

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

УТВЕРЖДАЮ
Зам. директора по УР
Н.А. Андреева
« 31 » 03 2025 г.

Б1.В.ДВ.02.01 Программирование мобильных устройств

рабочая программа дисциплины

Кафедра	Информатики и вычислительной техники
Направление подготовки	09.03.01. Информатика и вычислительная техника
Профиль	Искусственный интеллект и машинное обучение
Формы обучения	очная, заочная

Распределение часов дисциплины по семестрам (для очной формы обучения (ОФО), курсам (для заочной формы обучения (ЗФО))

Вид учебной работы	ОФО		ЗФО	
	ЗЕ	часов/сем.	ЗЕ	часов/курс
Общая трудоемкость дисциплины, в том числе (по семестрам, курсам):	3	108/6	3	36/3 72/4
Контактная работа, в том числе (по семестрам, курсам):		48/6		2/3 14/4
Лекции		16/6		2/3 2/4
Лабораторные работы		16/6		6/4
Практические занятия		16/6		6/4
Семинарские занятия				
Самостоятельная работа студентов		60/6		34/3 58/4
Контроль				
Число контрольных работ (по семестрам, курсам)				1/4
Число КР (по семестрам, курсам)				
Число КП (по семестрам, курсам)				
Число зачетов (с разбивкой по семестрам, курсам)		1/6		1/4
Число экзаменов (с разбивкой по семестрам, курсам)				

Программу составил:

ст. преподаватель кафедры ИВТ Осмоловская Н.С.

Рабочая программа дисциплины

«Программирование мобильных устройств»

Разработана в соответствии с ФГОС ВО

направления подготовки 09.03.01. Информатика и вычислительная техника,
утвержденным приказом Министерства образования и науки Российской Федерации
от 19 сентября 2017 г. № 929 .

Составлена на основании учебного плана

направления 09.03.01. Информатика и вычислительная техника профиля
«Искусственный интеллект и машинное обучение», одобренного Учёным советом СКФ
МТУСИ, протокол № 8 от 31.03.2025 г., и утвержденного директором СКФ МТУСИ
31.03.2025 г.

Рассмотрена и одобрена на заседании кафедры

«Информатика и вычислительная техника»

Протокол от «31» марта 2025 г. № 8

Зав. кафедрой _____



Соколов С.В.

1. Цели изучения дисциплины

Целью освоения дисциплины «Программирование мобильных устройств» является теоретическая и практическая подготовка, которая должна обеспечить получение у обучающихся знаний и навыков по организации структур данных и алгоритмах обработки данных.

2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у обучающегося способности решать: профессиональные задачи в соответствии с проектной деятельностью.

Результатом освоения дисциплины являются сформированные у обучающегося следующие компетенции:

Компетенции обучающегося, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)	
ПК-2: Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	
Знать:	
– основы математического аппарата для осуществления проектирования систем разной сложности.	
Уметь:	
– применять теоретические знания и современные технологии для проведения проектирования (логического, функционального) систем разной сложности;	
– разрабатывать и модифицировать архитектуру информационных систем в соответствии с требованиями.	
Владеть:	
– методами выполнения работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	

3. Место дисциплины в структуре образовательной программы

Требования к предварительной подготовке обучающегося предшествующие дисциплины, модули, темы):	
1	Б1.О.08. Информационные технологии и программирование
2	Б1.О.10. Математические основы баз данных
3	Б1.О.11. Дискретная математика
4	Б1.О.26. Введение в информационные технологии
Последующие дисциплины и практики, для которых освоение данной дисциплины необходимо:	
1	Б1.В.ДВ.05.01. Разработка кроссплатформенных приложений
2	Б2.В.02(П). Производственная практика (системы искусственного интеллекта)
3	Б3.01. Выполнение и защита выпускной квалификационной работы

4. Структура и содержание дисциплины

4.1 Очная форма обучения, 4 года (всего 108 часов, из них 48 часов аудиторных)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 3, Семестр 6					
Модуль 1 – Мобильные приложения и технологии – 56 26 (10 Лек+8 ЛР+8 ПЗ) часов контактной работы – 30 часов СРС					
1.1	Лекция 1. Первые мобильные ОС и платформы: Palm, Windows Mobile, Symbian, J2ME. Современные мобильные ОС (Android, iOS, Windows Phone, Blackberry, Ubuntu Touch, Firefox OS, Sailfish, Tizen). Современные среды разработки (Eclipse, Android Studio, Visual Studio, Xcode)	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.2	Практическое занятие 1. Среда разработки приложений для мобильных устройств Android Studio. Эмулятор мобильного устройства.	ПЗ	2	ПК-2	Л2.1 Л3.1 Л3.2
1.3	Первые мобильные ОС и платформы. Предпосылки появления: ограниченность вычислительных ресурсов, необходимость энергоэффективности. Архитектурные принципы ранних систем. Ограничения доступа к аппаратным возможностям (камера, GPS) через проприетарные API. Преобладание предустановленных приложений от производителя и оператора.	СРС	6	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
1.4	Лекция 2. История ОС Android. Версии Android. Ядро Linux. Служебные компоненты: системные библиотеки, виртуальная машина Dalvik/ART, библиотеки фреймворка приложений Android	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.5	Лабораторная работа 1. Знакомство со средой программирования Android Studio. Настройка среды для разработки мобильных приложений	ЛР	2	ПК-2	Л3.1 Л3.2
1.6	Лекция 3. Основные виды Android-приложений (гибридные приложения, Android для встраиваемых систем, автомобильные приложения). Безопасность и полномочия приложения. Ключевые аспекты безопасности приложения. Манифест приложения. Управление ресурсами.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.7	Практическое занятие 2. Создание простого приложения. Изучение структуры программы.	ПЗ	2	ПК-2	Л3.1 Л3.2
1.8	Лабораторная работа 2. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна	ЛР	2	ПК-2	Л2.1 Л3.1 Л3.2
1.9	Современные мобильные ОС: переход к многоядерным процессорам, вытесняющая многозадачность с управлением памятью.	СРС	8	ПК-2	Л1.1 Л1.2 Л2.1

	Песочница (sandbox) для каждого приложения. Декларативные фреймворки. Облачная синхронизация настроек и данных между устройствами. Встроенные сервисы push-уведомлений и аналитики. Унификация разработки.				Л2.2 Л3.1 Л3.2
1.10	Лекция 4. Жизненный цикл активности. Стеки Активностей. Состояние и отслеживание изменений состояний Активности. Ресурсы мобильного приложения. Отделение ресурсов от кода программы. Создание ресурсов. Визуальные стили и темы.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.11	Практическое занятие 3. Создание приложения, которое состоит из нескольких activities. Написание приложения, работающего с разными темами/стилями.	ПЗ	2	ПК-2	Л2.1 Л2.2 Л3.1 Л3.2
1.12	Лабораторная работа 3. Управление состояниями, ресурсами и визуальным оформлением Android-приложений.	ЛР	2	ПК-2	Л3.1 Л3.2 Л2.2
1.13	Современные среды разработки. Поддержка складных устройств и периферии (часы, ТВ, автомобили). Искусственный интеллект на устройстве (вычисления на NPU). Инструменты для анализа производительности (загрузка CPU, память, сетевой трафик). Профилирование энергопотребления. Интеграция с системами контроля версий (встроенные или подключаемые клиенты Git). Кроссплатформенные инструменты.	СРС	8	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
1.14	Лекция 5. Изучение основ разработки пользовательских интерфейсов. Работа с настройками. Создание меню. Обзор элементов ввода и управления. Компоновка элементов управления пользовательского интерфейса мобильных приложений.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.15	Практическое занятие 4. Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android	ПЗ	2	ПК-2	Л1.2 Л3.1 Л3.2
1.16	Лабораторная работа 4. Создание пользовательских интерфейсов. Разработка графического интерфейса приложения.	ЛР	2	ПК-2	Л1.2 Л3.1 Л3.2
1.17	Изучение основ разработки пользовательских интерфейсов. Работа с настройками. Создание меню. Обзор элементов ввода и управления. Компоновка элементов управления пользовательского интерфейса мобильных приложений. Создание альтернативных макетов для разных ориентаций, языков, конфигураций экрана. Использование векторной графики (SVG, Vector Drawable). Автоматизированное и ручное тестирование UI: чек-листы проверки отзывчивости, удобства, соответствия гайдлайнам Material Design.	СРС	8	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
Модуль 2 – Создание модулей для мобильных приложений – 52 22 (6 Лек+8 ЛР+8 ПЗ) часов контактной работы – 30 часа СРС					

