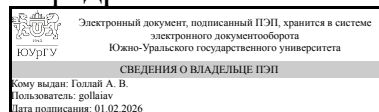


# ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:  
Заведующий выпускающей  
кафедрой



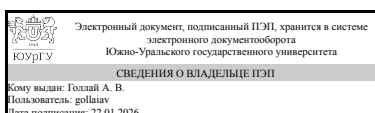
А. В. Голлай

## РАБОЧАЯ ПРОГРАММА

**дисциплины 1.Ф.П0.15** Разработка мобильных приложений  
**для направления 09.03.01** Информатика и вычислительная техника  
**уровень** Бакалавриат  
**профиль подготовки** ИТ-инженерия  
**форма обучения** очная  
**кафедра-разработчик** Центр подготовки топ-специалистов в сфере ИТ "Цифровой Урал"

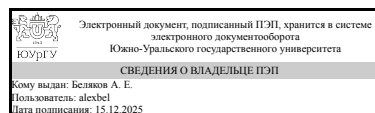
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённым приказом Минобрнауки от 19.09.2017 № 929

Зав.кафедрой разработчика,  
д.техн.н., доц.



А. В. Голлай

Разработчик программы,  
старший преподаватель



А. Е. Беляков

## 1. Цели и задачи дисциплины

Цель дисциплины: Формирование у студентов систематизированных знаний, практических умений и навыков в области проектирования, разработки и тестирования нативных мобильных приложений для операционной системы Android с использованием современного языка программирования Kotlin и актуальных инструментов разработки. Задачи дисциплины: 1. Изучить основы платформы Android, компоненты приложения и жизненные циклы. 2. Освоить синтаксис, идиомы и ключевые возможности языка Kotlin (null-safety, корутины, расширения функций, DSL). 3. Сформировать навыки проектирования пользовательского интерфейса (UI) с помощью Jetpack Compose и понимания принципов Material Design. 4. Научиться управлять данными в приложении, используя архитектурные паттерны (MVVM), работу с локальными базами данных (Room) и сетевыми запросами (Retrofit, Coroutines/Flow). 5. Привить навыки отладки, тестирования (Unit, UI тесты) и публикации приложения.

## Краткое содержание дисциплины

Дисциплина охватывает полный цикл создания современного Android-приложения на Kotlin. Студенты начинают с изучения основ Kotlin и Android-экосистемы, затем переходят к построению декларативного UI с помощью Jetpack Compose. Особое внимание уделяется архитектуре приложения, асинхронной работе, управлению состоянием и работе с данными (локальное хранение, REST API). Курс завершается изучением модульного и UI-тестирования, а также процесса подготовки приложения к публикации.

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

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ПК-1 Способен разрабатывать и отлаживать программный код	Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования. Умеет: применять выбранные языки программирования для написания программного кода мобильных приложений, использовать выбранную среду программирования, возможности имеющейся технической и/или программной архитектуры для написания программного кода мобильных приложений Имеет практический опыт: разработки программного кода модулей мобильных приложений с использованием языков программирования

## 3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Информатика, Алгоритмы и структуры данных, Объектно-ориентированное программирование, Математическая логика и теория алгоритмов, Программирование на языках высокого уровня, Основы программирования на Python	Микросервисная архитектура и основы DevOps, Вычислительные методы в ИТ

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

Дисциплина	Требования
Информатика	<p>Знает: принципы работы современных информационных технологий; принципы работы программных средств, базовые понятия информатики и информационных технологий, основные форматы представления информации для автоматизированной обработки; основные принципы работы вычислительных систем и их компонентов; ОПК-2.1. 3-3. Знает основные принципы решения задач с помощью компьютера, понятие алгоритма, основные алгоритмические структуры; состав, назначение функциональных компонентов и программного обеспечения персонального компьютера; методы и приемы алгоритмизации поставленных задач; нотации и программное обеспечение для графического отображения алгоритмов</p> <p>Умеет: использовать современные информационные технологий при решении задач профессиональной деятельности; использовать программные средства при решении задач профессиональной деятельности, работать с информацией, представленной в различных формах; разрабатывать алгоритмы для решения типовых задач; работать с персональным компьютером, применять современное ПО, в т.ч., отечественного производства, для решения практических задач, использовать методы и приемы формализации и алгоритмизации поставленных задач</p> <p>Имеет практический опыт: навыками использования современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности, разработки алгоритмов для решения типовых задач</p>
Объектно-ориентированное программирование	<p>Знает: основные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования, возможности компиляторов программных проектов под</p>

