

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Андрей Драгомирович Хлудков  
Должность: директор  
Дата подписания: 24.06.2026 11:16:51  
Уникальный программный ключ:  
880f7c07c583b07b775f6604c39281b15e9f12

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА и  
ГОСУДАРСТВЕННОЙ СЛУЖБЫ при ПРЕЗИДЕНТЕ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**СЕВЕРО-ЗАПАДНЫЙ ИНСТИТУТ УПРАВЛЕНИЯ**

---

Факультет среднего профессионального образования

УТВЕРЖДЕНА  
решением цикловой (методической)  
комиссии общепрофессиональных  
дисциплин и по профессиональным  
модулям специальности 09.02.07  
Информационные системы и  
программирование  
Протокол от 31.10.2025 № 2

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

### **МДК.03.02 Разработка приложений для мобильных платформ**

Специальность – 09.02.11 Разработка и управление программным обеспечением

Профиль – на базе основного общего образования

Квалификация – программист

Форма обучения – очная

Год набора – 2026

Санкт-Петербург 2025 год

**Автор-составитель:** Вилков Владислав Евгеньевич, преподаватель ФСПО СЗИУ РАНХиГС.

**СОДЕРЖАНИЕ**

1. Общие положения .....	4
1.1. Область применения программы .....	4
1.2. Место дисциплины в структуре основной профессиональной образовательной программы .....	4
1.3. Цели и задачи учебной дисциплины .....	4
1.4. Планируемые результаты обучения по дисциплине .....	4
2. Структура и содержание дисциплины .....	11
2.1. Объем учебной дисциплины и виды работ .....	11
2.2. Тематический план и содержание дисциплины .....	11
2.3. Регламент распределения видов работ по дисциплине с ДОТ .....	15
3. Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся .....	16
3.1. Формы и методы текущего контроля успеваемости обучающихся и промежуточной аттестации.....	16
3.2. Оценочные средства текущего контроля успеваемости обучающихся .....	17
3.3. Оценочные средства промежуточной аттестации обучающихся .....	20
4. Методические указания для обучающихся по освоению дисциплины .....	22
5. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет» .....	22
6. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы .....	25

## **1 Общие положения**

### **1.1 Область применения программы**

Рабочая программа учебной дисциплины «Разработка приложений для мобильных платформ» является частью основной профессиональной образовательной программы по специальности СПО 09.02.11 «Разработка и управление программным обеспечением».

### **1.2 Место дисциплины в структуре основной профессиональной образовательной программы**

Учебная дисциплина «Разработка приложений для мобильных платформ» является частью профессиональной подготовки, входит в общепрофессиональный цикл дисциплин. Базируется на таких дисциплинах, как «Информатика», «Операционные системы и среды», «Основы алгоритмизации и программирования» является основополагающей для такой дисциплины, как «Технологии безопасности мобильных платформ».

Дисциплина изучается на 3 курсе в 6 семестре, 7 семестре.

### **1.3 Цели и задачи учебной дисциплины**

Цель дисциплины «Разработка приложений для мобильных платформ» — формирование у обучающихся комплексных знаний и практических навыков в области разработки мобильных приложений для различных платформ (Android, Aurora, РЕД ОС М), включая освоение современных технологий программирования, работы с данными и публикации приложений в сторах.

Задачи дисциплины:

- Сформировать у обучающихся понимание архитектуры мобильных платформ и особенностей разработки под различные операционные системы (Android, Aurora, РЕД ОС М)
- Развить навыки работы с основными компонентами Android-приложений: Activity, Service, BroadcastReceiver, ContentProvider
- Обучить настройке и конфигурации проектов с использованием Gradle, включая работу с зависимостями и сборкой приложений
- Сформировать компетенции в области работы с данными: файловая система, SharedPreferences, базы данных
- Развить навыки взаимодействия с внешними сервисами через REST API и Retrofit
- Обучить работе с асинхронными операциями и многопоточностью с использованием Kotlin Coroutines и Flow

- Сформировать понимание работы с локальными базами данных через Room Database и DAO
- Развить навыки реализации паттерна репозитория и работы с кэшем
- Обучить работе с фоновыми задачами через WorkManager и AlarmManager
- Сформировать компетенции в области загрузки и обработки медиафайлов
- Развить навыки работы с бесконечными списками и пагинацией (Paging 3)
- Обучить принципам кроссплатформенной разработки и адаптации под разные устройства
- Сформировать понимание процессов публикации приложений в различных магазинах (Google Play, RuStore, Aurora Store)
- Развить навыки работы с CI/CD и автоматизацией сборки
- Обучить интеграции с различными API: карты, геолокация, Bluetooth, NFC
- Сформировать компетенции в области поддержки различных форм-факторов (планшеты, часы, ТВ)
- Развить навыки работы с push-уведомлениями и системами оповещения
- Сформировать понимание современных тенденций в мобильной разработке и готовность к их практическому применению

#### 1.4 Планируемые результаты обучения по дисциплине

##### Перечень компетенций

<b>Код ОК, ПК</b>	<b>Уметь</b>	<b>Знать</b>	<b>Владеть навыками</b>
ОК.01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и	-

		смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	
ОК.02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации, современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств	-
ОК.03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-	-

		планов; порядок выстраивания презентации; кредитные банковские продукты	
ОК.04	Эффективно взаимодействовать и работать в коллективе и команде	психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	-
ОК.05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	-
ОК.06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности; стандарты антикоррупционного поведения и последствия его нарушения	-
ОК.07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения	-

	в чрезвычайных ситуациях	ресурсосбережения; принципы бережливого производства; основные направления изменения климатических условий региона	
ОК.08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения	-
ОК.09	Пользоваться профессиональной документацией на государственном и иностранном языках	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	-
ПК 3.1.	– разрабатывать программный код; – отлаживать приложения на различных устройствах;	– основы языков программирования; – принципы ООП и функционального программирования;	– разработки модулей программного обеспечения для мобильных платформ;

