ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Директор института
Институт естественных и точных наук

Электронный документ, подписанный ПЭП, хранится в системе электронного документооборота Южно-Уральского государственного университета
СВЕДЕНИЯ О ВЛАДЕЛЬЦЕ ПЭП
Кому выдан: Замышляева А. А.
Пользователь: zamyshliaevaaa
Дата подписания: 30.01.2022

А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.Ф.П1.21.01 Анализ требований и проектирование ПО
для направления 02.03.01 Математика и компьютерные науки
уровень Бакалавриат
профиль подготовки Компьютерное моделирование в инженерном и технологическом проектировании
форма обучения очная
кафедра-разработчик Прикладная математика и программирование

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 02.03.01 Математика и компьютерные науки, утверждённым приказом Минобрнауки от 23.08.2017 № 807

Зав.кафедрой разработчика,
д.физ.-мат.н., проф.

Электронный документ, подписанный ПЭП, хранится в системе электронного документооборота Южно-Уральского государственного университета
СВЕДЕНИЯ О ВЛАДЕЛЬЦЕ ПЭП
Кому выдан: Замышляева А. А.
Пользователь: zamyshliaevaaa
Дата подписания: 29.01.2022

А. А. Замышляева

Разработчик программы,
доцент

Электронный документ, подписанный ПЭП, хранится в системе электронного документооборота Южно-Уральского государственного университета
СВЕДЕНИЯ О ВЛАДЕЛЬЦЕ ПЭП
Кому выдан: Демидов А. К.
Пользователь: demidovak
Дата подписания: 29.01.2022

А. К. Демидов

СОГЛАСОВАНО

Руководитель образовательной программы
д.физ.-мат.н., проф.

Электронный документ, подписанный ПЭП, хранится в системе электронного документооборота Южно-Уральского государственного университета
СВЕДЕНИЯ О ВЛАДЕЛЬЦЕ ПЭП
Кому выдан: Загребина С. А.
Пользователь: zagrebinasa
Дата подписания: 29.01.2022

С. А. Загребина

Челябинск

1. Цели и задачи дисциплины

Преподаваемая дисциплина посвящена изучению двух важных этапов жизненного цикла программного обеспечения - анализу предметной области и проектированию архитектуры программных систем. Преподавание и изучение дисциплины следует рассматривать как полезную составляющую профессиональной подготовки. Целью преподавания дисциплины является обучение студентов методам выявления, анализа и разработки требований и методам проектирования сложных программных систем. Задачи дисциплины заключаются в том, чтобы студенты овладели навыками анализа предметной области, создания и описания объектно-ориентированных моделей предметной области, выполнения системного анализа и разработки на его основе архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения; навыками создания спецификаций, как для всей системы в целом, так для отдельных подсистем и модулей.

Краткое содержание дисциплины

Основы управления требованиями к программному обеспечению. Методы выявления требований и заинтересованных лиц. Спецификация требований. Требования к типичным системам. Согласование требований и управление рисками. Проблемы разработки сложных программ. Структурный подход в проектировании ПО и классификация структурных методологий. Архитектура ПО, влияние архитектуры на свойства ПО. Унифицированный язык моделирования UML. Применения шаблонов проектирования к прикладным задачам. Исследование существующих архитектур промежуточного программного обеспечения. Компонентное проектирование. Классификация CASE-систем и их сравнительная характеристика.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ПК-1 Способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности	Умеет: решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности Имеет практический опыт: использования решений практических задач на основе математических и естественнонаучных подходов
ПК-3 Способен создавать и исследовать математические модели в естественных науках и промышленности, с учетом возможностей современных информационных технологий и программирования и компьютерной техники	Умеет: использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание

	программного продукта

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Программирование на языке Java, САПР технологических процессов, Искусственный интеллект и нейронные сети, Современные технологии разработки программного обеспечения, Web-программирование, Нейроматематика, Математическое моделирование физических и технических процессов, Производственная практика, научно-исследовательская работа (6 семестр)	Не предусмотрены

