс разработки ПО согласно современным парадигмам?

4 семестр

1. Назовите модели жизненных циклов разработки ПО, опишите подробно (основной принцип, достоинства и недостатки) любой из них.
2. Перечислите роли участников инспекции рабочих продуктов и опишите их, напишите функции одного из них на каждом этапе инспекции.
3. Опишите способ оценивания инспекций и приведите пример оценивания.
4. Опишите систему ClearDdts в 5-8 предложениях.
5. Какие понятия лежат в основе технологии Workflow.
6. Назовите 10 проектных видов деятельности.
7. Для каждого вида деятельности перечислите набор соответствующих им действий.
8. Приведите 3 примера метрик эффективности процесса производства.
9. Приведите 3 примера метрик качества продуктов.
10. Опишите (вкратце) процесс аттестации по модели СММ.
11. Опишите сущность модели СММ.
12. Напишите пример требования написания на любом языке.
13. Назовите наиболее известный стандарт программирования и причину его введения.
14. Назовите главный принцип тестирования. Перечислите фазы тестирования.
15. Что включает в себя комплексная отладка программного средства.
16. Опишите 5 классов проектов и сопровождающую их проектную документацию.
17. Опишите модели жизненных циклов разработки ПО Водопад и Водопад с перекрытиями. В чем различия.
18. Что такое инспекция.
19. Опишите этапы планирования и завершения инспекции в процедуре организации и проведения формальной инспекции.
20. Выскажите свое мнение о системе отслеживания дефектов CLEARDDTS
21. Какие понятия вложены в основу регистрации времени.

22. Перечислите проектные виды деятельности (не менее 8).
23. Нарисуйте схему иерархии методики GQ(I)M.
24. Примеры метрик качества продуктов.
25. Перечень КРА третьего уровня.
26. Что такое практики. Перечислите некоторые из них (не менее 3).
27. Для чего нужны общие и корпоративные правила и рекомендации.
28. Придумайте требование для ПО.
29. Каков главный принцип тестирования.
30. Опишите вторую задачу тестирования

Оценочные средства для текущего контроля

3 семестр

Темы групповых творческих заданий

Предлагаются студентами исходя из тематики их научной работы

Перечень тем для дискуссии

1. Существующие методы разработки программного обеспечения (классические). Их достоинства и недостатки.

Темы докладов

Темы докладов выбираются согласно темам практических занятий

4 семестр

Темы групповых творческих заданий

Предлагаются студентами исходя из тематики их научной работы

Перечень тем для дискуссии

1. Существующие методы разработки программного обеспечения (нетрадиционные). Их достоинства и недостатки.

Темы докладов

Темы докладов выбираются согласно темам практических занятий

Текущий контроль

Текущий контроль предполагает систематическую проверку усвоения учебного материала, сформированности компетенций или их элементов, регулярно осуществляемую на протяжении изучения дисциплины, в соответствии с ее рабочей программой.

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 аспирант точно определил содержание и составляющие части задания, умеет аргументировано 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 аспиранта 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 проектов

Менее 60 баллов	Не зачтено
От 61 до 75 баллов	зачтено
От 76 до 85 баллов	зачтено
От 86 до 100 баллов	зачтено