	<ul style="list-style-type: none"> <li>– работать с системами контроля версий;</li> <li>– использовать паттерны проектирования;</li> <li>– осуществлять тестирование кода;</li> <li>– производить рефакторинг;</li> <li>– интегрировать приложения с облачными сервисами</li> </ul>	<ul style="list-style-type: none"> <li>– архитектуры мобильных приложений (MVC, MVVM, VIPER);</li> <li>– принципы работы основных мобильных ОС (iOS, Android);</li> <li>– жизненный цикл мобильного приложения;</li> <li>– методы оптимизации производительности;</li> <li>– основы работы с графическим интерфейсом и анимацией;</li> <li>– основы безопасности в мобильной разработке;</li> <li>– основы работы с сетью и API;</li> <li>– принципы работы с базами данных на мобильных платформах; платформы по кроссплатформенной разработке, таких как Flutter, React Native или MAUI.</li> </ul>	<ul style="list-style-type: none"> <li>– разработки многопоточных приложений;</li> <li>– оптимизации производительности приложений;</li> <li>– работы с интеграцией сторонних библиотек</li> </ul>
ПК 3.2.	<ul style="list-style-type: none"> <li>– создавать интуитивно понятные и легко навигируемые интерфейсы;</li> <li>– использовать анимацию и переходы для улучшения пользовательского опыта;</li> <li>– оптимизировать интерфейс для работы на разных экранах и устройствах;</li> <li>– интегрировать элементы пользовательского интерфейса с серверной частью или базой данных приложения;</li> <li>– анализировать пользовательские данные и обратную</li> </ul>	<ul style="list-style-type: none"> <li>– принципы дизайна пользовательского интерфейса (UI) и пользовательского опыта (UX);</li> <li>– основы графического дизайна и типографики;</li> <li>– гайдлайны и стандарты для создания интерфейсов на платформах iOS и Android;</li> <li>– принципы адаптивного дизайна ;</li> <li>– основы работы с векторной и растровой графикой;</li> <li>– процесс проектирования интерфейса от идеи до реализации;</li> <li>– основные принципы дизайна</li> </ul>	<ul style="list-style-type: none"> <li>– создания пользовательских интерфейсов с использованием инструментов и библиотек, таких как UIKit (iOS) и Android XML (Android);</li> <li>– разработки адаптивных и мультирезолюционных интерфейсов;</li> <li>– тестирования пользовательского опыта;</li> <li>– проведения юзабилити-тестов;</li> <li>– проектирование пользовательского интерфейса (UI) и пользовательского опыта (UX) для</li> </ul>

	<p>связь для улучшения UX;</p> <ul style="list-style-type: none"> <li>– разрабатывать макеты и прототипы приложений;</li> <li>– владеть инструментами дизайна интерфейса;</li> <li>– глубоко понимать принципы дизайна пользовательского интерфейса и пользовательского опыта;</li> <li>– проводить пользовательские исследования, включая создание опросов, интервью с пользователями и анализ данных;</li> <li>– работать с прототипированием и созданием макетов пользовательского интерфейса;</li> <li>– работать в команде и эффективно взаимодействовать с разработчиками и менеджерами проектов.</li> </ul>	<p>пользовательского интерфейса, таких как иерархия информации, цветовая гамма, типографика и композиция;</p> <ul style="list-style-type: none"> <li>– психологию пользователей и их потребности при взаимодействии с веб-приложениями;</li> <li>– современные тенденции в дизайне пользовательского интерфейса и пользовательского опыта;</li> <li>– основные принципы разработки адаптивного и доступного пользовательского интерфейса;</li> <li>– основные технологии веб-разработки, такие как HTML, CSS и JavaScript.</li> </ul>	<p>различных веб-приложений и сайтов;</p> <ul style="list-style-type: none"> <li>– разработки прототипов и макетов пользовательского интерфейса с использованием инструментов, таких как Sketch, Adobe XD или Figma;</li> <li>– проведения пользовательских исследований, включая сбор обратной связи от пользователей и анализ конкурентного рынка;</li> <li>– создания дизайн-системы и стайл-гайдов для обеспечения единообразия визуального стиля и пользовательского опыта;</li> <li>– тестирования и итеративное улучшения пользовательского интерфейса на основе обратной связи пользователей.</li> </ul>
ПК 3.3.	<ul style="list-style-type: none"> <li>– проектировать и оптимизировать базы данных;</li> <li>– выполнять CRUD (Create, Read, Update, Delete) операции;</li> <li>– обеспечивать синхронизацию данных между устройствами;</li> <li>– работать с кэшированием данных;</li> <li>– обрабатывать конфликты данных в распределенных системах;</li> <li>– работать с многозадачностью и потоками данных;</li> </ul>	<ul style="list-style-type: none"> <li>– основы реляционных баз данных;</li> <li>– основы NoSQL и графовых баз данных;</li> <li>– принципы работы с транзакциями;</li> <li>– основы безопасности и шифрования данных;</li> <li>– принципы работы с миграциями баз данных;</li> <li>– основы работы с асинхронными операциями;</li> <li>– основные принципы работы с базами данных в программном обеспечении для мобильных платформ;</li> </ul>	<ul style="list-style-type: none"> <li>– работы с SQLite и другими СУБД для мобильных платформ;</li> <li>– разработки эффективных схем баз данных;</li> <li>– работы с NoSQL и графовыми базами данных;</li> <li>– работы с ORM (Object-Relational Mapping) инструментами;</li> <li>– работы с асинхронным доступом к данным;</li> <li>– разработки функций и возможностей для работы с базами данных</li> </ul>

	<ul style="list-style-type: none"> <li>– владеть языком SQL для работы с базами данных;</li> <li>– глубоко понимать принципы работы с базами данных в программном обеспечении для мобильных платформ;</li> <li>– создавать и оптимизировать структуру баз данных для хранения и обработки данных в мобильных приложениях;</li> <li>– работать с ORM (Object-Relational Mapping) инструментами для более удобного взаимодействия с базами данных;</li> <li>обеспечивать безопасность и защиту данных при работе с базами данных в мобильных приложениях.</li> </ul>	<ul style="list-style-type: none"> <li>– различные типы баз данных, таких как реляционные, NoSQL и графовые базы данных;</li> <li>– современные тенденции в разработке мобильных приложений с использованием баз данных;</li> <li>– основные принципы проектирования баз данных для эффективного хранения и обработки данных в мобильных приложениях;</li> <li>основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native, для работы с базами данных.</li> </ul>	<p>в программном обеспечении для мобильных платформ;</p> <ul style="list-style-type: none"> <li>– создания интерфейсов для работы с базами данных, включая CRUD операции (создание, чтение, обновление, удаление данных);</li> <li>– интеграции баз данных в пользовательский интерфейс приложений для удобного доступа и управления данными;</li> <li>оптимизации работы с базами данных для обеспечения высокой производительности и эффективного использования ресурсов устройства.</li> </ul>
ПК 3.4.	<ul style="list-style-type: none"> <li>– работать с разными форматами изображений и аудиофайлами;</li> <li>– создавать графические ресурсы с высоким разрешением;</li> <li>– проектировать интерфейс с учетом визуальных аспектов, таких как цвета, шрифты и стили;</li> <li>– осуществлять анимацию интерфейсных элементов;</li> <li>– обрабатывать и интегрировать аудио в приложение для воспроизведения звуков и музыки;</li> </ul>	<ul style="list-style-type: none"> <li>– основы графического дизайна и композиции;</li> <li>– различные форматы изображений и их применение;</li> <li>– основы аудиодизайна и звуковой обработки;</li> <li>– принципы анимации и визуальной привлекательности в мобильных приложениях;</li> <li>– основные принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ;</li> <li>– основные форматы и кодеки для работы с мультимедиа;</li> </ul>	<ul style="list-style-type: none"> <li>– создания и редактирования графических элементов для приложений с использованием специализированных инструментов;</li> <li>– интеграции изображений и иконок в пользовательский интерфейс;</li> <li>– разработки и анимации пользовательских элементов и переходов;</li> <li>– работы с аудиофайлами и интеграции аудио в приложение;</li> <li>– разработки мультимедийных</li> </ul>