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
Искусственный интеллект и нейронные сети	Знает: Умеет: применять базовые методы математических и естественных наук, программирования и информационных технологий, использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий, применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта
САПР технологических процессов	Знает: Умеет: применять базовые методы математических и естественных наук, программирования и информационных технологий, использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий, применения методов

	проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта
Математическое моделирование физических и технических процессов	Знает: методы исследования математических моделей физических и технических процессов, способы выбора решения практических задач на основе математических и естественнонаучных подходов Умеет: применять методы исследования математических моделей физических и технических процессов, решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности Имеет практический опыт: исследования математических моделей физических и технических процессов, использования решений практических задач на основе математических и естественнонаучных подходов
Программирование на языке Java	Знает: основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции) Умеет: использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий, применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта
Нейроматематика	Знает: Умеет: использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта, применять базовые методы математических и естественных наук, программирования и информационных технологий Имеет практический опыт: применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами,

	поддерживающими создание программного продукта, использования базовых методов математических и естественных наук, программирования и информационных технологий
Web-программирование	Знает: основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции) Умеет: использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий, применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта
Современные технологии разработки программного обеспечения	Знает: основные технологии разработки программного обеспечения, основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции) Умеет: работать с основными технологиями разработки программного обеспечения, использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта Имеет практический опыт: использования основных технологий разработки программного обеспечения, применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта
Производственная практика, научно-исследовательская работа (6 семестр)	Знает: Умеет: анализировать и систематизировать полученную информацию, выбирать приёмы и методы её обработки Имеет практический опыт: определения и решения круга задач в рамках поставленной цели,

	применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта, использования базовых методов математических и естественных наук, программирования и информационных технологий, применения основных методов обработки информации для решения практических задач, самостоятельного принятия обоснованных экономических решений в своей жизнедеятельности, использования необходимой информации из текстов профессиональной направленности

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 2 з.е., 72 ч., 52,25 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
		8
Общая трудоёмкость дисциплины	72	72
Аудиторные занятия:	48	48
Лекции (Л)	24	24
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	24	24
Лабораторные работы (ЛР)	0	0
Самостоятельная работа (СРС)	19,75	19,75
с применением дистанционных образовательных технологий	0	
Изучение теоретического материала к практическим занятиям	11,75	11.75
Подготовка к дифференцированному зачету	8	8
Консультации и промежуточная аттестация	4,25	4,25
Вид контроля (зачет, диф.зачет, экзамен)	-	диф.зачет

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Разработка и анализ требований	24	12	12	0
2	Проектирование архитектуры и разработка спецификаций модулей	24	12	12	0

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Основы управления требованиями к программному обеспечению. Процессы инженерии требований: выявление требований, спецификация, анализ и управление.	2
2	1	Типы требований: функциональные, нефункциональные, атрибуты качества. Потребности и функции продукта.	2
3	1	Выявление требований: определение потребностей, целей и требований. Заказчики и другие заинтересованные лица. Интервью, наблюдения, совещания, мозговой штурм, раскадровки, прототипирование.	2
4	1	Спецификация требований: текстовые и графические нотации и языки (UML, нотации пользовательских требований). Методы написания высококачественных требований. Стандарты документирования.	2
5	1	Анализ требований: инспекция, аттестация, завершенность, обнаружение конфликтов и несоответствий. Анализ взаимодействия элементов функциональности и разрешение противоречий.	2
6	1	Требования к типичным системам: встроенным системам, потребительским системам, web-системам, бизнес-системам, научным системам и другим инженерным системам. Согласование требований и управление рисками. Интеграция анализа требований и процессов разработки программного обеспечения (включая Agile-процессы).	2
7	2	Проблемы разработки сложных программ. Структурный подход в проектировании ПО и классификация структурных методологий. Диаграммы «сущность-связь» (ERD), диаграммы потоков данных (DFD), SADT-модели (стандарт IDEF0).	2
8	2	Архитектура ПО, влияние архитектуры на свойства ПО. Особенности разработки сложных программных систем: иерархичность, групповая разработка, сборочное проектирование. Преимущества и недостатки объектно-ориентированного подхода.	2
9	2	Основные понятия унифицированного языка моделирования (UML). Диаграммы прецедентов, диаграммы классов, диаграммы взаимодействий, диаграммы последовательности действий, диаграммы состояний, компонентные диаграммы.	2
10	2	Углубленное изучение проектирования программного обеспечения, шаблонов проектирования, сред разработки и архитектур. Применения шаблонов проектирования к прикладным задачам.	2
11	2	Проектирование распределенных систем с использованием промежуточного программного обеспечения (middleware). Изучение архитектур COM, CORBA и .NET. Исследование существующих архитектур промежуточного программного обеспечения. Компонентное проектирование.	2
12	2	Классификация CASE-систем и их сравнительная характеристика. Тенденции развития объектно-ориентированных инструментальных средств. Поддержка графических моделей. Репозитарий и контроль ошибок. Теория измерения и использование метрик в проектировании. Проектирование с учетом таких качеств, как производительность, безопасность, защищенность, возможность повторного использования, надежность и т.д. Измерение внутренних параметров и сложности программного обеспечения.	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов

1	1	Анализ множества существующих программных систем: измерение качества и восстановление требований (reverse engineering) по программе. Использование инструментов управления требованиями.	2
2	1	Анализ проблемы. Выявление причин проблемы.	2
3	1	Моделирование, прототипирование и анализ требований с помощью средств UML	2
4	1	Создание документа-концепции для учебного проекта	2
5,6	1	Создание спецификации для учебного проекта	4
7	2	Разработка архитектуры учебного проекта	2
8	2	Разработка диаграммы классов для учебного проекта	2
9	2	Создание спецификации для нескольких модулей учебного проекта	2
10	2	Разработка диаграмм взаимодействий, диаграмм последовательности действий для учебного проекта	2
11,12	2	Проектирование распределенных систем с использованием промежуточного программного обеспечения. Использование CASE-средств	4

5.3. Лабораторные работы

Не предусмотрены

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Изучение теоретического материала к практическим занятиям	ЭУМД,осн.лит.1.с.6-250, ЭУМД,доп.лит.2, гл.4,5,8,17,18,20, ЭУМД,осн.лит.3, гл.3,4,6-10	8	11,75
Подготовка к дифференцированному зачету	ЭУМД,осн.лит.1.с.6-250, ЭУМД, осн.лит.3, гл.3,4,6-10	8	8

6. Текущий контроль успеваемости, промежуточная аттестация

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-мест	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учи-тыва-ется в ПА
1	8	Текущий контроль	Задание 1	1	10	Выполнена регистрация участников группы в СУТ — 1 балл Зафиксированы не менее 5 существующих требований к ПО (reverse engineering) —	дифференцированный зачет

						5 баллов (по 1 баллу за каждое требование) Предложено изменение требования, улучшение функциональности ПО - 2 балла, иначе 0 баллов Выполнена классификация требований — 2 балла, иначе 0 баллов	
2	8	Текущий контроль	Задание 2	1	10	Поставлена реальная проблема(ы) - 1 балл Указано не менее 5 причин для проблем - 5 баллов (по 1 баллу за причину) Есть причины причин - 1 балл Есть причины, которые можно решить программным путем - 1 балл Есть список из не менее 2 конкурирующих продуктов, которые будут использованы для сравнения (ссылки) - 2 балла (по 1 баллу за продукт)	дифференцированный зачет
3	8	Текущий контроль	Задание 3	1	10	Нарисована диаграмма вариантов использования для всех прямых пользователей - 3 балла (оценка снижается на 1 балл за каждого отсутствующего пользователя) Варианты использования соответствуют потребностям и целям пользователей - 2 балла (оценка снижается на 1 балл за каждую ошибку) Есть описание одного из вариантов использования - 3 балла, иначе 0 баллов В описании нет противоречий - 2	дифференцированный зачет