различные операционные системы, наборы инструкций для системных утилит автоматической сборки программного обеспечения и установки программных пакетов объектно-ориентированных библиотек и фреймворков, принципы чистого кода, SOLID, DRY, KISS и др., принципы предметно-ориентированного проектирования (ПОП) программного обеспечения, паттерны проектирования и антипаттерны, методы и средства проверки работоспособности компьютерного программного обеспечения, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования; возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями

Умеет: использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах, разрабатывать модульный и тестируемый программный код, выполнять модульное, интеграционное и нагрузочное тестирование, проводить рефакторинг для повышения качества кода, применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах, применять методы и средства проверки работоспособности компьютерного программного обеспечения, применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода; регистрировать изменения исходного текста программного кода в системе управления версиями; сохранять изменения программного кода в соответствии с регламентом управления версиями; выполнять слияние, разделение и сравнение исходных текстов программного кода

Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на объектно-ориентированных языках, разработки,

	отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux
Математическая логика и теория алгоритмов	<p>Знает: основные понятия и законы математической логики и теории алгоритмов, необходимые для построения моделей профессиональных задач, алгоритмы решения типичных задач, области и способы их применения; методы и приемы алгоритмизации поставленных задач</p> <p>Умеет: формализовать профессиональные задачи средствами математической логики и строить модели процессов с использованием методов теории алгоритмов, использовать методы и приемы формализации и алгоритмизации поставленных задач</p> <p>Имеет практический опыт: разработки и проверки правильности логико-математических конструкций и алгоритмов, используемых в расчетах и проектировании</p>
Основы программирования на Python	<p>Знает: особенности синтаксиса языка Python и базовых библиотек для работы с информацией (NumPy, Pandas, Matplotlib), основы программирования на Python</p> <p>Умеет: создавать эффективные программы на Python для решения простых и сложных задач обработки данных, применять выбранные языки программирования для написания программного кода</p> <p>Имеет практический опыт: применения технологий написания читаемого и поддерживаемого кода на Python</p>
Алгоритмы и структуры данных	<p>Знает: фундаментальных свойств алгоритмов и структур данных; методы решения алгоритмических задачи с применением структур данных в соответствии с особенностями предметной области, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования</p> <p>Умеет: проектировать алгоритмическое решение на основе выбранной структуры данных; использовать средства разработки и отладки современной интегрированной среды программирования, применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; возможности имеющейся технической и/или программной архитектуры для написания программного кода</p> <p>Имеет практический опыт: применения широкого набора приемов, методов и технологий программирования различных задач, разработки программного кода с использованием языков программирования, определения и</p>

	манипулирования данными, применяя выбранную систему контроля версий и инструментальные программные средства
Программирование на языках высокого уровня	<p>Знает: основные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня, возможности компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования; возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями</p> <p>Умеет: использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах., применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода; регистрировать изменения исходного текста программного кода в системе управления версиями; сохранять изменения программного кода в соответствии с регламентом управления версиями; выполнять слияние, разделение и сравнение исходных текстов программного кода</p> <p>Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux</p>

#### 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
		4
Общая трудоёмкость дисциплины	72	72
<i>Аудиторные занятия:</i>	48	48
Лекции (Л)	16	16
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	32	32
Лабораторные работы (ЛР)	0	0
<i>Самостоятельная работа (СРС)</i>	19,75	19,75
Разработка компьютерных программ	15	15
Подготовка к зачету	4,75	4,75
Консультации и промежуточная аттестация	4,25	4,25
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	Введение в Kotlin и Android-экосистему	8	2	6	0
2	Основы языка Kotlin для Android	8	2	6	0
3	Пользовательский интерфейс с Jetpack Compose	10	4	6	0
4	Навигация и архитектура приложения (MVVM)	10	4	6	0
5	Работа с данными (Локальные и сетевые)	8	2	6	0
6	Продвинутые темы и финальная подготовка	4	2	2	0