	<ul style="list-style-type: none"> <li>– владеть инструментами для работы с мультимедиа;</li> <li>– понимать принципы работы с изображениями, видео и аудио в программном обеспечении для мобильных платформ;</li> <li>– создавать и редактировать мультимедийные файлы с использованием различных форматов и кодеков;</li> <li>– работать с анимацией и эффектами для создания привлекательных визуальных элементов в приложениях для мобильных устройств;</li> <li>– оптимизировать мультимедийные элементы для обеспечения быстрой загрузки и плавной работы на мобильных устройствах.</li> </ul>	<ul style="list-style-type: none"> <li>– современные тенденции в дизайне и использовании мультимедиа в приложениях для мобильных устройств;</li> <li>– основные принципы разработки мультимедийных функций с учетом ограниченных ресурсов мобильных устройств;</li> <li>– основные технологии разработки мобильных приложений, таких как Java, Kotlin, Swift или React Native.</li> </ul>	<p>функций и возможностей в программном обеспечении для мобильных платформ;</p> <ul style="list-style-type: none"> <li>– создания интерфейсов для работы с изображениями, видео и аудио в приложениях для мобильных устройств;</li> <li>– интеграции мультимедийных элементов в пользовательский интерфейс;</li> <li>– оптимизации работы с мультимедиа для обеспечения высокой производительности и эффективного использования ресурсов устройства;</li> <li>– получения медиа-данных с помощью механизмов в операционной системе</li> </ul>
ПК 3.5.	<ul style="list-style-type: none"> <li>– разрабатывать и запускать тестовые сценарии для проверки функциональности программного обеспечения для мобильных платформ;</li> <li>– выявлять и исправлять ошибки и несоответствия в работе ПО;</li> <li>– проводить аппаратное и программное тестирование программного обеспечения для мобильных платформ;</li> <li>– использовать инструменты анализа и отладки для поиска и устранения проблем;</li> </ul>	<ul style="list-style-type: none"> <li>– основы тестирования программного обеспечения;</li> <li>– виды тестирования (функциональное, нагрузочное, UI-тестирование и др.);</li> <li>– принципы работы с отладчиками;</li> <li>– основы continuous integration и continuous delivery (CI/CD);</li> <li>– основы создания тестовых сценариев;</li> <li>– принципы и методы тестирования программного обеспечения для мобильных платформ;</li> <li>– особенности отладки программного</li> </ul>	<ul style="list-style-type: none"> <li>– создания тестовых сценариев и единиц тестирования для мобильных платформ;</li> <li>– отладки и анализа проблем в работе мобильных приложений;</li> <li>– использования инструментов и оборудования для тестирования программных компонентов мобильных платформ;</li> <li>– работы с эмуляторами и симуляторами для программного обеспечения мобильных платформ</li> </ul>

	<ul style="list-style-type: none"> <li>– работать с инструментами для обнаружения и исправления ошибок;</li> <li>– работать с отчетами о тестировании;</li> <li>анализировать и устранять утечки памяти</li> </ul>	<p>обеспечения для мобильных платформ;</p> <ul style="list-style-type: none"> <li>– принципы работы эмуляторов и симуляторов;</li> <li>методы аппаратного и программного тестирования</li> </ul>	
ПК 3.6.	<ul style="list-style-type: none"> <li>– проектировать и реализовывать структуру запросов и ответов при работе с API;</li> <li>– аутентифицировать пользователей через сторонние сервисы, такие как OAuth;</li> <li>– обрабатывать и адаптировать данные, получаемые от сторонних сервисов, для использования в приложении;</li> <li>интегрировать функциональность социальных медиа, осуществлять доступ к аппаратным компонентам устройства и управление ими.</li> </ul>	<ul style="list-style-type: none"> <li>– принципы работы с RESTful API и другими протоколами;</li> <li>– основы OAuth и авторизации в сторонних сервисах;</li> <li>стандарты и протоколы взаимодействия с внешними сервисами</li> </ul>	<ul style="list-style-type: none"> <li>– работы с API сторонних сервисов и платформ для получения данных и функциональности;</li> <li>– интеграции социальных медиа и сетей для авторизации и обмена данными;</li> <li>– использования сторонних библиотек и SDK для расширения функциональности приложения;</li> <li>взаимодействия с аппаратными компонентами устройства</li> </ul>
ПК 3.7.	<ul style="list-style-type: none"> <li>– разрабатывать и реализовывать меры безопасности;</li> <li>– реализовывать хэширование паролей, сессионные токены и двухфакторную аутентификацию;</li> <li>– осуществлять валидацию данных, поступающих от пользователей;</li> <li>– разрабатывать политику доступа и права пользователей к данным и функциональности приложения;</li> </ul>	<ul style="list-style-type: none"> <li>– основные угрозы безопасности мобильных приложений;</li> <li>– принципы криптографии и шифрования данных;</li> <li>– стандарты и протоколы безопасности, такие как HTTPS, OAuth и OpenID Connect;</li> <li>– законодательные и регуляторные требования к защите данных, включая GDPR и HIPAA;</li> <li>– основные принципы безопасности</li> </ul>	<ul style="list-style-type: none"> <li>– разработки безопасных методов аутентификации и авторизации пользователей;</li> <li>– обработки и хранения конфиденциальных данных;</li> <li>– отслеживания и обработки уязвимостей безопасности;</li> <li>– использования шифрования для защиты данных в покое и в движении;</li> <li>– использования шифрования данных для защиты конфиденциальной</li> </ul>