						балла (оценка снижается на 1 балл за каждую ошибку)	
4	8	Текущий контроль	Задание 4	1	10	Структура документа-концепции соответствует шаблону - 2 балла, иначе 0 баллов Достаточное описание пользователя и его потребностей (пп 2.2,2.3,2.4) - 3 балла (оценка снижается на 1 балл за каждую ошибку) Рассмотрено не менее 2 альтернатив - 2 балла, 1 альтернатива - 1 балл, иначе 0 баллов В п.3 указано достаточное обоснование важности проекта - 1 балл Указано не менее 5 функций - 2 балла, от 3 до 4 функций - 1 балл, менее 3 функций - 0 баллов	дифференцированный зачет
5	8	Текущий контроль	Задание 5	1	10	Структура спецификации соответствует шаблону - 2 балла, иначе 0 баллов Выполняется трассируемость функций из документа концепции - 1 балл Имеется подробная спецификация не менее 5 функциональных требований - 2 балла, 3-4 требования - 1 балл, менее 3 - 0 баллов Имеется подробная спецификация не менее 3 нефункциональных требований - 2 балла, 2 требования - 1 балл, , менее 2 - 0 баллов Указана спецификация для интерфейсов пользователя (не менее 3 диалогов) - 2 балла, 2	дифференцированный зачет

						диалога - 1 балл, , менее 2 - 0 баллов Указана спецификация для хотя бы 1 программного интерфейса (API) - 1 балл	
6	8	Текущий контроль	Задание 6	1	10	Выбран язык программирования - 1 балл Выбран фреймворк, библиотеки для реализации интерфейса пользователя - 2 балла, устаревший фреймворк - 1 балл, не выбран - 0 баллов Выбрана СУБД - 1 балл Сделано обоснование выбора - 1 балл Нарисована диаграмма компонентов (подсистем) - 1 балл Обозначения на диаграмме соответствуют стандарту UML - 2 балла, оценка снижается на 1 балл за каждую ошибку Подсистемы и компоненты обеспечивают выполнение требований (функций) ПО - 2 балла, , оценка снижается на 1 балл за каждую ошибку	дифференцированный зачет
7	8	Текущий контроль	Задание 7	1	10	Разработана диаграмма классов - 3 балла, иначе 0 баллов Правильно использованы обозначения UML - 3 балла, оценка снижается на 1 балл за каждую ошибку Каждому компоненту (задание 5) соответствует хотя бы один класс - 2 балла, отсутствует 1 компонент - 1 балл, отсутствует более 1	дифференцированный зачет

						компонента - 0 баллов Направление связей в иерархии классах обеспечивают устойчивость компонент - 2 балла, иначе 0 баллов	
8	8	Текущий контроль	Задание 8	1	10	Нарисована диаграмма классов (не менее 2 классов) - 3 балла, иначе 0 баллов Использованы корректные графические обозначения UML для классов и связей - 2 балла, оценка снижается на 1 балл за каждую ошибку В диаграмме классов поля имеют правильные описания - 1 балл В диаграмме классов методы имеют правильные описания - 1 балл Указаны спецификации не менее 3 методов - 3 балла (по 1 баллу за каждую спецификацию, но не более 3 баллов)	дифференцированный зачет
9	8	Текущий контроль	Задание 9	1	10	Нарисована диаграмма - 3 балла, иначе 0 баллов Использованы правильные обозначения UML - 4 балла, оценка снижается на 1 балл за каждую ошибку Соответствует выбранному процессу - 3 балла, оценка снижается на 1 балл за каждую ошибку	дифференцированный зачет
10	8	Текущий контроль	Задание 10	1	10	Разработан API для задачи (не менее 4 услуг) — 4 балла (1 балл за каждую услугу, но не более 4 баллов) API описан с помощью WDSL/OpenAPI – 4	дифференцированный зачет