2.1	Лекция 6. Базовые механизмы создания и переключения окон, современные средства навигации в Android. Способы создания нескольких экранов. Управление стеком окон. Диалоговые окна как разновидность многооконности.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
2.2	Практическое занятие 5. Разработка многооконного мобильного приложения.	ПЗ	2	ПК-2	Л3.1 Л3.2
2.3	Лабораторная работа 5. Создание приложения, состоящего из нескольких активностей, использование диалоговых окон и элементов тач-интерфейса.	ЛР	2	ПК-2	Л1.1 Л1.2 Л3.1 Л3.2
2.4	Модели навигации в мобильных интерфейсах. Базовые механизмы создания и переключения окон, современные средства навигации в Android. Способы создания нескольких экранов. Управление стеком окон. Диалоговые окна как разновидность многооконности. Навигационные графы и конечные автоматы. Ограничения и компромиссы при проектировании многооконных интерфейсов.	СРС	6	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
2.5	Лекция 7. Локальное хранение данных в Android-приложениях: система управления базами данных SQLite, выполнение запросов, модификация данных и визуализация результатов с помощью адаптеров (ArrayAdapter, CursorAdapter, RecyclerView.Adapter) для связывания данных с пользовательским интерфейсом.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
2.6	Практическое занятие 6. Работа с базами данных в Android. Основные приемы работы с инструментами разработки.	ПЗ	2	ПК-2	Л2.2 Л3.1 Л3.2
2.7	Лабораторная работа 6. Хранение данных на устройстве. Локальные базы данных.	ЛР	2	ПК-2	Л3.1 Л3.2
2.8	Реляционная модель данных в мобильном контексте. Локальное хранение данных в Android-приложениях: система управления базами данных SQLite, выполнение запросов, модификация данных и визуализация результатов с помощью адаптеров. Сравнение SQLite с альтернативными способами локального хранения. Теоретические ограничения адаптеров при работе с большими наборами данных.	СРС	8	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
2.9	Лекция 8. Типы сенсоров и получение информации об их доступности. Использование сенсоров в Android: классификация (движение, положение, окружающая среда), проверка доступности и наличия датчиков на устройстве, выбор оптимальной частоты дискретизации, управление сенсорами с учётом жизненного цикла Activity.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
2.10	Практическое занятие 7. Изучение приемов работы с сенсорами Android.	ПЗ	2	ПК-2	Л3.1 Л3.2
2.11	Лабораторная работа 7. Разработка приложения, обладающего возможностями геолокации.	ЛР	2	ПК-2	Л3.1 Л3.2
2.12	Классификация сенсоров по принципу измерения и назначению. Аппаратные (физические) сенсоры и	СРС	8	ПК-2	Л1.1 Л1.2

	программные (синтетические) датчики. Вопрос о системных разрешениях и политиках производителя (наличие определённого типа сенсора в спецификации устройства не гарантирует его доступность для приложения). Влияние жизненного цикла Activity (переход в фон, поворот экрана) на зарегистрированные сенсоры.				Л2.1 Л2.2 Л3.1 Л3.2
2.13	Практическое занятие 8. Работа с датчиками, определение местоположения устройства.	ПЗ	2	ПК-2	Л3.1 Л3.2
2.14	Лабораторная работа 8. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек.	ЛР	2	ПК-2	Л1.1 Л3.1 Л3.2
2.15	«Виртуальные сенсоры» (например, датчик поворота, гравитации, линейного ускорения). Как они вычисляются на основе аппаратных датчиков? Способы эмуляции сенсоров в симуляторе Android. Визуализация потока событий сенсоров для отладки. Методы анализа трендов (сглаживание, устранение выбросов).	СРС	8	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
	Зачет			ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
Итого			108		

4.2. Заочная форма обучения, 5 лет (всего 108 часов, из них 16 часов аудиторных)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 3,4					
Модуль 1 – Мобильные приложения и технологии – 42 8 (2 Лек+2 ЛР+4 ПЗ) часов контактной работы – 34 часа СРС					
1.1	Лекция 1. Первые мобильные ОС и платформы. Современные мобильные ОС. Современные среды разработки. Основные виды Android-приложений (гибридные приложения, Android для встраиваемых систем, автомобильные приложения). Безопасность и полномочия приложения. Изучение основ разработки пользовательских интерфейсов.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
1.2	Практическое занятие 1. Среда разработки приложений для мобильных устройств Android Studio. Создание простого приложения.	ПЗ	2	ПК-2	Л1.1 Л3.1 Л3.2
1.3	Первые мобильные ОС и платформы. Предпосылки появления: ограниченность вычислительных ресурсов, необходимость энергоэффективности. Архитектурные принципы ранних систем. Ограничения доступа к аппаратным возможностям (камера, GPS) через проприетарные API. Преобладание предустановленных приложений от производителя и оператора.	СРС	10	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2

1.4	Современные среды разработки. Поддержка складных устройств и периферии (часы, ТВ, автомобили). Искусственный интеллект на устройстве (вычисления на NPU). Инструменты для анализа производительности (загрузка CPU, память, сетевой трафик). Профилирование энергопотребления. Интеграция с системами контроля версий (встроенные или подключаемые клиенты Git). Кроссплатформенные инструменты.	СРС	10	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
1.5	Практическое занятие 2. Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android	ПЗ	2	ПК-2	Л2.1 Л3.1 Л3.2
1.6	Лабораторная работа 1. Управление состояниями, ресурсами и визуальным оформлением Android-приложений.	ЛР	2	ПК-2	Л2.2 Л3.1 Л3.2
1.7	Изучение основ разработки пользовательских интерфейсов. Работа с настройками. Создание меню. Обзор элементов ввода и управления. Компоновка элементов управления пользовательского интерфейса мобильных приложений. Создание альтернативных макетов для разных ориентаций, языков, конфигураций экрана. Использование векторной графики (SVG, Vector Drawables). Автоматизированное и ручное тестирование UI: чек-листы проверки отзывчивости, удобства, соответствия гайдлайнам Material Design.	СРС	14	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
Модуль 2 – Создание модулей для мобильных приложений – 66 8 (2 Лек+4 ЛР+2 ПЗ) часов контактной работы – 58 часов СРС					
2.1	Лекция 2. Базовые механизмы создания и переключения окон, современные средства навигации в Android. Способы создания нескольких экранов. Диалоговые окна как разновидность многооконности. Локальное хранение данных в Android-приложениях: система управления базами данных SQLite. Типы сенсоров и получение информации об их доступности.	Лек.	2	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
2.2	Практическое занятие 3. Работа с базами данных в Android. Основные приемы работы с инструментами разработки.	ПЗ	2	ПК-2	Л2.2 Л3.1 Л3.2
2.3	Модели навигации в мобильных интерфейсах. Базовые механизмы создания и переключения окон, современные средства навигации в Android. Способы создания нескольких экранов. Управление стеком окон. Диалоговые окна как разновидность многооконности. Навигационные графы и конечные автоматы. Ограничения и компромиссы при проектировании многооконных интерфейсов.	СРС	28	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
2.4	Лабораторная работа 2. Создание приложения, состоящего из нескольких активностей, использование диалоговых окон и элементов тач-интерфейса.	ЛР	2	ПК-2	Л2.1 Л2.2 Л3.1 Л3.2

2.5	Лабораторная работа 3. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек.	ЛР	2	ПК-2	Л2.2 Л3.1 Л3.2
2.6	Реляционная модель данных в мобильном контексте. Локальное хранение данных в Android-приложениях: система управления базами данных SQLite, выполнение запросов, модификация данных и визуализация результатов с помощью адаптеров. Сравнение SQLite с альтернативными способами локального хранения. Классификация сенсоров по принципу измерения и назначению. Аппаратные (физические) сенсоры и программные (синтетические) датчики. Вопрос о системных разрешениях и политиках производителя.	СРС	30	ПК-2	Л1.1 Л1.2 Л2.1 Л2.2 Л3.1 Л3.2
	Зачет			ПК-2	Л1.1 Л1.2 Л2.1 Л2.2
Итого			108		

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Рекомендуемая литература				
5.1.1. Основная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л1.1	Соколова В. В.	Разработка мобильных приложений: учебное пособие	Москва: Ай Пи Ар Медиа, 2024	Э1
Л1.2	Березовская Ю. В., Юфрякова О. А., Вологодина В. Г.	Введение в разработку приложений для ОС Android: учебное пособие	Москва: Ай Пи Ар Медиа, 2025	Э2
5.1.2 Дополнительная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л2.1	Пирская Л. В.	Разработка мобильных приложений в среде Android Studio: учебное пособие	Ростов-на-Дону: Издательство ЮФУ, 2019	Э3
Л2.2	Семакова А.	Введение в разработку приложений для смартфонов на ОС Android: учебное пособие	Москва: Ай Пи Ар Медиа, 2025	Э4
5.1.3 Учебно-методическое обеспечение				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л3.1	Кравцов Д. В., Лосева М. А., Леонов Е. А.	Разработка приложений под мобильную платформу Android: Лабораторный практикум	Москва: ФЛИНТА, 2018	Э5
Л3.2	Лобзенко П.В., Щербань О.Г., Щербань И.В.	Учебное пособие для проведения лабораторных работ по дисциплине «Технологии языков программирования»	Ростов-на-Дону, СКФ МТУСИ, 2019 г.	Э6

5.2 Электронные образовательные ресурсы	
Э1	https://www.iprbookshop.ru/147287.html
Э2	https://www.iprbookshop.ru/146335.html
Э3	https://www.iprbookshop.ru/100196.html
Э4	https://www.iprbookshop.ru/146336.html
Э5	https://e.lanbook.com/book/113495
Э6	https://elib.mtuci.ru/view.php?book_id=170030
5.3 Информационно-справочные системы и профессиональные базы данных	
5.3.1	Консультант Плюс – https://www.consultant.ru/
5.3.2	Гарант – https://www.garant.ru/
5.3.3	Консорциум Кодекс – https://docs.cntd.ru/
5.3.4	Научная электронная библиотека – https://www.elibrary.ru/
5.3.5	Профессиональная база данных: Федеральный портал «Российское образование»: [Электронный ресурс] – Режим доступа: http://www.edu.ru/ (открытый доступ)
5.3.6	Профессиональная база данных: Федеральный центр информационно-образовательных ресурсов: [Электронный ресурс] – Режим доступа: http://fcior.edu.ru/ (открытый доступ)
5.4 Программное обеспечение	
П.1	CodeBlocks Среда программирования на языке C/C++
П.2	Android Studio – интегрированная среда разработки (IDE) для работы с платформой Android
П.3	LibreOffice – кроссплатформенный, свободно распространяемый офисный пакет
П.4	Яндекс браузер
П.6	Антивирус Касперского

6. Материально-техническое обеспечение дисциплины	
6.1 МТО лекционных занятий	
1	Лекционная аудитория, оснащенная проектором, ПК (ноутбуком), экраном
6.2 МТО лабораторных работ и практических занятий	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.3 МТО рубежных контролей, зачетов, экзаменов	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.4 МТО самостоятельной работы обучающихся	
1	Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ

7. Оценочные материалы	
Оценочные материалы и перечень видов оценочных средств текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.	
8. Методические рекомендации по освоению дисциплины	
Методические рекомендации по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.	

