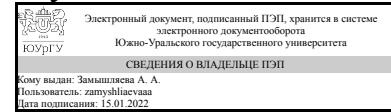


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Директор института
Институт естественных и точных
наук



А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.Ф.П1.13 Современные технологии разработки программного
обеспечения

для направления 02.03.01 Математика и компьютерные науки

уровень Бакалавриат

профиль подготовки Компьютерное моделирование в инженерном и
технологическом проектировании

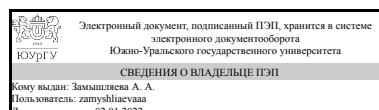
форма обучения очная

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

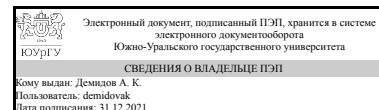
Зав.кафедрой разработчика,
д.физ.-мат.н., проф.

А. А. Замышляева



Разработчик программы,
доцент

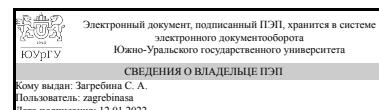
А. К. Демидов



СОГЛАСОВАНО

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

С. А. Загребина



Челябинск

1. Цели и задачи дисциплины

Цели преподавания дисциплины: познакомить бакалавров с основными особенностями и проблемами разработки программного обеспечения; сформировать представление о современных тенденциях развития разработки ПО; изучить методические основы создания современных программных систем; изучить требования предъявляемые к современным технологиям создания программного обеспечения; познакомить с технологиями создания ПО ведущих компаний в области разработки программных продуктов. Задачи изучения дисциплины: - познакомить студентов с современными технологиями разработки программных обеспечения; - изучить методы проектирования и производства программ, способы описания их структуры; - получить опыт работы с инструментальными средствами, поддерживающими создание программного продукта; - подготовить к командной работе над программными продуктами.

Краткое содержание дисциплины

Модели жизненного цикла программного обеспечения и методы управления разработкой. Современные технологии разработки программ: подготовка технического задания, проектирование, кодирование, тестирование, сопровождение (эволюция).

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ПК-2 Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий	Знает: основные технологии разработки программного обеспечения Умеет: работать с основными технологиями разработки программного обеспечения Имеет практический опыт: использования основных технологий разработки программного обеспечения
ПК-3 Способен создавать и исследовать математические модели в естественных науках и промышленности, с учетом возможностей современных информационных технологий и программирования и компьютерной техники	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: 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
Офисные приложения и технологии, Практикум по интерактивным графическим системам, Практикум по основам компьютерного моделирования, Основы компьютерного моделирования	Применение системы ANSYS к решению инженерных задач, Основы проектирования человека-машинного интерфейса, Дискретная оптимизация, Методы и средства научной визуализации, Нейроматематика, Применение системы ANSYS к моделированию физических процессов, Теория оптимизации, Web-программирование, Функциональное и логическое программирование, Параллельные и распределенные вычисления, Высокопроизводительные вычисления на графических ускорителях, Математика в современном естествознании, Анализ требований и проектирование ПО, Анализ и обработка больших массивов данных, Введение в компьютерный анализ и интерпретация данных, Программирование для мобильных устройств, Искусственный интеллект и нейронные сети, САПР технологических процессов, Вычислительная геометрия в инженерном проектировании, Математическое моделирование физических и технических процессов, Программирование на языке Java, Производственная практика, преддипломная практика (8 семестр), Производственная практика, научно-исследовательская работа (6 семестр)

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

Дисциплина	Требования
Практикум по интерактивным графическим системам	Знает: Умеет: математических и естественных наук, программирования и информационных технологий Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий
Офисные приложения и технологии	Знает: основные методы использования

	информационных технологий Умеет: работать с современными информационными технологиями Имеет практический опыт: использования современных информационных технологий
Практикум по основам компьютерного моделирования	Знает: Умеет: применять базовые методы математических и естественных наук, программирования и информационных технологий Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий
Основы компьютерного моделирования	Знает: базовые методы математических и естественных наук, программирования и информационных технологий Умеет: применять базовые методы математических и естественных наук, программирования и информационных технологий Имеет практический опыт: использования базовых методов математических и естественных наук, программирования и информационных технологий