	<p>реализовывать меры контроля доступа и аудита для отслеживания действий пользователей и обнаружения несанкционированных действий.</p>	<p>информации и методов ее защиты;</p> <ul style="list-style-type: none"> <li>– стандартные криптографические алгоритмы для шифрования данных;</li> <li>– методы аутентификации и авторизации пользователей, таких как OAuth или JWT;</li> <li>– многоуровневые механизмы контроля доступа к данным;</li> <li>– методы тестирования на уязвимости безопасности и опыт применения инструментов для их обнаружения;</li> <li>– принципы обеспечения безопасности передачи данных по сети;</li> </ul> <p>законодательство и регуляции в области защиты данных и умение применять их в практической разработке мобильных приложений.</p>	<p>информации, такой как пароли, персональные данные пользователей и другие чувствительные данные;</p> <ul style="list-style-type: none"> <li>– реализации механизмов аутентификации и авторизации для обеспечения доступа только авторизованным пользователям;</li> <li>– применения механизмов хеширования для защиты паролей пользователей от несанкционированного доступа;</li> <li>– обеспечения безопасности передачи данных между клиентскими устройствами и серверами с использованием протоколов шифрования, таких как SSL/TLS;</li> <li>– разработки механизмов контроля доступа к данным, чтобы предотвратить несанкционированное чтение, изменение или удаление данных;</li> <li>– проектирования и реализации систем резервного копирования и восстановления данных для обеспечения их сохранности в случае сбоев или потери устройства;</li> <li>– тестирования приложений на уязвимости безопасности, такие как SQL-инъекции, межсайтовые сценарии и другие уязвимости, и</li> </ul>
--	---	--	---

			принятие мер по их устранению; соблюдение законодательства и регуляций в области защиты данных
--	--	--	---

В результате освоения учебной дисциплины студент должен:

иметь практический опыт	<ul style="list-style-type: none"> <li>– разработки мобильных приложений под различные платформы (Android, Aurora, РЕД ОС М)</li> <li>– настройки и конфигурации проектов с использованием Gradle</li> <li>– работы с основными компонентами Android-приложений (Activity, Service, BroadcastReceiver, ContentProvider)</li> <li>– реализации взаимодействия с REST API и обработки сетевых запросов</li> <li>– работы с локальными базами данных через Room Database</li> <li>– создания асинхронных операций с использованием Kotlin Coroutines</li> <li>– реализации паттерна репозитория и работы с кэшем</li> <li>– работы с фоновыми задачами через WorkManager и AlarmManager</li> <li>– загрузки и обработки медиафайлов</li> <li>– построения экранов с бесконечной подгрузкой данных (Paging)</li> <li>– публикации приложений в различных сторсах (Google Play, RuStore, Aurora Store)</li> <li>– настройки CI/CD процессов</li> <li>– интеграции с различными API (карты, геолокация, Bluetooth, NFC)</li> <li>– адаптации приложений под разные устройства</li> </ul>
уметь	<ul style="list-style-type: none"> <li>– создавать и настраивать проекты Android-приложений</li> <li>– работать с системой сборки Gradle</li> <li>– реализовывать основные компоненты приложения</li> <li>– взаимодействовать с REST API и обрабатывать сетевые запросы</li> <li>– работать с локальными базами данных</li> <li>– реализовывать асинхронные операции</li> <li>– управлять состоянием приложения</li> <li>– работать с медиафайлами</li> <li>– реализовывать пагинацию и бесконечную прокрутку</li> <li>– настраивать публикацию приложений</li> <li>– работать с системами непрерывной интеграции</li> <li>– интегрировать различные API</li> <li>– адаптировать приложения под разные устройства</li> </ul>
знать	<ul style="list-style-type: none"> <li>– архитектуру мобильных платформ Android и Aurora</li> <li>– принципы работы основных компонентов Android-приложений</li> <li>– систему сборки и конфигурации проектов Gradle</li> <li>– принципы работы с REST API</li> <li>– основы работы с Room Database и DAO</li> <li>– механизмы асинхронного программирования в Kotlin</li> <li>– паттерн репозитория и работу с кэшем</li> <li>– принципы работы с фоновыми задачами</li> <li>– механизмы работы с медиафайлами</li> <li>– принципы построения экранов с пагинацией</li> <li>– процессы публикации приложений в сторсах</li> <li>– основы CI/CD</li> <li>– принципы интеграции с различными API</li> </ul>

	– особенности адаптации приложений под разные устройства
--	--

## 2 Структура и содержание дисциплины

### 2.1 Объем учебной дисциплины и виды работ

Виды учебной работы	Объем учебной работы, час.	
	6 семестр	7 семестр
Учебная нагрузка обучающихся всего, в том числе:	84	72
лекции	12	14
практические занятия	44	44
курсовая работа	20	-
самостоятельная работа обучающихся	8	12
консультации	-	2
промежуточная аттестация	-	-
Форма промежуточной аттестации	Курсовая работа Другая форма контроля	Зачет с оценкой

### 2.2 Тематический план и содержание дисциплины

№ п/п	Наименование тем (разделов)	Содержание тем (разделов)	Распределение часов			Формируемые компетенции	Формы текущего контроля
			Л	ПР	СРС		
<b>Раздел 1. РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ</b>							
1	Тема 1.1. Нативная разработка под Android, Aurora, РЕД ОС М	<p><b>Содержание учебного материала</b></p> <p>Архитектура Android и Aurora OS</p> <p>Activity, Service, BroadcastReceiver, ContentProvider</p> <p>Gradle, сборка и конфигурация проекта</p> <p>Работа с файловой системой и SharedPreferences</p> <p>Взаимодействие с REST API и Retrofit</p> <p>Сетевые вызовы, JSON, обработка ошибок, retry logic</p> <p>В том числе практических и лабораторных занятий</p> <p>1. Создание проекта и настройка build.gradle</p> <p>2. Работа с интендами и запуск внешних Activity</p> <p>3. Настройка Retrofit и получение данных с сервера</p> <p>4. Кэширование данных в SharedPreferences и Room</p> <p>5. Режимы работы в offline</p>	8	30	-	ОК.01 – ОК.09, ПК 3.1 – ПК 3.7	Т, ПЗ, О