9. Особенности реализации дисциплины (модуля) при обучении инвалидов и лиц с ограниченными возможностями здоровья

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков.

Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания.

Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.).

Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь.

Текущий контроль успеваемости осуществляется в устной форме.

При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочесть задание, оформить ответ, общаться с преподавателем).

10. Дополнения и изменения в Рабочей программе

Лист актуализации рабочей программы дисциплины «Б1.В.ДВ.02.01 Программирование мобильных устройств» для использования в 20__/20__ учебном году

Утверждаю

Зам. директора по УР _____
«__» _____ 20__ г.

Направление: 09.03.01. Информатика и вычислительная техника

Профиль: Искусственный интеллект и машинное обучение

Форма обучения: очная, заочная

(Возможны следующие варианты):

а) Рабочая программа действует без изменений.

б) В рабочую программу вносятся следующие изменения:

- 1)
- 2)
- 3)

Разработчик (и): _____
(ФИО, ученая степень, ученое звание)

«__» _____ 20__ г.

Рабочая программа пересмотрена и одобрена на заседании кафедры _____

Протокол № _____ от «__» _____ 20__ г.

Заведующий кафедрой _____

Оценочные материалы для проведения текущего контроля и промежуточной аттестации по дисциплине

1. Оценочные материалы для проведения промежуточной аттестации по дисциплине

1.1 Шкала оценивания компетенций

Шкала оценивания компетенций		
Оценка	Уровень освоения компетенции	Критерии оценивания
«Отлично»	Высокий уровень	Обучающийся показывает всестороннее, систематическое и глубокое знание основного и дополнительного учебного материала, дает полные, профессиональные и грамотные ответы на поставленные вопросы, что свидетельствует об освоении обучающимся взаимосвязи основных понятий дисциплины; усвоил основную и знаком с дополнительной рекомендованной литературой; проявляет творческие способности в понимании, изложении и использовании учебно-программного материала.
«Хорошо»	Повышенный уровень	Обучающийся показывает достаточный уровень знания основного и дополнительного учебного материала, дает в целом грамотные ответы на поставленные вопросы, что свидетельствует об освоении обучающимся взаимосвязи основных понятий дисциплины; усвоил основную и знаком с дополнительной рекомендованной литературой. Допускает не существенные погрешности в ответах, устраняет их без помощи преподавателя.
«Удовлетворительно»	Пороговый уровень	Обучающийся демонстрирует знание основного учебного материала в минимальном объеме, необходимом для дальнейшей учебы; в целом отвечает на поставленные вопросы, допуская не принципиальные ошибки; знаком с основной литературой. Допускает ошибки в ответах различного уровня, которые исправляет после наводящих вопросов и под руководством преподавателя.
«Неудовлетворительно»	Минимальный уровень не достигнут	Обучающийся обнаруживает существенные пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении, предусмотренных программой, заданий; не знаком с рекомендованной литературой, не может исправить допущенные ошибки. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей

		дисциплине.
--	--	-------------

1.2 Показатели, критерии и шкалы оценивания компетенций

Показатели компетенции	Критерии оценивания	Шкала оценивания
ПК-2: Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности		
Знать:		
– основы математического аппарата для осуществления проектирования систем разной сложности.	Контрольная работа	Модуль 1 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
	Контрольная работа	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Уметь:		
– применять теоретические знания и современные технологии для проведения проектирования (логического, функционального) систем разной сложности;	Лабораторные работы 1-2 Практические занятия 1-2	Модуль 1 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
– разрабатывать и модифицировать архитектуру информационных систем в соответствии с требованиями.	Лабораторные работы 5-6 Практические занятия 5-6	Модуль 2 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
Владеть:		
– методами выполнения работ по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	Лабораторные работы 3-4 Практические занятия 3-4	Модуль 1 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
	Лабораторные работы 7-8 Практические занятия 7-8	Модуль 2 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
	Зачет	0-100 «Не зачтено» - 0-40 «Зачтено» - 41-100

1.3 Оценочные материалы: типовые контрольные задания, иные материалы

1.3.1.1 Оценочные материалы для очной формы обучения (ОФО)

Модуль 1 (50 баллов):

Модуль содержит 5 лекционных занятий, 4 практических и 4 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 40 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ПК-2):

1. Какая мобильная операционная система, считается одной из первых, предоставивших полноценный графический интерфейс?
2. Какие интегрированные среды разработки (IDE) являются основными при создании нативных приложений для iOS и Android?
3. В каком году компания Google приобрела Android Inc., и что послужило причиной этого приобретения?
4. Какие ключевые функции выполняет ядро Linux в ОС Android (назовите минимум три)?
5. Для чего предназначена системная библиотека Bionic в Android, и чем она отличается от стандартной libc?
6. В чем принципиальное отличие виртуальной машины ART от Dalvik с точки зрения компиляции приложений?
7. Что такое гибридное мобильное приложение, и на каких веб-технологиях оно обычно строится?
8. Какие особенности архитектуры и интерфейса характерны для автомобильных приложений под управлением Android Auto?
9. Как в Android реализована модель безопасности, основанная на полномочиях?
10. Назовите четыре ключевых аспекта безопасности, которые необходимо учитывать при разработке Android-приложения.
11. Какую роль играет файл AndroidManifest.xml в структуре приложения, и какие компоненты обязательно в нем регистрируются?
12. Опишите основные механизмы управления ресурсами (памятью, батареей, процессором) в Android.
13. Перечислите все основные методы жизненного цикла Activity в порядке их вызова от создания до уничтожения.
14. Что такое стек активностей (task back stack), и как происходит навигация назад между несколькими Activity?
15. Какие методы жизненного цикла активности вызываются при повороте экрана, и как сохранить временное состояние (например, введенный текст)?
16. Для чего используются методы onSaveInstanceState() и onRestoreInstanceState() при отслеживании изменений состояния активности?
17. В чем заключается преимущество отделения ресурсов (строк, цветов, размеров) от программного кода с точки зрения локализации и поддержки?
18. Как создать альтернативные ресурсы для разных ориентаций экрана (portrait/landscape) или разных языков?

19. Что такое визуальный стиль (Style) и тема (Theme) в Android, и чем они отличаются друг от друга?
20. Как программно или декларативно (в XML) применить кастомную тему ко всему приложению или к отдельной активности?
21. Назовите основные типы компоновок (layouts) для построения пользовательского интерфейса в Android (минимум три).
22. Какими способами можно реализовать сохранение и загрузку настроек приложения (например, SharedPreferences, DataStore)?
23. Перечислите виды меню, доступные в Android: опции, контекстное меню, всплывающее меню (popup), и в чем их основное различие?
24. Какие элементы ввода и управления (виджеты) наиболее часто используются для взаимодействия с пользователем?
25. Для чего предназначен элемент RecyclerView, и чем он отличается от ListView?
26. Как работает паттерн Adapter при связывании данных с элементами списка в Android?
27. В чем разница между LinearLayout, RelativeLayout и ConstraintLayout с точки зрения позиционирования дочерних элементов?
28. Как добавить строковый ресурс в файл strings.xml и использовать его в коде Java/Kotlin или в XML-разметке?
29. Какие преимущества дают векторные ресурсы (VectorDrawable) по сравнению с растровыми изображениями в Android?
30. Как обработать ситуацию, когда активность переходит в состояние onPause() из-за перекрытия другим приложением или звонком?
31. Какие механизмы безопасности (шифрование, песочница, SELinux) используются в Android для защиты данных приложения?
32. Как зарегистрировать в манифесте кастомную активность или сервис, чтобы система могла их запускать?
33. Что такое Intent, и какие два типа Intent существуют для навигации между активностями?
34. Назовите три способа оптимизации энергопотребления Android-приложения для продления времени работы от батареи.
35. Перечислите этапы создания простого пользовательского интерфейса в Android Studio (от создания проекта до запуска на эмуляторе).

Практическое занятие №1. Среда разработки приложений для мобильных устройств Android Studio. Эмулятор мобильного устройства.

Контрольные вопросы (ПК-2):

1. Какие два основных языка программирования официально поддерживаются Android Studio для написания логики приложения?
2. Перечислите шаги для создания нового эмулятора Android в Android Studio.
3. Какие аппаратные функции можно настраивать у эмулятора?
4. Как запустить созданное приложение на эмуляторе вместо физического устройства?
5. Как эмулировать поворот экрана (смену ориентации) на запущенном эмуляторе?
6. Для чего в эмуляторе используется инструмент «Extended controls»?
7. Что такое AVD Manager в Android Studio, и для чего он используется?
8. Как на эмуляторе Android имитировать звонок или отправку SMS без участия реальной сети?

9. С помощью какого инструмента в Android Studio можно просматривать дневник событий работы приложения в реальном времени?
10. Как подключить физическое Android-устройство к компьютеру для отладки вместо использования эмулятора?

Практическое занятие №2. Создание простого приложения.

Контрольные вопросы (ПК-2):

1. Из каких основных папок (каталогов) состоит проект в Android Studio после создания пустого приложения? Для чего предназначены папки?
2. Где в структуре проекта хранятся все ресурсы приложения (изображения, разметки, строки, цвета)?
3. Где находится Java/Kotlin-файл основной активности, и какой класс он расширяет?
4. Какой метод в классе активности вызывается первым при запуске простого приложения?
5. Что означает идентификатор ресурса вида `R.layout.activity_main`, и как его используют в коде?
6. В чём разница между файлами `build.gradle` на уровне проекта и на уровне модуля (app)?
7. Какой файл отвечает за список всех разрешений (permissions) и компонентов приложения (активности, сервисы)?
8. Где в структуре Android-проекта находятся скомпилированные файлы приложения (APK) после сборки?
9. Для чего нужна папка `res/values`, и какие файлы в ней обычно создаются?
10. Как программно обратиться к строковому ресурсу из файла `strings.xml`?
11. Что представляет собой файл `AndroidManifest.xml` с точки зрения регистрации главной активности (какой intent-filter используется)?
12. Какие три основных компонента визуальной структуры приложения можно увидеть в окне «Project» Android Studio (режим просмотра «Android» и «Project»)?