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 3 з.е., 108 ч., 54,25 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах	
		Номер семестра	5
Общая трудоёмкость дисциплины	108	108	
<i>Аудиторные занятия:</i>			
Лекции (Л)	24	24	
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	0	0	
Лабораторные работы (ЛР)	24	24	
<i>Самостоятельная работа (CPC)</i>	53,75	53,75	
с применением дистанционных образовательных технологий	0		
Подготовка к выполнению лабораторных работ	40	40	
Подготовка к зачету	13,75	13.75	
Консультации и промежуточная аттестация	6,25	6,25	
Вид контроля (зачет, диф.зачет, экзамен)	-		зачет

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Жизненный цикл и этапы разработки ПО	42	20	0	22
2	Управление разработкой ПО	6	4	0	2

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Программная инженерия. Проблемы разработки ПО. Стадии и процессы жизненного цикла ПО. Модели ЖЦ ПО.	2
2	1	Планирование и определение системы. Термины и методики (интервью, мозговой штурм), разработка спецификаций, управление масштабом.	2
3	1	Проектирование архитектуры системы. Структура системы, модели управления, виды декомпозиции, архитектуры распределенных систем.	2
4	1	Объектно-ориентированное проектирование. Диаграммы UML. Выявление классов и выбор операций	2
5	1	Принципы проектирования интерфейсов пользователя, способы взаимодействия, представление информации, материалы для обучения.	2
6	1	Реализация. Понятность программы: комментарии, отступы, имена. Выбор языка, повторное использование, оптимизация.	2
7	1	Разработка через тестирование. Рефакторинг.	2
8	1	Принципы тестирования. Восходящее и нисходящее тестирование. Тестирование модуля как черного и белого ящика.	2
9	1	Системное тестирование. Отладка (задача, методы, принципы).	2
10	1	Эксплуатация и сопровождение	2
11	2	Классические и гибкие методы управление разработкой ПО. Бригада главного программиста. Экстремальное программирование. Scrum. RUP.	2
12	2	CASE-средства. Классификация, примеры CASE-средств и их назначение (IDE, VCS и др.)	2

5.2. Практические занятия, семинары

Не предусмотрены

5.3. Лабораторные работы

№ занятия	№ раздела	Наименование или краткое содержание лабораторной работы	Кол-во часов
1	1	Проведение интервью с заказчиком	2
2	1	Проведение мозгового штурма для определения функций ПО	2
3	1	Подведение итогов мозгового штурма и выбор функций для 1-й версии	2
4	1	Написание спецификации для функциональных и нефункциональных требований	2
5	1	Проектирование архитектуры системы с использованием UML	2
6	1	Разработка диаграммы классов UML для объектно-ориентированной системы	2
7	1	Разработка диаграммы состояний UML для объектно-ориентированной системы	2
8	1	Разработка через тестирование	2
9	1	Тестирование модуля как белого ящика	2
10	1	Тестирование модуля как чёрного ящика и тестирование системы	2
11	1	Использование библиотек для логирования, локализация ошибки с помощью отладчика в IDE	2

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Подготовка к выполнению лабораторных работ	ЭУМД,осн.лит.2.с.10-415, ЭУМД,осн.лит.3.с. 15-160, ЭУМД,доп.лит.4.с. 41-190	5	40
Подготовка к зачету	ЭУМД,осн.лит.2.с.10-415, ЭУМД,осн.лит.3.с. 15-160, ЭУМД,доп.лит.1.с. 5-221, ЭУМД,доп.лит.4.с. 41-190	5	13,75

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-мestr	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учи-тыва-ется в ПА
1	5	Текущий контроль	ЛР 1. Проведение интервью с заказчиком	1	10	1. Степень погружения в предметную область: от 0 до 3 баллов в зависимости от достоверность описания выбранной предметной области. 2. Выявлены технологические потребности/проблемы от 0 до 3 баллов, оценивается по заключению аналитика, по 1 баллу за каждую проблему, связанную с разработкой ПО. 3. Есть краткая запись результатов интервью в соответствии с планом: 4 балла, если есть, 0, если нет, 2 при ошибках, неполном выполнении плана интервью.	зачет
2	5	Текущий контроль	ЛР 2. Мозговой штурм	1	10	Участвовал в мозговом штурме: 3 балла, если да, 0, если задание по генерации было выполнено вне группы. Предложил не менее 5 функций: 2 балла, если есть, 0, если менее 5 функций Предложенные функции связаны с компьютерными технологиями и ПО:	зачет