№ п/п	Наименование тем (разделов)	Содержание тем (разделов)	Распределение часов			Формируемые компетенции	Формы текущего контроля
			Л	ПР	СРС		
		6. Работа с системными логами и отладкой Logcat					
2	Тема 1.2. Асинхронность, работа с данными, базы	<p><b>Содержание учебного материала</b></p> <p>Kotlin Coroutines и Flow Room Database и работа с DAO Репозиторная архитектура Работа с WorkManager, AlarmManager Хранение медиафайлов, загрузка/отправка Paging 3 и бесконечные списки В том числе практических и лабораторных занятий</p> <ol style="list-style-type: none"> <li>1. Создание асинхронных запросов с Coroutines</li> <li>2. Подключение базы данных Room</li> <li>3. Реализация репозитория с двумя источниками (сеть + кэш)</li> <li>4. Отложенные задачи с WorkManager</li> <li>5. Загрузка фото с камеры и отправка на сервер</li> <li>6. Построение экрана с подгрузкой данных (Paging)</li> </ol>	10	28	-	ОК.01 – ОК.09, ПК 3.1 – ПК 3.7	Т, ПЗ, О
3	Тема 1.3. Кроссплатформенность и публикация	<p><b>Содержание учебного материала</b></p> <p>Поддержка разных платформ: архитектура под Android, Aurora, РЕД ОС Публикация приложений: Google Play, RuStore, Aurora Store Jetpack Compose vs Flutter: возможности и ограничения CI/CD: сборка, тестирование, доставка Интеграция с API: карты, геолокация, Bluetooth, NFC Поддержка устройств: планшеты, часы, ТВ В том числе практических и лабораторных занятий</p> <ol style="list-style-type: none"> <li>1. Сборка .apk/.aab для разных маркетов</li> </ol>	8	30		ОК.01 – ОК.09, ПК 3.1 – ПК 3.7	Т, ПЗ, О

№ п/п	Наименование тем (разделов)	Содержание тем (разделов)	Распределение часов			Формируемые компетенции	Формы текущего контроля
			Л	ПР	СРС		
		2. Интеграция с Yandex Maps и Geo API 3. Работа с Bluetooth-сканером 4. Интеграция push-уведомлений (Firebase, RuStore Push) 5. Создание CI-сборки на GitHub Actions 6. Адаптация под планшеты (multi-pane интерфейс)					
		Итого часов:	26	88	20		

### 2.3. Регламент распределения видов работ по дисциплине с ДОТ

Данная дисциплина реализуется с применением дистанционных образовательных технологий (ДОТ). Распределение видов учебной работы, форматов текущего контроля представлены в Таблице 2.3.

Таблица 2.3. — Распределение видов учебной работы и текущей аттестации

Вид учебной работы	Формат проведения
Лекционные занятия	Частично с применением ДОТ
Практические занятия	Частично с применением ДОТ
Текущий контроль	Частично с применением ДОТ
Промежуточная аттестация	Контактная аудиторная работа
Формы текущего контроля	Формат проведения
Тестирование	Частично с применением ДОТ
Опрос	Контактная аудиторная работа
Практические задания	Частично с применением ДОТ

Доступ к системе дистанционных образовательных программ осуществляется каждым обучающимся самостоятельно с любого устройства на портале: <https://sziu-de.ranepa.ru>, в соответствии с их индивидуальным паролем и логином к личному кабинету/ профилю.

Текущий контроль, проводимый в системе дистанционного обучения, оцениваются как в системе дистанционного обучения, так и преподавателем вне системы.

Доступ к материалам лекций предоставляется в течение всего семестра по мере прохождения освоения программы. Доступ к каждому виду работ и количество попыток на выполнение задания предоставляется ограниченное время согласно регламенту дисциплины,

опубликованному в системе дистанционного обучения. Преподаватель оценивает выполненные обучающимися работы не позднее 14 рабочих дней после окончания срока выполнения.

### **3 Материалы текущего контроля успеваемости и промежуточной аттестации обучающихся**

#### **3.1 Формы и методы текущего контроля успеваемости и промежуточной аттестации обучающихся**

Формы текущего контроля успеваемости:

**Опрос (О)** позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления.

**Тестирование (Т)** – задания, с вариантами ответов.

Критерии оценивания

Оценки «отлично» заслуживает студент, если он ответил правильно на 90-100% вопросов теста;

Оценки «хорошо» заслуживает студент, если он ответил правильно на часть вопросов 75%-89%;

Оценки «удовлетворительно» заслуживает студент, если он правильно ответил часть вопросов 50%-74%;

Оценки «неудовлетворительно» заслуживает студент, если он правильно ответил менее чем на 50% вопросов или не представлен тест на проверку.

**Практическое задание (ПЗ)** используется для закрепления теоретических знаний и отработки навыков и умений, способности применять знания при решении конкретных задач.

**Критерии оценивания текущих и промежуточных форм аттестаций:**

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы; при выполнении теста правильно даны ответы на 90-100% вопросов

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки; при выполнении теста правильно даны ответы на 89-75% вопросов

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя; при выполнении теста правильно даны ответы на 74-50% вопросов

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя; при выполнении теста правильно даны ответы менее чем на 49%

### Формы текущего контроля

№ п/п	Название темы	Формы текущего контроля успеваемости
1	Тема 1.1. Нативная разработка под Android, Aurora, РЕД ОС М	Т, ПЗ, О
2	Тема 1.2. Асинхронность, работа с данными, базы	Т, ПЗ, О
3	Тема 1.3. Кроссплатформенность и публикация	Т, ПЗ, О

Примечание. В столбце «Форма текущего контроля успеваемости, промежуточной аттестации» перечисляются все используемые в учебном процессе по данной дисциплине формы контроля освоения материала. (Т – тестирование; ПЗ – практическое задание, О - опрос).

Условием допуска обучающегося к промежуточной аттестации является освоение материалов учебной дисциплины в объеме не менее 75 %, определенное по результатам систематического текущего контроля. Промежуточная аттестация состоит из другой формы контроля в 6 семестре и зачета с оценкой в 7 семестре. Другая форма контроля выставляется как среднеарифметическое по сумме оценок, полученных студентом за семестр по текущим аттестациям. Зачет с оценкой проводится в виде собеседования с преподавателем по изученной дисциплине и включает 1 практический вопрос (задача, кейс, расчёт, анализ ситуации) — проверяет умение применять теоретические знания для решения профессиональных или учебных задач, и 2 теоретических вопроса — проверяет знание понятийного аппарата, закономерностей, классификаций, основных положений дисциплины. По результатам выставляется отметка по пятибалльной шкале.