Практическое занятие №3. Создание приложения, которое состоит из нескольких Activities. Написание приложения, работающего с разными темами/стилями.

Контрольные вопросы (ПК-2):

1. Как создать новую Activity в проекте Android Studio (через меню и/или вручную)?
2. Каким образом в `AndroidManifest.xml` объявляется вторая Activity?
3. Как запустить вторую Activity из первой с помощью явного Intent (напишите базовый код на Kotlin/Java)?
4. Что такое неявный Intent, и в каком случае его использование предпочтительнее явного при работе с несколькими Activity?
5. Как передать строковое значение из одной Activity в другую с помощью метода `putExtra()` и затем извлечь его?
6. Как получить результат обратно из запущенной Activity (например, выбранный пункт списка)?
7. Что происходит с первой Activity, когда запускается вторая (какие методы жизненного цикла вызываются у первой при переходе)?
8. Как установить тему для конкретной Activity отдельно от темы всего приложения?
9. В чём разница между стилем (style) и темой (theme) в Android, и как стиль применяется к отдельному элементу интерфейса?
10. Как создать собственную тему, наследуемую от `Theme.MaterialComponents`, и изменить в ней цветовую палитру (primary, secondary, background)?
11. Как динамически сменить тему приложения во время выполнения (например, с тёмной на светлую) без перезапуска активности?
12. Каким образом приложение с несколькими Activity может использовать общие ресурсы стилей, чтобы все экраны выглядели единообразно?

13. Как передать сложный объект (свой класс данных) между Activity, если он не является примитивом или строкой?
14. Что такое «обратный стек» (back stack) для нескольких Activity, и как поведёт себя кнопка «Назад» после перехода между тремя экранами?
15. Как сделать так, чтобы при нажатии кнопки «Назад» во второй Activity первая Activity не восстанавливалась, а приложение закрывалось?

Практическое занятие №4. Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android.

Контрольные вопросы (ПК-2):

1. Как добавить кнопку (Button) на экран программно (в коде) без использования XML-разметки, и как затем привязать её к корневому Layout?
2. Какими тремя способами можно обработать нажатие на кнопку (onClick): в XML, анонимным слушателем, реализацией интерфейса View.OnClickListener?
3. Для чего используется элемент EditText, и как получить из него введённый пользователем текст в виде строки? Как ограничить ввод в EditText только цифрами?
4. В чем разница между CheckBox (флажок) и RadioButton (переключатель), и какой контейнер нужен для группы RadioButton, чтобы они были взаимоисключающими?
5. Как отследить изменение состояния переключателя ToggleButton или Switch?
6. Для чего используется элемент ProgressBar, и как изменить его видимость (показать/скрыть) во время выполнения фоновой операции?
7. Как создать выпадающий список (Spinner) и связать его с массивом строковых данных через ArrayAdapter?
8. Как добавить изображение в ImageView из ресурсов (drawable) и изменить его размеры и масштабирование (scaleType) в XML?
9. Как реализовать всплывающее сообщение (Toast) с коротким текстом, и чем отличается Toast от Snackbar?
10. Что такое диалоговое окно AlertDialog, и как добавить в него кнопки «OK» и «Cancel» с соответствующими действиями?
11. Как сделать так, чтобы при долгом нажатии на элемент (например, на кнопку) выполнялось другое действие?
12. Какие виды компоновок (Layouts) лучше всего подходят для создания формы ввода (логин/пароль с подписями) и почему?
13. Как программно изменить текст на кнопке или цвет фона элемента после какого-либо события (например, после нажатия)?

Лабораторная работа №1. Знакомство со средой программирования Android Studio. Настройка среды для разработки мобильных приложений

Контрольные вопросы (ПК-2):

1. Что такое Android SDK, и для чего он нужен при разработке?
2. Какой язык программирования официально поддерживается Android Studio по умолчанию для написания кода приложений?
3. Какие параметры нужно указать при создании нового проекта?
4. Что такое AVD (Android Virtual Device) и где в Android Studio находится менеджер для его создания?
5. Почему Android Studio базируется на IntelliJ IDEA, а не на Eclipse, как раньше? Какие это дало преимущества?

6. Как в Android Studio включить тёмную тему интерфейса и влияет ли это на тестируемое приложение или только на саму среду?
7. Что такое «Gradle» и почему при первом создании проекта Android Studio тратит много времени на его синхронизацию и загрузку файлов из интернета?
8. Чем отличается отладка приложения на реальном устройстве от отладки на эмуляторе? В каком случае лучше использовать каждое из них?
9. Почему в Android Studio существуют два режима отображения проекта – «Android» и «Project»?
10. Как с помощью настроек Android Studio ускорить сборку проекта?

Лабораторная работа №2. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна

Контрольные вопросы (ПК-2):

1. Почему при создании нового проекта в Android Studio поле «Package name» обычно выглядит как «com.example.myapplication». Что произойдёт, если назвать пакет «com.example»?
2. В файле MainActivity.kt (или .java) есть метод `onCreate`. Что означает «override», и почему без него программа не скомпилируется, но при этом ошибка будет не сразу очевидна?
3. Если в комментарии написать // TODO: исправить баг, то Android Studio подсветит это слово жёлтым. Где потом можно найти все такие пометки во всём проекте, и чем они полезны?
4. В файле activity_main.xml есть строка android:text="Hello World!". Как изменить этот текст так, чтобы он автоматически появился на 20 разных языках без изменения кода самой разметки?
5. Почему при изменении атрибута android:background у кнопки на красный цвет, а затем на зелёный, кнопка может «мигать» или менять размер? В чём отличие задания цвета через @color/red от прямого указания #FF0000?
6. Что значит конструкция setContentView(R.layout.activity_main)? Почему нельзя написать просто имя файла "activity_main.xml", а нужно R.layout.activity_main?
7. В коде активности есть закомментированные строки с Log.d(...). Что произойдёт, если раскомментировать их и запустить приложение? Где потом увидеть эти сообщения, и почему оставляют закомментированные логи?
8. Почему в XML-файле дизайна у кнопки есть атрибут android:id="@+id/myButton", а у текстового поля – android:id="@+id/myText"? Для чего нужен символ @+id, и можно ли дать двум разным элементам одинаковый ID в одной разметке?
9. Как узнать об ошибке на этапе компиляции, а не во время работы, и какой инструмент Android Studio помогает это отследить?
10. Как в комментариях к коду использовать специальные теги вроде @param, @return или @see, чтобы потом при наведении мыши на метод всплывала подсказка с документацией? Приведите пример для метода, который складывает два числа.

Лабораторная работа №3. Управление состояниями, ресурсами и визуальным оформлением Android-приложений.

Контрольные вопросы (ПК-2):

1. Почему при повороте экрана Activity пересоздаётся, и как сохранить введённый пользователем текст в поле EditText?

2. Что произойдёт с приложением, если пользователь свернул его (нажал «Домой»), а система в этот момент решила освободить память и уничтожила Activity?
3. Как приложение должно восстановить своё состояние?
4. В чём разница между сохранением состояния через onSaveInstanceState и через ViewModel? Какой способ удобнее для хранения данных при смене конфигурации (поворот, смена языка)?
5. Где в проекте лучше хранить строки, цвета и размеры, чтобы потом легко перевести приложение на другой язык или изменить всю цветовую схему без правки кода и разметки?
6. Как создать альтернативные ресурсы для ночной темы (темной темы) и для устройств с большим экраном (планшетов), не меняя код приложения?
7. Почему при изменении темы приложения с Theme.MaterialComponents.Light на Theme.MaterialComponents.Dark цвета некоторых элементов (кнопок, полей ввода) меняются автоматически, а другие – нет? Где описана эта цветовая схема?
8. Что происходит с ресурсами (картинками из папки drawable), когда приложение переходит в фоновый режим?
9. Освобождается ли память, или изображения остаются загруженными?
10. Как использовать селектор для кнопки, чтобы её цвет менялся при нажатии, и где в проекте нужно создать соответствующий XML-файл?

Лабораторная работа №4. Создание пользовательских интерфейсов. Разработка графического интерфейса приложения.

Контрольные вопросы (ПК-2):

1. Почему в Android нельзя просто нарисовать кнопку в любом месте экрана, указав координаты X и Y? Что используют вместо этого?
2. Какая разница между wrap_content и match_parent у размера элемента? Приведите пример, когда лучше использовать каждый из них.
3. Если у вас есть два поля ввода (логин и пароль) и кнопка, какой компоновщик (Layout) проще всего выбрать, чтобы они расположились вертикально друг под другом, и почему не стоит использовать AbsoluteLayout?
4. Что случится, если в одном LinearLayout с вертикальной ориентацией у двух кнопок задать layout_weight="1"? Как это повлияет на их высоту?
5. Как сделать кнопку круглой с иконкой внутри, не рисуя отдельное изображение? Какие стандартные средства Android для этого подойдут (Material Design, ShapeDrawable)?
6. Почему при создании макета для экрана входа (login) лучше использовать ConstraintLayout, чем вложенные друг в друга LinearLayout?
7. Что такое «плоская иерархия» и почему она важна для производительности?
8. Как сделать, чтобы при нажатии на любую область картинки (ImageView) выполнялось действие, но при этом сама картинка не была кнопкой?
9. Как добавить подсказку (hint) в поле ввода EditText, которая исчезает, когда пользователь начинает печатать, и появляется снова, если поле пустое? Зачем это нужно с точки зрения UX?
10. Почему при разработке интерфейса рекомендуется использовать не фиксированные размеры в пикселях (px), а dp (плотностно-независимые пиксели), и что произойдёт, если указать ширину кнопки 200px на телефоне с высокой плотностью экрана?