### 5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Введение. Обзор платформы Android. Настройка среды разработки (Android Studio, SDK, эмулятор)	1
2	1	Установка и конфигурация Android Studio. Первое приложение "Hello World". Обзор структуры проекта	1
3	2	Kotlin: базовый синтаксис, типы данных, null-safety, функции, ООП (классы, объекты, data-классы)	1
4	2	Решение задач на Kotlin: работа с коллекциями, стандартные функции-расширения (let, apply, run)	1
5	3	Основы Jetpack Compose. Композируемые функции, модификаторы, стандартные компоненты (Column, Row, Box, Text, Button). Состояние в Compose (mutableStateOf)	2
6	3	Создание экрана приложения с использованием базовых composables и управления состоянием	2
7	4	Архитектурный компонент Navigation. Паттерн Model-View-ViewModel (MVVM). ViewModel и LiveData/StateFlow	2
8	4	Реализация многоэкранного приложения с навигацией и передачей данных между экранами с использованием ViewModel	2
9	5	Локальная база данных с Room Persistence Library. Работа с сетью:	1

		библиотека Retrofit, обработка JSON, использование корутин или Flow	
10	5	Создание репозитория для загрузки данных из REST API и сохранения их в локальную БД	1
11	6	Dependency Injection (Hilt/Dagger). Основы модульного и инструментального тестирования. Подготовка к публикации в Google Play	1
12	6	Написание Unit-тестов для ViewModel. Сборка release-APK	1

## 5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Каркас первого приложения	6
2	2	Логика на Kotlin	6
3	3	Экран списка задач с Compose	6
4	4	Архитектура и навигация	6
5	5	Работа с API и БД	6
6	6	Тестирование ПО	2

## 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
Разработка компьютерных программ	Слайды лекций, методические указания к заданиям	4	15
Подготовка к зачету	Слайды лекций, методические указания к заданиям	4	4,75

## 6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

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	4	Текущий контроль	ЛР №1	20	20	Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE.	зачет

						<p>Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются.</p> <p>Критерии:</p> <ol style="list-style-type: none"> <li>Наличие компилируемого без ошибок программного кода задания - 5 балла.</li> <li>Понимание и комментирование функциональных блоков кода студентом - 5 баллов.</li> <li>Ответы на вопросы по заданию - 10 балла.</li> </ol>	
2	4	Текущий контроль	ЛР №2	20	20	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду.</p> <p>Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE.</p> <p>Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются.</p> <p>Критерии:</p> <ol style="list-style-type: none"> <li>Наличие компилируемого без ошибок программного кода задания - 5 балла.</li> <li>Понимание и комментирование функциональных блоков кода студентом - 5 баллов.</li> <li>Ответы на вопросы по заданию - 10 балла.</li> </ol>	зачет
3	4	Текущий контроль	ЛР №3	20	20	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду.</p> <p>Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE.</p> <p>Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются.</p> <p>Критерии:</p> <ol style="list-style-type: none"> <li>Наличие компилируемого без ошибок программного кода задания - 5 балла.</li> <li>Понимание и комментирование функциональных блоков кода студентом - 5 баллов.</li> <li>Ответы на вопросы по заданию - 10 балла.</li> </ol>	зачет
4	4	Текущий контроль	ЛР №4	20	20	<p>Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду.</p> <p>Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE.</p> <p>Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются.</p> <p>Критерии:</p> <ol style="list-style-type: none"> <li>Наличие компилируемого без ошибок</li> </ol>	зачет

						программного кода задания - 5 балла. 2. Понимание и комментирование функциональных блоков кода студентом - 5 баллов. 3. Ответы на вопросы по заданию - 10 балла.	
5	4	Текущий контроль	ЛР №5	20	20	Баллы начисляются по результатам демонстрации программы и ответов на вопросы по исходному коду. Шкала начисления дискретная, оформлена в виде набора критериев "рубрика" системы MOODLE. Баллы за каждый критерий либо засчитываются в итоговую сумму баллов полностью, либо обнуляются. Критерии: 1. Наличие компилируемого без ошибок программного кода задания - 5 балла. 2. Понимание и комментирование функциональных блоков кода студентом - 5 баллов. 3. Ответы на вопросы по заданию - 10 балла.	зачет
6	4	Промежуточная аттестация	Зачет	-	100	По результатам ответов на вопросы устного зачета. Первый теоретический вопрос - максимум 30 баллов, второй теоретический вопрос - максимум 30 баллов, третий теоретический вопрос - максимум 40 баллов. За успешное выполнение ставится максимум баллов по заданию, если задание выполнено частично или с ошибками, то ставится от 0 до максимума по заданию пропорционально выполненному объему. Длительность зачета - 2 академических часа.	зачет