### 3.2 Оценочные средства текущего контроля успеваемости обучающихся

Примерная тематика курсовых проектов, работ (КР):

1. Разработка мобильного приложения с адаптивным интерфейсом под смартфоны и планшеты.
2. Создание дизайн-системы и её внедрение в мобильное приложение.
3. Проектирование и реализация интерфейса для смарт-часов (Wear OS).
4. Тестирование пользовательских интерфейсов с применением Espresso.
5. Оптимизация производительности UI средствами Android Studio.
6. Реализация многопоточности и асинхронной обработки данных (Kotlin Coroutines).
7. Интеграция мобильного приложения с REST API и обработка сетевых ошибок.
8. Хранение данных в Room Database и синхронизация с удалённым сервером.
9. Поддержка offline-режима в мобильном приложении.
10. Публикация приложения в RuStore и организация процесса обновлений.
11. Сравнительный анализ Jetpack Compose и Flutter при разработке интерфейсов.
12. Внедрение push-уведомлений в мобильное приложение.
13. Использование WorkManager для отложенных задач в Android-приложениях.
14. Разработка мобильного приложения с использованием CI/CD пайплайна.
15. Интеграция геолокации и карт (Yandex Maps API) в мобильное приложение.
16. Реализация аутентификации с использованием OAuth2 и JWT.
17. Защита мобильного приложения от MITM-атак (SSL Pinning).
18. Использование биометрической аутентификации в Android-приложении.
19. Обеспечение безопасного хранения данных с применением Android Keystore.
20. Обфускация и защита исходного кода мобильных приложений (ProGuard, R8).
21. Анализ безопасности мобильного приложения с помощью MobSF.
22. Адаптация мобильного приложения под отечественную платформу РЕД ОС.
23. Реализация кастомных UI-компонентов с использованием Canvas и Paint.
24. Создание приложения с поддержкой мультиязычности и локализацией интерфейса.
25. Внедрение системы обратной связи пользователей в мобильное приложение.

### **Примеры типовых заданий для практических работ**

#### **Тема 1.1. Нативная разработка под Android, Aurora, РЕД ОС М**

1. **Создание проекта и настройка сборки в Gradle.** Задание: создайте новый Android-проект в Android Studio и выполните:
  - настройте build.gradle (уровни проекта и модуля): зависимости для Retrofit, OkHttp, Moshi/Gson, Coroutines;

- определите правильные версии SDK, buildTools и compileOptions;
- добавьте флаги для поддержки нескольких ABI и оптимизацию размера APK.

Приложите фрагменты файлов settings.gradle, build.gradle (project и module) и краткое (3–4 предложения) описание назначения каждой зависимости. Объясните, как изменение minSdkVersion влияет на поддерживаемые устройства и доступные API.

2. **Работа с интендами и запуск внешних Activity.** Задание: реализуйте экран с кнопкой, которая:

- открывает внешнее приложение (например, браузер с URL или галерею для выбора фото);
- обрабатывает результат через registerForActivityResult;
- корректно обрабатывает отсутствие приложения (fallback-логика).

Добавьте логирование всех этапов в Logcat. Приложите код Activity/Composable, регистрацию callback и примеры логов. Кратко (2–3 предложения) опишите, какие разрешения и настройки манифеста требуются для каждого сценария.

3. **Настройка Retrofit и получение данных с сервера.** Задание: подключите Retrofit для работы с публичным REST API (например, JSONPlaceholder или аналогичное). Реализуйте:

- интерфейс API с GET-запросами и обработкой ошибок;
- OkHttp Interceptor для логирования запросов/ответов;
- обработку сетевых ошибок, таймаутов и retry logic (простая реализация без библиотек).

Приведите код интерфейса, клиента Retrofit, интерцептора и примеры успешных/ошибочных ответов из логов. Кратко (3–4 предложения) объясните, как такой подход помогает в отладке и тестировании в реальных условиях.

4. **Кэширование данных: SharedPreferences и Room.** Задание: сохраняйте и загружайте данные двумя способами:

- простые настройки (токен, тема, язык) в SharedPreferences;
- структурированные данные (список задач, пользователей) в Room (Entity, DAO, Database).

Протестируйте сценарии: перезапуск приложения, изменение данных, очистка кэша. Приложите код DAO, Entity, репозитория и примеры операций. Кратко (2–3 предложения)

сравните, когда уместно использовать каждый тип хранилища и как избежать проблем с миграциями Room.

5. **Режимы работы в offline.** Задание: реализуйте логику работы приложения без сети:

- определение доступности сети (ConnectivityManager / NetworkCallback);
- показ данных из кэша (Room) при отсутствии сети;
- индикация статуса сети пользователю и автоматическая повторная синхронизация при восстановлении соединения.

Добавьте логирование переходов между состояниями. Приложите код проверки сети, логику fallback и примеры логов. Кратко (3–4 предложения) сформулируйте требования к UX в offline-режиме (индикаторы, сообщения, очереди задач).

6. **Работа с системными логами и отладка в Logcat.** Задание: настройте структурированное логирование:

- используйте разные уровни (DEBUG, INFO, WARN, ERROR) и теги;
- исключите чувствительные данные из логов (пароли, токены);
- продемонстрируйте поиск и фильтрацию в Logcat по тегам и уровням.

Приложите примеры кода логирования и скриншоты/фрагменты Logcat с фильтрами. Кратко (2–3 предложения) объясните, почему важно контролировать логи в релизных сборках и как это делать (BuildConfig, ProGuard/R8).

## Примеры тестовых заданий

### Часть 1. Задания с выбором одного правильного ответа

1. Что делает registerForActivityResult в современной Android-разработке?
  - а) Запускает новую Activity и сразу возвращает результат
  - б) Регистрирует callback для получения результата от другой Activity или внешнего приложения
  - в) Создает новый поток для выполнения задачи
  - г) Сохраняет данные в SharedPreferences
  
2. Какой компонент Android отвечает за выполнение фоновых задач с учётом условий (Wi-Fi, зарядка)?

- a) AlarmManager
- б) Service
- в) WorkManager
- г) IntentService

3. Что означает принцип «сеть как источник истины, кэш как ускорение» в архитектуре Repository?