Модуль 2 (50 баллов):

Модуль содержит 3 лекционных занятий, 4 практических и 4 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 40 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ПК-2):

1. Как в Android реализован механизм переключения между окнами (Activity) с помощью Intent? Чем отличаются явный и неявный Intent?
2. Что такое NavController и NavGraph в современной навигации Android?
3. Как с помощью NavHostFragment организовать переходы между фрагментами без использования Activity?
4. Что такое «обратный стек» (back stack) для Activity, и как можно управлять его поведением?
5. Какие атрибуты в манифесте влияют на управление стеком окон, и как работает singleTop?
6. Чем отличается startActivity() от startActivityForResult() и от современного ActivityResultLauncher?
7. Как передать данные между окнами через Bundle или Intent.putExtra(), и какие ограничения по типам данных существуют?
8. Как с помощью ViewPager2 и FragmentStateAdapter создать несколько экранов с возможностью листания?
9. Что такое глубокие ссылки и как они используются для перехода на определённый экран приложения извне?
10. Как в Jetpack Compose организована навигация между composable-экранами (функция NavHost, composable, navigate)?
11. Что такое AlertDialog и чем он отличается от DialogFragment? Почему DialogFragment считается более правильным подходом?
12. Как создать диалог с несколькими вариантами выбора с использованием AlertDialog.Builder?
13. Как предотвратить закрытие диалогового окна при нажатии снаружи (вне области диалога) и при нажатии кнопки «Назад»?
14. Что такое BottomSheetDialogFragment, и в каких сценариях он удобнее обычного диалогового окна?
15. Как передать результат из диалога обратно в вызывающую Activity или Fragment?
16. Каким образом в Android можно создать и открыть базу данных SQLite с помощью класса SQLiteOpenHelper?
17. Какие методы обязательно нужно переопределить в наследнике SQLiteOpenHelper (onCreate, onUpgrade)? Для чего служит метод onDowngrade?
18. Как получить доступ к базе данных для чтения и записи?
19. Какие два основных способа выполнения SQL-запросов к SQLite?
20. Что представляет собой класс Cursor, и какие методы позволяют перемещаться по результатам запроса?
21. Какова роль ContentValues при вставке или обновлении данных в SQLite?

22. Что такое SQLiteDatabase.insert(), update(), delete() и как они отличаются от написания прямых SQL-команд?
23. Как выполнить сложный запрос с JOIN и агрегацией (например, подсчет суммы) в SQLite на Android?
24. Для чего используется класс SimpleCursorAdapter, и какие у него недостатки по сравнению с CursorAdapter?
25. Как RecyclerView взаимодействует с данными из SQLite через CursorAdapter или Room?
26. Что такое ORM (Object-Relational Mapping) и как библиотека Room упрощает работу с SQLite по сравнению с прямыми запросами?
27. Как обеспечить безопасность данных при работе с SQLite (избежать SQL-инъекций)? Почему нельзя просто конкатенировать строки в rawQuery()?
28. Перечислите три основные категории сенсоров в Android (движение, положение, окружающая среда) и приведите примеры для каждой.
29. Как получить список всех доступных сенсоров на устройстве с помощью SensorManager.getSensorList()?
30. Как проверить, есть ли на устройстве конкретный датчик (например, акселерометр или датчик приближения), и что вернёт getDefaultSensor() при его отсутствии?
31. Какие шаги необходимо выполнить, чтобы зарегистрировать слушатель событий сенсора (SensorEventListener)?
32. Что такое частота дискретизации (sampling period) и как задать оптимальную частоту?
33. Почему не рекомендуется использовать максимальную частоту для всех сенсоров? Какие проблемы с батареей и производительностью это вызывает?
34. Как правильно отключать сенсоры в методах жизненного цикла Activity (onPause / onResume), чтобы избежать разряда батареи и утечек памяти?
35. Какие данные предоставляет акселерометр (три оси), и как их использовать для определения тряски устройства (shake)?
36. В чём разница между акселерометром и гироскопом? Для каких задач применяется каждый из них?
37. Как получить данные от датчика магнитного поля (магнитометра) и объединить их с акселерометром для определения ориентации в пространстве?
38. Что такое датчик приближения (proximity sensor), и как его типично используют?
39. Как работать с датчиком освещённости (light sensor) и динамически изменять яркость интерфейса или тему приложения?
40. Какие ограничения существуют на эмуляторе Android при тестировании сенсоров, и как эмулировать показания датчиков?

Практическое занятие №5. Разработка многооконного мобильного приложения.

Контрольные вопросы (ПК-2):

1. Какими способами можно организовать переход с одного окна (Activity) на другое, если оба окна принадлежат одному приложению? Приведите пример кода с явным Intent.
2. Чем отличается фрагмент (Fragment) от Activity с точки зрения многооконности? В каком случае фрагмент удобнее, чем отдельное окно?
3. Как реализовать навигацию между несколькими фрагментами внутри одной Activity с помощью NavController и NavHostFragment?

4. Что произойдёт с первым окном (Activity A), если из него запустить второе окно (Activity B) стандартным способом, а затем нажать кнопку «Назад»? Опишите стек активностей.
5. Как сделать так, чтобы при запуске второго окна первое окно завершилось (чтобы при нажатии «Назад» приложение закрывалось, а не возвращалось к первому окну)? Укажите флаг Intent или метод.
6. Для чего в Android используется startActivityForResult()? Приведите пример сценария, когда нужно вернуть результат из дочернего окна.
7. Как создать диалоговое окно с подтверждением действия (AlertDialog) и обработать нажатие кнопок «Да» и «Нет» без создания отдельной Activity?
8. В чём преимущество использования DialogFragment перед обычным AlertDialog.Builder при разработке многооконного приложения (особенно с учётом жизненного цикла и поворота экрана)?
9. Как реализовать многооконный режим (split-screen / multi-window) в Android, и как проверить, находится ли приложение в этом режиме программно?
10. Как при запуске новой Activity очистить всю историю предыдущих окон, чтобы пользователь не мог вернуться к ним кнопкой «Назад»?
11. Как передать данные обратно из второго окна в первое, если первое окно уже уничтожено системой (например, при нехватке памяти)? Какой механизм гарантирует получение результата?
12. Чем отличается поведение кнопки «Назад» в системе, если приложение использует фрагменты с добавлением в back stack, от ситуации, когда используется только несколько Activity без фрагментов?

Практическое занятие №6. Работа с базами данных в Android. Основные приемы работы с инструментами разработки.

Контрольные вопросы (ПК-2):

1. Как создать базу данных SQLite в проекте Android с помощью класса SQLiteOpenHelper? Какие два обязательных метода нужно переопределить?
2. Где на устройстве (или эмуляторе) физически хранится файл базы данных .db, созданный приложением, и как получить к нему доступ через Android Studio?
3. Как скопировать файл базы данных с эмулятора на компьютер с помощью инструментов Android Studio (Device File Explorer) для просмотра сторонними?
4. С помощью какого инструмента в Android Studio можно выполнять прямые SQL-запросы к базе данных приложения во время отладки?
5. Как в Database Inspector отслеживать изменения данных в реальном времени при взаимодействии с приложением (вставка, обновление, удаление)?
6. Чем отличается метод rawQuery() от метода query() (с параметрами ContentValues) с точки зрения безопасности и удобства? В каком случае используется каждый?
7. Как правильно вставить новую запись в таблицу с помощью SQLiteDatabase.insert()? Что вернёт этот метод в случае успеха, а что при ошибке?
8. Как обновить данные в существующей записи, используя SQLiteDatabase.update()? Какие параметры нужно передать?
9. Как в Android Studio проверить, что SQL-запрос к базе данных выполняется без ошибок, и как увидеть текст ошибки (например, через Logcat и try-catch)?

10. Какие инструменты разработки помогают визуализировать связи между таблицами и проверять схему базы данных без написания кода?
11. Как выполнить миграцию базы данных (добавить новый столбец в существующую таблицу), не потеряв старые данные?

Практическое занятие №7. Изучение приемов работы с сенсорами Android.

Контрольные вопросы (ПК-2):

1. Как получить экземпляр `SensorManager` в `Activity` и какой системный сервис для этого используется?
2. С помощью какого метода `SensorManager` можно получить список всех доступных на устройстве датчиков и какого типа параметр туда передаётся?
3. Как проверить наличие конкретного датчика (например, акселерометра) и что вернёт метод `getDefaultSensor()`, если датчик отсутствует на устройстве?
4. Какой интерфейс должен реализовать класс, чтобы получать события от сенсоров, и какие два обязательных метода нужно переопределить?
5. В каком методе жизненного цикла `Activity` рекомендуется регистрировать слушатель сенсоров, а в каком – отменять регистрацию, и почему это критически важно для энергопотребления?
6. Какие константы частоты дискретизации можно передать в метод `registerListener()`, и как выбор частоты влияет на нагрузку на батарею?
7. Для чего предназначен метод `onAccuracyChanged()` и в каких случаях он вызывается (при изменении точности показаний датчика, например, у магнитометра)?
8. Как с помощью акселерометра программно определить, что пользователь встряхнул устройство (shake-детектор)? Опишите простой алгоритм на основе вычисления модуля ускорения.
9. В чём принципиальная разница между акселерометром (измеряет ускорение) и гироскопом (измеряет угловую скорость)? Для каких задач применяется каждый?
10. Как получить показания датчика освещённости (light sensor) и адаптировать интерфейс приложения (например, переключить тему или яркость экрана) в зависимости от уровня освещённости?
11. Почему при разработке приложения с использованием сенсоров нельзя полагаться на то, что датчик присутствует на всех устройствах, и как правильно обработать ситуацию его отсутствия (показать сообщение или отключить функционал)?
12. Как можно оптимизировать частоту обновления сенсора при длительной работе приложения (например, снижать частоту, когда приложение находится в фоне или экран выключен)?

