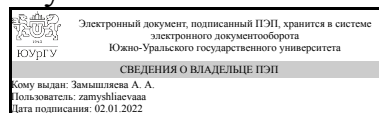


# ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:  
Директор института  
Институт естественных и точных  
наук



А. А. Замышляева

## РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.28 Современные технологии разработки программного обеспечения

для направления 01.03.02 Прикладная математика и информатика

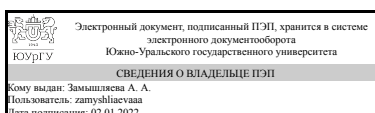
уровень Бакалавриат

форма обучения очная

кафедра-разработчик Прикладная математика и программирование

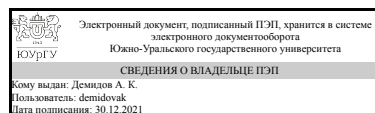
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.02 Прикладная математика и информатика, утверждённым приказом Минобрнауки от 10.01.2018 № 9

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



А. А. Замышляева

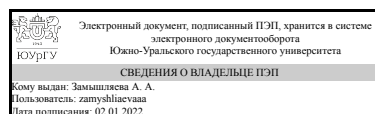
Разработчик программы,  
доцент



А. К. Демидов

СОГЛАСОВАНО

Руководитель направления  
д.физ.-мат.н., проф.



А. А. Замышляева

## 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 комплексов; - познакомить современными подходами к выполнению основных технологических операций; - подготовить к командной работе над программными комплексами.

## Краткое содержание дисциплины

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

## 2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	Знает: требования предъявляемые к современным технологиям создания программного обеспечения Имеет практический опыт: адаптации процесса разработки ПО к требованиям информационной безопасности
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	Знает: современные технологии разработки программного обеспечения Имеет практический опыт: разработки алгоритмов и компьютерных программ с использованием современных компьютерных технологий

## 3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
1.О.24 Объектно-ориентированное программирование, 1.О.15 Архитектура ЭВМ, Производственная практика, технологическая (проектно-технологическая) практика (4 семестр), Учебная практика, технологическая (проектно-технологическая) практика (2 семестр)	1.О.35 Администрирование и проектирование хранилищ данных, 1.О.29 Функциональное и логическое программирование

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

Дисциплина	Требования
1.О.24 Объектно-ориентированное программирование	<p>Знает: синтаксис языка объектно-ориентированного программирования С++; устройство и принципы построения объектно-ориентированных библиотек, методику разработки программ с использованием технологии объектно-ориентированного программирования Умеет: адаптировать и использовать шаблоны объектно-ориентированного программирования для решения профессиональных задач, реализовывать программно и использовать на практике математические алгоритмы, с применением высокоуровневого языка программирования С++ Имеет практический опыт: применения объектных технологий разработки программных систем, разработки компьютерных программ на языке С++</p>
1.О.15 Архитектура ЭВМ	<p>Знает: системные принципы функционирования компьютерных систем, достаточные для успешной деятельности в области разработки программного обеспечения и компьютерного моделирования Умеет: выбрать архитектуру вычислительной системы, адекватную решаемым задачам, с учётом основных требований информационной безопасности Имеет практический опыт:</p>
Производственная практика, технологическая (проектно-технологическая) практика (4 семестр)	<p>Знает: эффективные стратегии командного сотрудничества для достижения поставленной цели, причины, признаки и последствия опасностей, способы защиты от чрезвычайных ситуаций Умеет: самостоятельно изучать новые технологии, используемые на предприятии, с помощью информационно-коммуникационных систем, нести личную ответственность за результат, идентифицировать опасные и вредные факторы в рамках осуществляемой деятельности, оценить потребность в ресурсах и планировать их использование при решении задач профессиональной деятельности Имеет практический опыт: участия в разработке научно-исследовательского проекта, применяя изученные технологии, применения полученных математических знаний и навыков программирования для решения прикладных задач, работы в направлении личностного, образовательного и профессионального роста, решения поставленных задач, с учётом имеющихся ресурсов и ограничений</p>
Учебная практика, технологическая (проектно-технологическая) практика (2 семестр)	<p>Знает: этические нормы и установленные правила командной работы, способы первичной</p>

	<p>обработки информации Умеет: использовать математический аппарат в решении профессиональных задач, находить и критически анализировать информацию, необходимую для решения поставленной задачи, разрабатывать математические модели, алгоритмы и компьютерные программы для предложенных задач, критически оценить эффективность использования времени при решении поставленных задач, а также, относительно полученного результата Имеет практический опыт: программной реализации алгоритмов задач профессиональной деятельности, участия в обмене информацией, знаниями и опытом в интересах выполнения командной задачи, декомпозиции поставленной задачи, выделяя её базовые составляющие, оценки личностных ресурсов по достижению целей управления своим временем для успешного выполнения порученной работы и саморазвития</p>
--	--

#### 4. Объём и виды учебной работы

Общая трудоёмкость 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>	48	48	
Лекции (Л)	24	24	
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	0	0	
Лабораторные работы (ЛР)	24	24	
<i>Самостоятельная работа (СРС)</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

12	2	Работа с системой контроля версий	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 или ПУМД,доп.лит.,с. 45-317, 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. 5-120, 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	зачет

						компьютерными технологиями и ПО: 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 функций, 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>	зачет

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



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

						ответа, знания терминов. 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 не 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
ОПК-4	Знает: требования предъявляемые к современным технологиям создания программного обеспечения	+	+	+	+	+	+	+	+	+	+	+	+	+	
ОПК-4	Имеет практический опыт: адаптации процесса разработки ПО к требованиям информационной безопасности			+	+	+	+			+	+		+	+	
ОПК-5	Знает: современные технологии разработки программного обеспечения	+	+	+	+	+	+	+	+	+	+	+	+	+	
ОПК-5	Имеет практический опыт: разработки алгоритмов и компьютерных программ с использованием современных компьютерных технологий			+	+	+	+	+	+		+	+	+	+	

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:

1. Гагарина, Л. Г. Технология разработки программного обеспечения [Текст] учеб. пособие для вузов по направлению 230100 "Информатика и вычисл. техника" специальности 230105 "Програм. обеспечение вычисл. техники и автоматизир. систем" Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул ; под ред. Л. Г. Гагариной. - М.: Форум : ИНФРА-М, 2012. - 399 с. ил. 22 см.

### 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	Шопырин, Д. Г. Управление проектами разработки ПО. Дисциплина «Гибкие технологии разработки программного обеспечения» / Д. Г. Шопырин. — Санкт-Петербург : НИУ ИТМО, 2007. — 131 с. — Текст : электронный // Лань : электронно-библиотечная система. <a href="https://e.lanbook.com/book/43554">https://e.lanbook.com/book/43554</a>
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. <a href="https://e.lanbook.com/book/110632">https://e.lanbook.com/book/110632</a>
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. <a href="https://e.lanbook.com/book/40879">https://e.lanbook.com/book/40879</a>
4	Дополнительная литература	Электронно-библиотечная система издательства Лань	Котляров, В. П. Основы тестирования программного обеспечения : учебное пособие / В. П. Котляров. — 2-е изд. — Москва : ИНТУИТ, 2016. — 248 с. — ISBN 5-9556-0027-2. — Текст : электронный // Лань : электронно-библиотечная система. <a href="https://e.lanbook.com/book/100352">https://e.lanbook.com/book/100352</a>

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
Практические занятия и семинары	332 (36)	Компьютеры с Visual Studio, UML-редактором, офисным ПО
Лекции	336 (36)	Проектор