- a) Данные всегда берутся из сети, кэш не используется
- б) Сначала проверяется кэш, затем запрашиваются свежие данные из сети, кэш обновляется
- в) Кэш используется только при отсутствии сети, сеть игнорируется
- г) Данные хранятся только в кэше, сеть нужна лишь для первоначальной загрузки

### Часть 2. Задания на установление соответствия

4. Установите соответствие между компонентом и его назначением:

Компонент	Назначение
1. Retrofit	А. Декларативный HTTP-клиент для работы с REST API
2. Room	Б. Библиотека для работы с локальной базой данных SQLite
3. Coroutines	В. Инструмент для написания асинхронного кода в Kotlin
4. Paging 3	Г. Библиотека для эффективной подгрузки больших списков

### Часть 3. Задания на последовательность действий

5. Установите правильную последовательность шагов при интеграции геолокации в Android-приложение:

- a) Запросить разрешения на доступ к местоположению у пользователя
  - б) Проверить, есть ли уже разрешение
  - в) Получить координаты через FusedLocationProvider или аналоги
  - г) Обработать сценарии: нет разрешения, отказ пользователя, ошибка получения
  - д) Отобразить местоположение на карте или использовать в логике приложения
- Правильный порядок: б, а, в, г, д.

6. .

### **Примеры типовых вопросов для устного опроса**

1. Расскажите про основные компоненты архитектуры Android (Activity, Service, BroadcastReceiver, ContentProvider) и приведите пример использования каждого.
2. Объясните, зачем нужны Coroutines в Kotlin и чем они лучше коллбэков или RxJava для большинства задач.
3. Как устроен Room и какие проблемы он решает по сравнению с прямым использованием SQLite?
4. Что такое Paging 3 и какие преимущества он даёт при отображении больших списков?
5. Опишите жизненный цикл Intent и как правильно передавать данные между Activity.
6. Какие стратегии кэширования данных вы знаете и когда какую стоит применять (SharedPreferences, Room, in-memory)?
7. В чём отличия работы с геолокацией и Bluetooth на Android и как учитывать особенности разных устройств?
8. Расскажите, как устроен CI/CD для мобильных приложений и какие этапы обычно включают в пайплайн.
9. Как обеспечить адаптивность интерфейса под разные устройства (телефоны, планшеты, ТВ) и какие инструменты для этого есть в Android SDK?
10. Какие особенности публикации приложений в российских и международных маркетах вы знаете и как их учитывать при разработке?

### **3.3 Оценочные средства по дисциплине для промежуточной аттестации**

#### **Вопросы для подготовки к промежуточной аттестации, зачет с оценкой:**

1. Что такое Gradle и для чего он используется в Android-разработке?
1. Какие основные компоненты Android-приложения вы знаете?
2. Как настроить проект в build.gradle для разных платформ?
3. В чём разница между Activity и Service?
4. Как работает система BroadcastReceiver?
5. Для чего используется ContentProvider?
6. Как организовать работу с SharedPreferences?
7. Что такое Retrofit и как его настроить?
8. Как правильно обрабатывать сетевые ошибки в Android?

9. Что такое retry logic и как её реализовать?
10. Как работают Kotlin Coroutines?
11. В чём отличие Flow от Coroutines?
12. Как настроить Room Database?
13. Что такое DAO в контексте Room?
14. Как реализовать репозиторийный паттерн?
15. Для чего нужен WorkManager?
16. Как работает AlarmManager?
17. Как организовать хранение медиафайлов?
18. Что такое Paging 3 и как его использовать?
19. Как опубликовать приложение в Google Play?
20. Какие требования к приложениям в RuStore?
21. Как адаптировать приложение под Aurora Store?
22. В чём разница между Jetpack Compose и Flutter?
23. Как настроить CI/CD для Android-приложения?
24. Как интегрировать Yandex Maps?
25. Что такое Geo API и как с ним работать?
26. Как реализовать работу с Bluetooth?
27. Как настроить NFC в приложении?
28. Как работают push-уведомления в Firebase?
29. Как реализовать RuStore Push?
30. Как создать GitHub Actions для сборки?
31. Как адаптировать приложение под планшеты?
32. Что такое multi-pane интерфейс?
33. Как работать с системными логами?
34. Как использовать Logcat для отладки?
35. Как настроить кэширование в приложении?
36. Что такое offline-режим и как его реализовать?
37. Как организовать работу с JSON?
38. Как правильно настроить зависимости в Gradle?
39. Как оптимизировать сборку приложения?
40. Как работать с разными версиями Android?
41. Как обеспечить совместимость с РЕД ОС М?
42. Как реализовать тёмную тему?

43. Как работать с ресурсами приложения?
44. Как организовать локализацию приложения?
45. Как тестировать сетевые запросы?
46. Как оптимизировать использование памяти?
47. Как работать с базами данных в фоновом режиме?
48. Как обеспечить безопасность приложения?
49. Как обновить приложение через Google Play?

### **Примеры практических заданий(кейсов)**

#### 1.вариант

Приложение должно показывать список новостей с пагинацией, кэшировать их локально и обновлять раз в 2 часа. Опишите архитектуру решения: какие компоненты (Repository, ViewModel, Paging, Room, Retrofit) и в какой последовательности будут задействованы. Приведите краткий план реализации (5–7 пунктов).

#### 2.вариант

Пользователь запускает приложение без интернета. Какие экраны и данные должны быть доступны? Опишите UX-стратегию (минимум 4 пункта) и технические шаги (кэш, офлайн-режим, уведомления, синхронизация).

#### 3.вариант

Требуется опубликовать приложение в RuStore и Aurora Store. Перечислите отличия в подготовке сборок (подписи, форматы, требования к метаданным) и предложите единый процесс сборки, который позволит выпускать артефакты для обоих магазинов

### **4. Методические указания для обучающихся по освоению дисциплины**

Приступая к изучению дисциплины «Разработка приложений для мобильных платформ», студент должен ознакомиться с содержанием данной «Рабочей учебной программы дисциплины» с тем, чтобы иметь четкое представление о своей работе.

В первую очередь необходимо уяснить цель и задачи изучаемой дисциплины, оценить объем материала, познакомиться с предложенной и подобрать основную и дополнительную литературу, выявить наиболее важные проблемы, стоящие по вопросам изучаемой дисциплины.

Выполнение заданий осуществляется в соответствии с учебным планом и программой. Они должны выполняться в соответствии с методическими рекомендациями, выданными преподавателем, и представлены в установленные преподавателем сроки.