Практическое занятие №8. Работа с датчиками, определение местоположения устройства.

Контрольные вопросы (ПК-2):

1. Какие два основных провайдера местоположения используются в Android для определения координат без интернета (GPS) и с интернетом (Network)? Какой из них точнее и какой быстрее?
2. Как проверить, включён ли GPS на устройстве программно, и что делать, если он выключен?
3. Как зарегистрировать слушатель обновлений местоположения и какие параметры нужно указать?
4. Как получить последнее известное местоположение перед началом активного отслеживания, и почему это значение может быть устаревшим или null?

5. Какой интерфейс нужно реализовать для получения обновлений GPS-координат, и какие методы в нём обязательны?
6. Как определить, что пользователь предоставил разрешение на определение местоположения, и какие методы для проверки и запроса используются?
7. Как привязать данные о текущем местоположении к карте (например, Google Maps или OpenStreetMap) в приложении, и какая библиотека для этого чаще всего используется?
8. Как объединить данные с акселерометра и гироскопа с показаниями GPS для более точного отслеживания перемещения, когда сигнал GPS временно пропадает?
9. Как оптимизировать энергопотребление приложения?
10. Как получить адрес (город, улицу) по координатам и какие могут быть проблемы?

Лабораторная работа №5. Создание приложения, состоящего из нескольких активностей, использование диалоговых окон и элементов тач-интерфейса.

Контрольные вопросы (ПК-2):

1. Как проще всего организовать переход между двумя активностями и передать при этом строку, например, имя пользователя?
2. Что нужно добавить в AndroidManifest.xml, чтобы вторая активность была видна системе, и что будет, если этого не сделать?
3. В каком методе жизненного цикла первой активности она теряет фокус, когда открывается вторая?
4. Какой диалог проще всего показать для запроса подтверждения действия (например, «Удалить файл?») с кнопками «Да» и «Нет»?
5. Почему для диалогов лучше использовать DialogFragment, а не просто AlertDialog?
6. Как определить, что пользователь провёл пальцем по экрану (свайп) и, например, переключить на следующую активность?
7. Какое стандартное диалоговое окно позволяет пользователю выбрать дату (например, для регистрации дня рождения), и как его вызвать из активности?
8. При переходе на вторую активность первая может быть уничтожена системой. Как сохранить состояние (например, введённый текст) перед уходом в фоновый режим?
9. Как сделать диалоговое окно, которое не закрывается при касании снаружи (вне его области)?
10. Чем отличается касание (tap) от долгого касания (long press) на элементе интерфейса, и как для кнопки обработать долгое нажатие?

Лабораторная работа №6. Хранение данных на устройстве. Локальные базы данных.

Контрольные вопросы (ПК-2):

1. Какие три простых способа локального хранения данных в Android (кроме полноценной SQLite) вы изучили?
2. Когда лучше использовать SharedPreferences, а когда – базу данных SQLite? Приведите примеры.
3. Какой класс в Android отвечает за создание и управление версиями SQLite-базы данных?
4. Какой метод вызывается при первом создании БД, а какой – при обновлении версии?
5. Как вставить новую запись в таблицу SQLite, не составляя SQL-запрос вручную? Какой класс удобно использовать для пар ключ-значение?
6. Почему все операции с базой данных (особенно вставка, обновление, выборка большого объёма) рекомендуется выполнять в фоновом потоке, а не в UI-потоке?

7. Что такое Room Persistence Library, и какое основное преимущество она даёт по сравнению с чистым SQLiteOpenHelper?
8. Как в Room определить сущность для таблицы users с полями id (первичный ключ) и name?
9. Что произойдёт с данными в базе при обновлении приложения, если вы увеличили версию базы в SQLiteOpenHelper с 1 до 2, но не реализовали onUpgrade()?
10. Как в Database Inspector (в Android Studio) выполнить запрос к своей базе, чтобы увидеть все записи в таблице? Можно ли там же изменить данные прямо во время отладки?

Лабораторная работа №7. Разработка приложения, обладающего возможностями геолокации.

Контрольные вопросы (ПК-2):

1. Как называется системный сервис в Android, который отвечает за получение данных о местоположении устройства, и как его получить в коде активности?
2. Какие два основных провайдера местоположения встроены в Android?
3. Какой из них обычно быстрее выдает первые координаты, а какой – точнее?
4. Какое опасное разрешение нужно добавить в AndroidManifest.xml, чтобы получить доступ к точному местоположению по GPS?
5. Как получить последнее известное местоположение устройства без запуска постоянного отслеживания? Какой метод LocationManager для этого используется?
6. Какой интерфейс (слушатель) нужно реализовать, чтобы ваше приложение получало обновления координат по мере перемещения устройства?
7. С помощью какого метода LocationManager вы начинаете получать обновления геопозиции?
8. Как вычислить расстояние (в метрах) между текущим местоположением и заданной точкой (например, достопримечательностью), используя готовый метод класса Location?
9. Для чего в приложении с геолокацией нужно отключать обновления местоположения?
10. Как на эмуляторе Android Studio проверить работу приложения с геолокацией, если у вас нет физического устройства? Укажите простой способ «послать» выбранные координаты в приложение.

Лабораторная работа №8. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек.

Контрольные вопросы (ПК-2):

1. Как передать текущие координаты (широту и долготу) из одной активности в другую с помощью Intent и putExtra()? Приведите пример кода отправки двух чисел double.
2. Какая самая популярная сторонняя библиотека от Google для работы с геолокацией упрощает получение координат вместо прямого использования LocationManager?
3. Почему при использовании FusedLocationProviderClient не нужно самостоятельно выбирать провайдера (GPS/Network)? Какое преимущество это даёт разработчику?
4. Как в многооконном приложении (например, с двумя активностями) правильно останавливать обновления геолокации, когда пользователь перешёл на второй экран, чтобы не разряжать батарею? В каком методе жизненного цикла это лучше сделать?
5. Какая сторонняя библиотека чаще всего используется для отображения карт в Android-приложении, и какой API ключ нужно получить для её работы?
6. Как запросить разрешение на точное местоположение во второй активности, если оно ещё не выдано, и обработать результат перехода из первой активности?
7. Что произойдёт с фоновым обновлением геопозиции, если пользователь свернул приложение или переключился на другую активность, но вы не отписали слушатель?

8. Как с помощью сторонней библиотеки Retrofit или OkHttp отправить текущие координаты на удалённый сервер (например, при нажатии кнопки) из любой активности?
9. В многооконном режиме (split-screen) две активности одного приложения могут быть видны одновременно. Как проверить, что ваше приложение находится в этом режиме, и нужно ли как-то по-особому обрабатывать геолокацию?
10. Что такое Google Maps Compose Library (для Jetpack Compose) или Maps SDK for Android (для XML)? Приведите пример, как добавить карту на экран второй активности с маркером в текущем местоположении.

1.3.1.2 Оценочные материалы для заочной формы обучения

Модуль 1 (50 баллов):

Модуль содержит 1 лекционное занятие, 2 практических занятия и 1 лабораторную работу. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 20. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 30 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ПК-2):

1. Какая мобильная операционная система, считается одной из первых, предоставивших полноценный графический интерфейс?
2. Какие интегрированные среды разработки (IDE) являются основными при создании нативных приложений для iOS и Android?
3. В каком году компания Google приобрела Android Inc., и что послужило причиной этого приобретения?
4. Какие ключевые функции выполняет ядро Linux в ОС Android (назовите минимум три)?
5. Для чего предназначена системная библиотека Bionic в Android, и чем она отличается от стандартной libc?
6. В чем принципиальное отличие виртуальной машины ART от Dalvik с точки зрения компиляции приложений?
7. Что такое гибридное мобильное приложение, и на каких веб-технологиях оно обычно строится?
8. Какие особенности архитектуры и интерфейса характерны для автомобильных приложений под управлением Android Auto?
9. Как в Android реализована модель безопасности, основанная на полномочиях?
10. Назовите четыре ключевых аспекта безопасности, которые необходимо учитывать при разработке Android-приложения.
11. Какую роль играет файл AndroidManifest.xml в структуре приложения, и какие компоненты обязательно в нем регистрируются?
12. Опишите основные механизмы управления ресурсами (памятью, батареей, процессором) в Android.
13. Перечислите все основные методы жизненного цикла Activity в порядке их вызова от создания до уничтожения.

14. Что такое стек активностей (task back stack), и как происходит навигация назад между несколькими Activity?
15. Какие методы жизненного цикла активности вызываются при повороте экрана, и как сохранить временное состояние (например, введенный текст)?
16. Для чего используются методы onSaveInstanceState() и onRestoreInstanceState() при отслеживании изменений состояния активности?
17. В чем заключается преимущество отделения ресурсов (строк, цветов, размеров) от программного кода с точки зрения локализации и поддержки?
18. Как создать альтернативные ресурсы для разных ориентаций экрана (portrait/landscape) или разных языков?
19. Что такое визуальный стиль (Style) и тема (Theme) в Android, и чем они отличаются друг от друга?
20. Как программно или декларативно (в XML) применить кастомную тему ко всему приложению или к отдельной активности?
21. Назовите основные типы компоновок (layouts) для построения пользовательского интерфейса в Android (минимум три).
22. Какими способами можно реализовать сохранение и загрузку настроек приложения (например, SharedPreferences, DataStore)?
23. Перечислите виды меню, доступные в Android: опции, контекстное меню, всплывающее меню (popup), и в чем их основное различие?
24. Какие элементы ввода и управления (виджеты) наиболее часто используются для взаимодействия с пользователем?
25. Для чего предназначен элемент RecyclerView, и чем он отличается от ListView?
26. Как работает паттерн Adapter при связывании данных с элементами списка в Android?
27. В чем разница между LinearLayout, RelativeLayout и ConstraintLayout с точки зрения позиционирования дочерних элементов?
28. Как добавить строковый ресурс в файл strings.xml и использовать его в коде Java/Kotlin или в XML-разметке?
29. Какие преимущества дают векторные ресурсы (VectorDrawable) по сравнению с растровыми изображениями в Android?
30. Как обработать ситуацию, когда активность переходит в состояние onPause() из-за перекрытия другим приложением или звонком?
31. Какие механизмы безопасности (шифрование, песочница, SELinux) используются в Android для защиты данных приложения?
32. Как зарегистрировать в манифесте кастомную активность или сервис, чтобы система могла их запускать?
33. Что такое Intent, и какие два типа Intent существуют для навигации между активностями?
34. Назовите три способа оптимизации энергопотребления Android-приложения для продления времени работы от батареи.
35. Перечислите этапы создания простого пользовательского интерфейса в Android Studio (от создания проекта до запуска на эмуляторе).