## 6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
зачет	<p>При оценивании результатов учебной деятельности обучающегося по дисциплине используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (Положение о БРС утверждено приказом ректора от 24.05.2019 г. № 179, в редакции приказа ректора от 10.03.2022 г. № 25-13/09). Оценка за дисциплину формируется на основе полученных оценок за контрольно-рейтинговые мероприятия текущего контроля. Отлично: Величина рейтинга обучающегося по дисциплине 85...100 %.</p> <p>Хорошо: Величина рейтинга обучающегося по дисциплине 75...84 %.</p> <p>Удовлетворительно: Величина рейтинга обучающегося по дисциплине 60...74 %.</p> <p>Неудовлетворительно: Величина рейтинга обучающегося по дисциплине 0...59 %.</p>	В соответствии с пп. 2.5, 2.6 Положения

	<p>Если студент согласен с оценкой, полученной по результатам текущего контроля, то он может в день, предшествующий промежуточной аттестации дать свое согласие в личном кабинете. В случае явки студента на промежуточную аттестацию, давшего свое согласие на автомат в личном кабинете, студент имеет право пройти мероприятия текущего контроля по дисциплине на промежуточной аттестации для улучшения своего рейтинга в день ее проведения. Снижение оценки в этом случае запрещено. Если студент не дал согласия в личном кабинете, то он может согласиться с оценкой лично на промежуточной аттестации в день ее проведения. Если студент не согласен с оценкой, то он имеет право пройти контрольно-рейтинговые мероприятия на промежуточной аттестации для улучшения своего рейтинга в день ее проведения. Фиксация результатов учебной деятельности по дисциплине проводится в день промежуточной аттестации на основе согласия студента, данного им в личном кабинете. При отсутствии согласия в журнале дисциплины фиксация результатов происходит при личном присутствии студента. Если студент не дал согласие в личном кабинете и не явился на промежуточную аттестацию – ему выставляется «неявка».</p>	
--	--	--

### 6.3. Паспорт фонда оценочных средств

Компетенции	Результаты обучения	№ КМ					
		1	2	3	4	5	6
ПК-1	Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования.	+	+	+	+	+	+
ПК-1	Умеет: применять выбранные языки программирования для написания программного кода мобильных приложений, использовать выбранную среду программирования, возможности имеющейся технической и/или программной архитектуры для написания программного кода мобильных приложений	+	+	+	+	+	+
ПК-1	Имеет практический опыт: разработки программного кода модулей мобильных приложений с использованием языков программирования	+	+	+	+	+	+

Типовые контрольные задания по каждому 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:*

Не предусмотрены

г) *методические указания для студентов по освоению дисциплины:*

1. Методические рекомендации по выполнению практических занятий

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Методические рекомендации по выполнению практических занятий

### Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная литература	ЭБС издательства Лань	Петросян, Л. Э. Разработка мобильных приложений на языке Kotlin : учебное пособие для вузов / Л. Э. Петросян, К. В. Гусев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 104 с. — ISBN 978-5-507-52328-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/448577">https://e.lanbook.com/book/448577</a> . — Режим доступа: для авториз. пользователей.
2	Основная литература	ЭБС издательства Лань	Воронцов, Ю. А. Платформы разработки мобильных приложений : учебное пособие / Ю. А. Воронцов, М. А. Овчинников, Е. А. Чернов. — Москва : РТУ МИРЭА, 2023. — 172 с. — ISBN 978-5-7339-1857-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/382436">https://e.lanbook.com/book/382436</a> . — Режим доступа: для авториз. пользователей.
3	Основная литература	ЭБС издательства Лань	Коузен, К. Kotlin. Сборник рецептов / К. Коузен ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2021. — 220 с. — ISBN 978-5-97060-883-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/241007">https://e.lanbook.com/book/241007</a> . — Режим доступа: для авториз. пользователей.

Перечень используемого программного обеспечения:

1. -Microsoft Visual Studio (бессрочно)
2. Docker-Docker(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	809 (36)	Компьютеры, ОС Windows, MS Visual Studio, VS Code, Android Studio, Docker
Лекции	240 (36)	Компьютеры, ОС Windows, MS Visual Studio, VS Code, Android Studio, Docker, доска, проектор