Работая с учебниками и учебными пособиями, целесообразно законспектировать тот материал, который не сообщался студентам на лекциях.

На занятиях лекционного и практического характера студентам для работы требуется тетрадь для записи лекций и заданий.

Для успешного овладения программой дисциплины необходимо выполнять следующие требования:

- посещать все лекционные и практические занятия;
- все рассматриваемые на лекциях и практических занятиях темы и вопросы обязательно фиксировать в тетради;
- в случае пропуска занятий по каким-либо причинам необходимо обязательно самостоятельно изучать соответствующий материал в Moodle, фиксируя записи в тетради, а также выполнять практические задания.

Подготовка к зачету с оценкой осуществляется по представленным в списке основной и дополнительной литературе. Рекомендуемые литература и интернет-ресурсы будут полезны при выполнении практических заданий и для подготовки к тестированиям.

Методические рекомендации по составлению конспекта

Конспект — сложный способ изложения содержания книги или статьи в логической последовательности. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта. Выделите главное, составьте план, представляющий собой перечень заголовков, подзаголовков, вопросов, последовательно раскрываемых затем в конспекте.

Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

При оформлении конспекта необходимо стремиться к емкости каждого предложения.

Методические рекомендации по составлению опорного конспекта

Опорный конспект — вид внеаудиторной самостоятельной работы студента по созданию краткой информационной структуры, обобщающей и отражающей суть материала лекции, темы учебника.

Опорный конспект — это наилучшая форма подготовки к ответу на вопросы.

Основная цель опорного конспекта — облегчить запоминание. Этапы составления опорного конспекта:

1. Изучить материалы темы, выбрать главное и второстепенное;
2. Установить логическую связь между элементами темы;
3. Представить характеристику элементов в краткой форме;

4. Выбрать опорные сигналы для акцентирования главной информации и отобразить в структуре работы.

Методические рекомендации по прохождению тестирования

Тестирование — это исследовательский метод, который позволяет выявить уровень знаний, умений и навыков, способностей, а также их соответствие определенным нормам усвоения, путем выполнения испытуемым ряда специальных заданий.

Следует понимать, что тестовые задания могут быть представлены в различных формах:

- задания закрытой формы, в которых обучающийся выбирает один или несколько правильных ответов из заданного набора:

- задания на дополнение (открытые задания), требующие самостоятельного получения ответов:

- задания на установления соответствия (с множественным выбором), выполнение которых связано с выявлением соответствия между элементами нескольких множеств:

- задания на установление правильной последовательности, в которых от учащегося требует указать порядок действий или процессов и другие. Этапы подготовки к тестированию:

1. Внимательно прочитайте материал по конспекту, составленному на учебном занятии. Прочитайте тот же материал по учебнику, учебному пособию.

2. Постарайтесь разобраться с непонятным, в частности новыми терминами и конструкциями.

3. Ответьте на контрольные вопросы для самопроверки, имеющиеся в учебнике, конспекте и т. д.

4. Кратко перескажите содержание изученного материала «своими словами».

5. Выучите определения основных понятий, условные обозначения, формулы и конструкции.

Подготовка к практическим занятиям

В ходе подготовки к практическим занятиям необходимо изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях, ознакомиться с программным обеспечением. Следует дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

Заканчивать подготовку следует закреплением материала с использованием соответствующих программных продуктов.

Все практические задания, предусмотренные рабочей программой, представлены в фонде оценочных средств по дисциплине.

Критерии оценивания выполненных практических работ:

- правильность выполнения работы (отсутствие фактических, логических и других ошибок);
- полнота выполнения работы;
- своевременность выполнения;
- правильность оформления отчета.

За задания, выполненные позже установленного срока или с нарушениями требований к оформлению, оценка на балл снижается.

Порядок организации самостоятельной работы студентов

Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Целью самостоятельной работы студентов является: овладение практическими знаниями, профессиональными умениями и навыками деятельности по специальности, опытом творческой, исследовательской деятельности.

Самостоятельная работа студентов предполагает:

- самостоятельный поиск ответов и необходимой информации в рамках изучаемых тем;
- выполнение заданий для самостоятельной работы, в том числе тестов;
- изучение теоретического и лекционного материала, а также основной и дополнительной литературы при подготовке к практическим занятиям.

## **5 Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»**

### **Основная литература**

1. Введение в разработку приложений для ОС Android : учебное пособие для СПО / Ю. В. Березовская, О. А. Юфрякова, В. Г. Вологодина [и др.]. — 2-е изд. — Саратов : Профобразование, 2024. — 427 с. — ISBN 978-5-4488-0993-4. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/139746>

2. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android : учебное пособие для СПО / А. Семакова. — 2-е изд. — Саратов : Профобразование, 2024. — 102 с. — ISBN 978-5-4488-0994-1. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО ПроФобразование : [сайт]. — URL: <https://profspo.ru/books/139747>

3. Нужный, А. М. Разработка мобильных приложений : учебное пособие для СПО / А. М. Нужный, Н. И. Гребенникова, В. В. Сафронов. — Саратов : Профобразование, 2022. — 92 с. — ISBN 978-5-4488-1494-5. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО ПроФобразование : [сайт]. — URL: <https://profspo.ru/books/121301>

## **6 Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы**

### *Для реализации дисциплины необходимы:*

- рабочие станции (персональные компьютеры) с характеристиками не ниже: процессор — Intel Core i5 (или аналогичный AMD), ОЗУ — 16 ГБ, SSD — не менее 256 ГБ;
- проекционное оборудование (проектор/интерактивная доска) для демонстрации материалов;
- сетевое подключение со скоростью не менее 100 Мбит/с.

### *Программное обеспечение*

Операционные системы: Windows 10/11, Linux (Ubuntu, CentOS, Astra, Alt)

Офисные пакеты: Microsoft Office 365, LibreOffice

### *Электронно-библиотечные системы (ЭБС)*

1. ЭБС «BOOK.RU». — URL: <https://book.ru/>
2. ЭБС «Znanium». — URL: <https://znanium.ru/>
3. ЭБС «Айбукс». — URL: <https://ibooks.ru/>
4. ЭБС «Лань». — URL: <https://e.lanbook.com/>
5. ЭБС «Юрайт». — URL: <https://urait.ru/>
6. Электронные каталоги библиотеки СЗИУ РАНХиГС. — URL: <https://sziu-lib.ranepa.ru/>