Практическое занятие №1. Среда разработки приложений для мобильных устройств Android Studio. Эмулятор мобильного устройства.
Контрольные вопросы (ПК-2):

1. Какие два основных языка программирования официально поддерживаются Android Studio для написания логики приложения?
2. Перечислите шаги для создания нового эмулятора Android в Android Studio.
3. Какие аппаратные функции можно настраивать у эмулятора?
4. Как запустить созданное приложение на эмуляторе вместо физического устройства?
5. Как эмулировать поворот экрана (смену ориентации) на запущенном эмуляторе?
6. Для чего в эмуляторе используется инструмент «Extended controls»?
7. Что такое AVD Manager в Android Studio, и для чего он используется?
8. Как на эмуляторе Android имитировать звонок или отправку SMS без участия реальной сети?
9. С помощью какого инструмента в Android Studio можно просматривать дневник событий работы приложения в реальном времени?
10. Как подключить физическое Android-устройство к компьютеру для отладки вместо использования эмулятора?

Практическое занятие №2. Создание пользовательских интерфейсов и использование элементов управления в приложениях под Android.

Контрольные вопросы (ПК-2):

1. Как добавить кнопку (Button) на экран программно (в коде) без использования XML-разметки, и как затем привязать её к корневому Layout?
2. Какими тремя способами можно обработать нажатие на кнопку (onClick): в XML, анонимным слушателем, реализацией интерфейса View.OnClickListener?
3. Для чего используется элемент EditText, и как получить из него введенный пользователем текст в виде строки? Как ограничить ввод в EditText только цифрами?
4. В чем разница между CheckBox (флажок) и RadioButton (переключатель), и какой контейнер нужен для группы RadioButton, чтобы они были взаимоисключающими?
5. Как отследить изменение состояния переключателя ToggleButton или Switch?
6. Для чего используется элемент ProgressBar, и как изменить его видимость (показать/скрыть) во время выполнения фоновой операции?
7. Как создать выпадающий список (Spinner) и связать его с массивом строковых данных через ArrayAdapter?
8. Как добавить изображение в ImageView из ресурсов (drawable) и изменить его размеры и масштабирование (scaleType) в XML?
9. Как реализовать всплывающее сообщение (Toast) с коротким текстом, и чем отличается Toast от Snackbar?
10. Что такое диалоговое окно AlertDialog, и как добавить в него кнопки «ОК» и «Cancel» с соответствующими действиями?
11. Как сделать так, чтобы при долгом нажатии на элемент (например, на кнопку) выполнялось другое действие?
12. Какие виды компоновок (Layouts) лучше всего подходят для создания формы ввода (логин/пароль с подписями) и почему?
13. Как программно изменить текст на кнопке или цвет фона элемента после какого-либо события (например, после нажатия)?

Лабораторная работа №1. Управление состояниями, ресурсами и визуальным оформлением Android-приложений.

Контрольные вопросы (ПК-2):

1. Почему при повороте экрана Activity пересоздаётся, и как сохранить введённый пользователем текст в поле EditText?
2. Что произойдёт с приложением, если пользователь свернул его (нажал «Домой»), а система в этот момент решила освободить память и уничтожила Activity?
3. Как приложение должно восстановить своё состояние?
4. В чём разница между сохранением состояния через onSaveInstanceState и через ViewModel? Какой способ удобнее для хранения данных при смене конфигурации (поворот, смена языка)?
5. Где в проекте лучше хранить строки, цвета и размеры, чтобы потом легко перевести приложение на другой язык или изменить всю цветовую схему без правки кода и разметки?
6. Как создать альтернативные ресурсы для ночной темы (темной темы) и для устройств с большим экраном (планшетов), не меняя код приложения?
7. Почему при изменении темы приложения с Theme.MaterialComponents.Light на Theme.MaterialComponents.Dark цвета некоторых элементов (кнопок, полей ввода) меняются автоматически, а другие – нет? Где описана эта цветовая схема?
8. Что происходит с ресурсами (картинками из папки drawable), когда приложение переходит в фоновый режим?
9. Освобождается ли память, или изображения остаются загруженными?
10. Как использовать селектор для кнопки, чтобы её цвет менялся при нажатии, и где в проекте нужно создать соответствующий XML-файл?

Модуль 2 (50 баллов):

Модуль содержит 1 лекционное занятие, 1 практическое занятие и 2 лабораторные работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 20. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 30 баллов. Общее максимальное количество баллов за модуль 2 составляет 50 баллов.

Вопросы для контрольной работы (ПК-2):

1. Как в Android реализован механизм переключения между окнами (Activity) с помощью Intent? Чем отличаются явный и неявный Intent?
2. Что такое NavController и NavGraph в современной навигации Android?
3. Как с помощью NavHostFragment организовать переходы между фрагментами без использования Activity?
4. Что такое «обратный стек» (back stack) для Activity, и как можно управлять его поведением?
5. Какие атрибуты в манифесте влияют на управление стеком окон, и как работает singleTop?
6. Чем отличается startActivity() от startActivityForResult() и от современного ActivityResultLauncher?
7. Как передать данные между окнами через Bundle или Intent.putExtra(), и какие ограничения по типам данных существуют?
8. Как с помощью ViewPager2 и FragmentStateAdapter создать несколько экранов с возможностью листания?

9. Что такое глубокие ссылки и как они используются для перехода на определённый экран приложения извне?
10. Как в Jetpack Compose организована навигация между composable-экранами (функция `NavHost`, `composable`, `navigate`)?
11. Что такое `AlertDialog` и чем он отличается от `DialogFragment`? Почему `DialogFragment` считается более правильным подходом?
12. Как создать диалог с несколькими вариантами выбора с использованием `AlertDialog.Builder`?
13. Как предотвратить закрытие диалогового окна при нажатии снаружи (вне области диалога) и при нажатии кнопки «Назад»?
14. Что такое `BottomSheetDialogFragment`, и в каких сценариях он удобнее обычного диалогового окна?
15. Как передать результат из диалога обратно в вызывающую `Activity` или `Fragment`?
16. Каким образом в Android можно создать и открыть базу данных `SQLite` с помощью класса `SQLiteOpenHelper`?
17. Какие методы обязательно нужно переопределить в наследнике `SQLiteOpenHelper` (`onCreate`, `onUpgrade`)? Для чего служит метод `onDowngrade`?
18. Как получить доступ к базе данных для чтения и записи?
19. Какие два основных способа выполнения SQL-запросов к `SQLite`?
20. Что представляет собой класс `Cursor`, и какие методы позволяют перемещаться по результатам запроса?
21. Какова роль `ContentValues` при вставке или обновлении данных в `SQLite`?
22. Что такое `SQLiteDatabase.insert()`, `update()`, `delete()` и как они отличаются от написания прямых SQL-команд?
23. Как выполнить сложный запрос с `JOIN` и агрегацией (например, подсчет суммы) в `SQLite` на Android?
24. Для чего используется класс `SimpleCursorAdapter`, и какие у него недостатки по сравнению с `CursorAdapter`?
25. Как `RecyclerView` взаимодействует с данными из `SQLite` через `CursorAdapter` или `Room`?
26. Что такое ORM (Object-Relational Mapping) и как библиотека `Room` упрощает работу с `SQLite` по сравнению с прямыми запросами?
27. Как обеспечить безопасность данных при работе с `SQLite` (избежать SQL-инъекций)? Почему нельзя просто конкатенировать строки в `rawQuery()`?
28. Перечислите три основные категории сенсоров в Android (движение, положение, окружающая среда) и приведите примеры для каждой.
29. Как получить список всех доступных сенсоров на устройстве с помощью `SensorManager.getSensorList()`?
30. Как проверить, есть ли на устройстве конкретный датчик (например, акселерометр или датчик приближения), и что вернёт `getDefaultSensor()` при его отсутствии?
31. Какие шаги необходимо выполнить, чтобы зарегистрировать слушатель событий сенсора (`SensorEventListener`)?
32. Что такое частота дискретизации (`sampling period`) и как задать оптимальную частоту?
33. Почему не рекомендуется использовать максимальную частоту для всех сенсоров? Какие проблемы с батареей и производительностью это вызывает?
34. Как правильно отключать сенсоры в методах жизненного цикла `Activity` (`onPause` / `onResume`), чтобы избежать разряда батареи и утечек памяти?

35. Какие данные предоставляет акселерометр (три оси), и как их использовать для определения тряски устройства (shake)?
36. В чём разница между акселерометром и гироскопом? Для каких задач применяется каждый из них?
37. Как получить данные от датчика магнитного поля (магнитометра) и объединить их с акселерометром для определения ориентации в пространстве?
38. Что такое датчик приближения (proximity sensor), и как его типично используют?
39. Как работать с датчиком освещённости (light sensor) и динамически изменять яркость интерфейса или тему приложения?
40. Какие ограничения существуют на эмуляторе Android при тестировании сенсоров, и как эмулировать показания датчиков?