						по 1 баллу за каждую функцию, но не более 5 баллов	
3	5	Текущий контроль	ЛР 3. Подведение итогов мозгового штурма и выбор функций для 1-й версии	1	10	<p>Выполнено отсечение, уточнение формулировки - 2 балла, должно остаться не более 20-25 функций и отброшены нереалистичные функции, оценка снижается на 1 балл за каждую ошибку.</p> <p>Каждый участник выполнил оценку важности функций - 2 балла, баллы не начисляются участникам, которые не участвовали в оценке важности функций</p> <p>Выполнена реалистичная оценка трудоемкости - 2 балла, оценка снижается на 1 балл за каждую некорректную оценку более чем в 2 раза</p> <p>Выполнена реалистичная оценка рисков - 2 балла, оценка снижается на 1 балл за каждую некорректную оценку более чем в 2 раза</p> <p>Выполнена сортировка и выбраны функции для 1-й версии - 2 балла, или 0 баллов, если сортировка не выполнена, или суммарная трудоемкость существенно отличается от времени на разработку 1 версии (2-3 недели)</p>	зачет
4	5	Текущий контроль	ЛР 4. Написание спецификации для функциональных и нефункциональных требований	1	10	<p>Написана спецификация для функционального требования - 2 балла или 0 баллов, если отсутствует</p> <p>Написана спецификация для нефункционального требования - 2 балла или 0 баллов, если отсутствует</p> <p>Спецификация является полной - 2 балла или 0 баллов, если есть существенная необходимость в уточнениях для начала разработки</p> <p>Спецификация является верифицируемой - 2 балла, оценка снижается на 1 балл за каждый пункт спецификации, который невозможно проверить</p> <p>Спецификация является понимаемой и недвусмысленной - 2 балла, оценка снижается на 1 балл за каждый пункт спецификации с неправильным термином или ошибкой в формулировке.</p>	зачет
5	5	Текущий контроль	ЛР 5. Проектирование архитектуры системы с использованием UML	1	10	<p>Диаграмма нарисована с помощью средств рисования диаграмм UML - 2 балла, или 0 баллов, если использованы не соответствующие инструменты</p> <p>Выбраны правильные обозначения</p>	зачет

						UML для всех элементов диаграммы - 4 балла, оценка снижается на 1 балл за каждую ошибку Диаграмма является полной - 2 балла, или 0 баллов, если важный компонент был пропущен Выбраны компоненты, соответствующие предметной области- 2 балла, оценка снижается на 1 балл за каждую ошибку	
6	5	Текущий контроль	ЛР 6. Разработка диаграммы классов UML для объектно-ориентированной системы	1	10	Диаграмма нарисована с помощью средств рисования диаграмм UML - 2 балла, или 0 баллов, если использованы не соответствующие инструменты Выбраны правильные обозначения UML для всех элементов диаграммы и связей между ними - 3 балла, оценка снижается на 1 балл за каждую ошибку Правильно указаны поля классов, их типы и методы - 3 балла, оценка снижается на 1 балл за каждую ошибку Диаграмма является полной - 2 балла, или 0 баллов, если важный элемент, описанный в задании, был пропущен	зачет
7	5	Текущий контроль	Разработка диаграммы состояний/диаграммы активности UML для объектно-ориентированной системы	1	10	Диаграмма нарисована с помощью средств рисования диаграмм UML - 2 балла, или 0 баллов, если использованы не соответствующие инструменты Выбраны правильные обозначения UML для всех элементов диаграммы и связей между ними - 3 балла, оценка снижается на 1 балл за каждую ошибку Правильно указаны события и условия - 3 балла, оценка снижается на 1 балл за каждую ошибку Диаграмма является полной - 2 балла, или 0 баллов, если важный элемент, описанный в задании, был пропущен	зачет
8	5	Текущий контроль	ЛР 8. Разработка через тестирование	1	10	Представлена история модификаций кода - 2 балла, иначе 0 баллов. Предложено не менее 6 тестов - 4 балла, если менее 6, то оценка снижается на (6-количество тестов) баллов Выполнен рефакторинг кода после добавления тестов - 2 балла, иначе 0 баллов Набор тестов полный - 2 балла, иначе 0 баллов.	зачет
9	5	Текущий контроль	ЛР 9. Тестирование модуля как белого	1	10	Выполнено знакомство со средствами проверки покрытия кода - 4 балла,	зачет