						балла, оценка снижается на 1 балл за каждую ошибку в описании Выполнено тестирование API – 2 балла, иначе 0 баллов	
11	8	Проме-жуточная аттестация	дифференцированный зачет	-	10	Это контрольное мероприятие проводится в форме собеседования. Задаются два вопроса по пройденным темам. В первую очередь предлагаются вопросы по темам, которые были оценены на "неудовлетворительно" по текущему контролю. Каждый ответ оценивается от 0 до 5 баллов в зависимости от полноты ответа, знания терминов. Шкала оценивания Полный, правильный ответ - 5 баллов Одна неточность, неправильный термин - 4 балла Частичный ответ - 3 балла В ответе есть некоторые правильные определения - 2 балла Нет ответа - 0 баллов Оценка ставится как сумма баллов за оба ответа.	дифференцированный зачет
12	8	Бонус	Бонус-рейтинг	-	8	Активность на занятиях, посещаемость 100% посещение (допускаются пропуски по уважительной причине) - 3 балла 85-99% посещение - 2 балла Работа у доски 1 балл за решение задачи у доски, но не более 5 баллов за семестр	дифференцированный зачет

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
дифференцированный зачет	Возможно определение рейтинга студента по дисциплине по результатам текущего контроля в соответствии с п.2.6. Если студент не набрал необходимый рейтинг по текущему контролю, то проводится устное собеседование. Студент должен ответить на 2 вопроса из вопросов к дифференцированному зачету. В первую очередь предлагаются вопросы по темам, которые были оценены на "неудовлетворительно" по текущему контролю.	В соответствии с пп. 2.5, 2.6 Положения

6.3. Оценочные материалы

Компетенции	Результаты обучения	№ КМ											
		1	2	3	4	5	6	7	8	9	10	11	12
ПК-1	Умеет: решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности	+	+	+	+	+						+	+
ПК-1	Имеет практический опыт: использования решений практических задач на основе математических и естественнонаучных подходов		+	+	+	+	+	+	+	+		+	+
ПК-3	Умеет: использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта	+		+			+	+	+	+	+	+	+
ПК-3	Имеет практический опыт: применения методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного продукта	+		+				+	+	+	+	+	

Фонды оценочных средств по каждому контрольному мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) основная литература:

Не предусмотрена

б) дополнительная литература:

Не предусмотрена

в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:

1. Программирование : науч. журн. /Рос. акад. наук, Отд-ние информатики, вычисл. техники и автоматизации, Моск. гос. ун-т

г) методические указания для студентов по освоению дисциплины:

1. Методические указания для СРС по выполнению лабораторных работ (на сервере кафедры Н:\Учебные материалы)

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Методические указания для СРС по выполнению лабораторных работ (на сервере кафедры Н:\Учебные материалы)

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная литература	Электронно-библиотечная система издательства Лань	Новиков, Ф.А. Учебно-методическое пособие по дисциплине «Анализ и проектирование на UML». [Электронный ресурс] — Электрон. дан. — СПб. : НИУ ИТМО, 2007. — 286 с. http://e.lanbook.com/book/43540
2	Дополнительная литература	Электронно-библиотечная система издательства Лань	Буч, Г. Язык UML. Руководство пользователя. [Электронный ресурс] / Г. Буч, Д. Рамбо, И. Якобсон. — Электрон. дан. — М. : ДМК Пресс, 2008. — 496 с. http://e.lanbook.com/book/1246
3	Основная литература	Электронно-библиотечная система издательства Лань	Косяков, А. Системная инженерия. Принципы и практика. [Электронный ресурс] / А. Косяков, У. Свит. — Электрон. дан. — М. : ДМК Пресс, 2014. — 624 с. http://e.lanbook.com/book/66484

Перечень используемого программного обеспечения:

1. -WhiteStarUML (инструмент работы с диаграммами UML)(бессрочно)
2. Microsoft-Office(бессрочно)
3. Microsoft-Visio(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Лабораторные занятия	332 (3б)	Компьютеры, редактор для UML диаграмм (UMLet, WhiteStar UML или MS Visio), MS Office