			ящика			иначе 0 баллов Выполнено покрытие операторов на 100% - 4 балла, иначе 0 баллов Выполнено комбинаторное покрытие условий на 100% - 2 балла, иначе 0 баллов	
10	5	Текущий контроль	ЛР 10. Тестирование модуля как чёрного ящика и тестирование системы	1	10	Выполнена разработка тестов для подзадачи 1, определяющих все ошибки в наборе программ - 4 балла, иначе 0 баллов Выполнена разработка тестов для подзадачи 2, определяющих все ошибки в наборе программ - 6 баллов, если в половине программ из набора - 4 балла, иначе 0 баллов	зачет
11	5	Текущий контроль	ЛР 11. Использование библиотек для логирования, локализация ошибки с помощью отладчика в IDE	1	10	Выполнен запуск на тесте в отладчике, продемонстрировано способы пошагового выполнения программы, просмотра текущего - 4 балла, иначе 0 баллов Подключена библиотека для логирования, расставлены операции логирования в программе - 2 балла, иначе 0 баллов Локализована ошибка и указана причина - - 4 балла, иначе 0 баллов	зачет
12	5	Текущий контроль	ЛР 12. Работа с системой контроля версий	1	10	Выполнена регистрация на github и подключение к проекту - 2 балла, иначе 0 баллов Получен начальный код и создана собственная ветка - 2 балла, иначе 0 баллов Написан код модуля и выполнено слияние ветки разработчика с основной веткой- 2 балла, иначе 0 баллов Получена основная ветка с изменениями от всех разработчиков и выполнено её слияние с веткой и проверка работы программы - 2 балла, иначе 0 баллов Внесены исправления и выполнено слияние ветки разработчика с основной веткой - 2 балла, иначе 0 баллов	зачет
13	5	Промежуточная аттестация	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.	зачет

						Оценка ставится как сумма баллов за оба ответа.	
14	5	Бонус	Бонус-рейтинг	-	15	Участие в олимпиадах по программированию Личное первенство ЮУрГУ (очный тур), соревнования командного чемпионата мира по программированию - по 0,5 балла за решенную задачу, но ее более 3 баллов за соревнование Активность на занятиях, посещаемость 100% посещение (допускаются пропуски уважительной причине) - 3 балла 85-99% посещение - 2 балла по 0,1 балла за ответы на практических занятиях	зачет

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
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 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	13	14
ПК-2	Знает: основные технологии разработки программного обеспечения	++++					+++++			+			+		
ПК-2	Умеет: работать с основными технологиями разработки программного обеспечения	++++					+++++			+			+		
ПК-2	Имеет практический опыт: использования основных технологий разработки программного обеспечения	++++					+++++			+			+		
ПК-3	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)									++++++		+	+	+	+
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

a) 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. Презентации для лекций

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Презентации для лекций

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Дополнительная литература	Электронно-библиотечная система издательства Лань	Ехлаков, Ю. П. Управление программными проектами. Стандарты, модели : учебное пособие для вузов / Ю. П. Ехлаков. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 244 с. — ISBN 978-5-8114-8362-4. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/175498
2	Основная литература	Электронно-библиотечная система издательства Лань	Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие / Т. М. Зубкова. — Оренбург : ОГУ, 2017. — 468 с. — ISBN 978-5-7410-1785-2. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/110632
3	Основная литература	Электронно-библиотечная система издательства Лань	Иванов, Д. Моделирование на UML / Д. Иванов, Ф. Новиков. — Санкт-Петербург : НИУ ИТМО, 2010. — 200 с. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/40879
4	Дополнительная литература	Электронно-библиотечная система	Котляров, В. П. Основы тестирования программного обеспечения : учебное пособие / В. П. Котляров. — 2-е изд. — Москва : ИНТУИТ, 2016. — 248 с. — ISBN 5-9556-0027-

		издательства Лань	2. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/100352
--	--	-------------------	---

Перечень используемого программного обеспечения:

1. -WhiteStarUML (инструмент работы с диаграммами UML)(бессрочно)
2. The Git Development Community-Git(бессрочно)
3. Microsoft-Office(бессрочно)
4. -MinIDE (сборка из SciTE, MinGW C/C++, GDB)(бессрочно)
5. -Microsoft Visual Studio (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
Лекции	336 (36)	Проектор
Практические занятия и семинары	332 (36)	Компьютеры с Visual Studio, UML-редактором, офисным ПО