Практическое занятие №3. Работа с базами данных в Android. Основные приемы работы с инструментами разработки.

Контрольные вопросы (ПК-2):

1. Как создать базу данных SQLite в проекте Android с помощью класса SQLiteOpenHelper? Какие два обязательных метода нужно переопределить?
2. Где на устройстве (или эмуляторе) физически хранится файл базы данных .db, созданный приложением, и как получить к нему доступ через Android Studio?
3. Как скопировать файл базы данных с эмулятора на компьютер с помощью инструментов Android Studio (Device File Explorer) для просмотра сторонними?
4. С помощью какого инструмента в Android Studio можно выполнять прямые SQL-запросы к базе данных приложения во время отладки?
5. Как в Database Inspector отслеживать изменения данных в реальном времени при взаимодействии с приложением (вставка, обновление, удаление)?
6. Чем отличается метод rawQuery() от метода query() (с параметрами ContentValues) с точки зрения безопасности и удобства? В каком случае используется каждый?
7. Как правильно вставить новую запись в таблицу с помощью SQLiteDatabase.insert()? Что вернёт этот метод в случае успеха, а что при ошибке?
8. Как обновить данные в существующей записи, используя SQLiteDatabase.update()? Какие параметры нужно передать?
9. Как в Android Studio проверить, что SQL-запрос к базе данных выполняется без ошибок, и как увидеть текст ошибки (например, через Logcat и try-catch)?
10. Какие инструменты разработки помогают визуализировать связи между таблицами и проверять схему базы данных без написания кода?
11. Как выполнить миграцию базы данных (добавить новый столбец в существующую таблицу), не потеряв старые данные?

Лабораторная работа №2. Создание приложения, состоящего из нескольких активностей, использование диалоговых окон и элементов тач-интерфейса.

Контрольные вопросы (ПК-2):

1. Как проще всего организовать переход между двумя активностями и передать при этом строку, например, имя пользователя?
2. Что нужно добавить в AndroidManifest.xml, чтобы вторая активность была видна системе, и что будет, если этого не сделать?

3. В каком методе жизненного цикла первой активности она теряет фокус, когда открывается вторая?
4. Какой диалог проще всего показать для запроса подтверждения действия (например, «Удалить файл?») с кнопками «Да» и «Нет»?
5. Почему для диалогов лучше использовать DialogFragment, а не просто AlertDialog?
6. Как определить, что пользователь провёл пальцем по экрану (свайп) и, например, переключить на следующую активность?
7. Какое стандартное диалоговое окно позволяет пользователю выбрать дату (например, для регистрации дня рождения), и как его вызвать из активности?
8. При переходе на вторую активность первая может быть уничтожена системой. Как сохранить состояние (например, введенный текст) перед уходом в фоновый режим?
9. Как сделать диалоговое окно, которое не закрывается при касании снаружи (вне его области)?
10. Чем отличается касание (tap) от долгого касания (long press) на элементе интерфейса, и как для кнопки обработать долгое нажатие?

Лабораторная работа №3. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек.

Контрольные вопросы (ПК-2):

1. Как передать текущие координаты (широту и долготу) из одной активности в другую с помощью Intent и putExtra()? Приведите пример кода отправки двух чисел double.
2. Какая самая популярная сторонняя библиотека от Google для работы с геолокацией упрощает получение координат вместо прямого использования LocationManager?
3. Почему при использовании FusedLocationProviderClient не нужно самостоятельно выбирать провайдера (GPS/Network)? Какое преимущество это даёт разработчику?
4. Как в многооконном приложении (например, с двумя активностями) правильно останавливать обновления геолокации, когда пользователь перешёл на второй экран, чтобы не разряжать батарею? В каком методе жизненного цикла это лучше сделать?
5. Какая сторонняя библиотека чаще всего используется для отображения карт в Android-приложении, и какой API ключ нужно получить для её работы?
6. Как запросить разрешение на точное местоположение во второй активности, если оно ещё не выдано, и обработать результат перехода из первой активности?
7. Что произойдёт с фоновым обновлением геопозиции, если пользователь свернул приложение или переключился на другую активность, но вы не отписали слушатель?
8. Как с помощью сторонней библиотеки Retrofit или OkHttp отправить текущие координаты на удалённый сервер (например, при нажатии кнопки) из любой активности?
9. В многооконном режиме (split-screen) две активности одного приложения могут быть видны одновременно. Как проверить, что ваше приложение находится в этом режиме, и нужно ли как-то по-особому обрабатывать геолокацию?
10. Что такое Google Maps Compose Library (для Jetpack Compose) или Maps SDK for Android (для XML)? Приведите пример, как добавить карту на экран второй активности с маркером в текущем местоположении.

1.3.2.1 Вопросы, для очной и заочной форм обучения, выносимые на зачет по дисциплине «Программирование мобильных устройств» (ПК-2)

1. Какая мобильная операционная система, считается одной из первых, предоставивших полноценный графический интерфейс?
2. Какие интегрированные среды разработки (IDE) являются основными при создании нативных приложений для iOS и Android?
3. В каком году компания Google приобрела Android Inc., и что послужило причиной этого приобретения?
4. Какие ключевые функции выполняет ядро Linux в ОС Android (назовите минимум три)?
5. Для чего предназначена системная библиотека Bionic в Android, и чем она отличается от стандартной libc?
6. В чем принципиальное отличие виртуальной машины ART от Dalvik с точки зрения компиляции приложений?
7. Что такое гибридное мобильное приложение, и на каких веб-технологиях оно обычно строится?
8. Какие особенности архитектуры и интерфейса характерны для автомобильных приложений под управлением Android Auto?
9. Как в Android реализована модель безопасности, основанная на полномочиях?
10. Назовите четыре ключевых аспекта безопасности, которые необходимо учитывать при разработке Android-приложения.
11. Какую роль играет файл AndroidManifest.xml в структуре приложения, и какие компоненты обязательно в нем регистрируются?
12. Опишите основные механизмы управления ресурсами (памятью, батареей, процессором) в Android.
13. Перечислите все основные методы жизненного цикла Activity в порядке их вызова от создания до уничтожения.
14. Что такое стек активностей (task back stack), и как происходит навигация назад между несколькими Activity?
15. Какие методы жизненного цикла активности вызываются при повороте экрана, и как сохранить временное состояние (например, введенный текст)?
16. Для чего используются методы onSaveInstanceState() и onRestoreInstanceState() при отслеживании изменений состояния активности?
17. В чем заключается преимущество отделения ресурсов (строк, цветов, размеров) от программного кода с точки зрения локализации и поддержки?
18. Как создать альтернативные ресурсы для разных ориентаций экрана (portrait/landscape) или разных языков?
19. Что такое визуальный стиль (Style) и тема (Theme) в Android, и чем они отличаются друг от друга?
20. Как программно или декларативно (в XML) применить кастомную тему ко всему приложению или к отдельной активности?
21. Назовите основные типы компоновок (layouts) для построения пользовательского интерфейса в Android (минимум три).
22. Какими способами можно реализовать сохранение и загрузку настроек приложения (например, SharedPreferences, DataStore)?

23. Перечислите виды меню, доступные в Android: опции, контекстное меню, всплывающее меню (popup), и в чем их основное различие?
24. Какие элементы ввода и управления (виджеты) наиболее часто используются для взаимодействия с пользователем?
25. Для чего предназначен элемент RecyclerView, и чем он отличается от ListView?
26. Как работает паттерн Adapter при связывании данных с элементами списка в Android?
27. В чем разница между LinearLayout, RelativeLayout и ConstraintLayout с точки зрения позиционирования дочерних элементов?
28. Как добавить строковый ресурс в файл strings.xml и использовать его в коде Java/Kotlin или в XML-разметке?
29. Какие преимущества дают векторные ресурсы (VectorDrawable) по сравнению с растровыми изображениями в Android?
30. Как обработать ситуацию, когда активность переходит в состояние onPause() из-за перекрытия другим приложением или звонком?
31. Какие механизмы безопасности (шифрование, песочница, SELinux) используются в Android для защиты данных приложения?
32. Как зарегистрировать в манифесте кастомную активность или сервис, чтобы система могла их запускать?
33. Назовите три способа оптимизации энергопотребления Android-приложения для продления времени работы от батареи.
34. Перечислите этапы создания простого пользовательского интерфейса в Android Studio (от создания проекта до запуска на эмуляторе).
35. Как в Android реализован механизм переключения между окнами (Activity) с помощью Intent? Чем отличаются явный и неявный Intent?
36. Что такое NavController и NavGraph в современной навигации Android?
37. Как с помощью NavHostFragment организовать переходы между фрагментами без использования Activity?
38. Что такое «обратный стек» (back stack) для Activity, и как можно управлять его поведением?
39. Какие атрибуты в манифесте влияют на управление стеком окон, и как работает singleTop?
40. Чем отличается startActivity() от startActivityForResult() и от современного ActivityResultLauncher?
41. Как передать данные между окнами через Bundle или Intent.putExtra(), и какие ограничения по типам данных существуют?
42. Что такое глубокие ссылки и как они используются для перехода на определённый экран приложения извне?
43. Как в Jetpack Compose организована навигация между composable-экранами (функция NavHost, composable, navigate)?
44. Что такое AlertDialog и чем он отличается от DialogFragment? Почему DialogFragment считается более правильным подходом?
45. Как создать диалог с несколькими вариантами выбора с использованием AlertDialog.Builder?
46. Как предотвратить закрытие диалогового окна при нажатии снаружи (вне области диалога) и при нажатии кнопки «Назад»?
47. Как передать результат из диалога обратно в вызывающую Activity или Fragment?

48. Каким образом в Android можно создать и открыть базу данных SQLite с помощью класса SQLiteOpenHelper?
49. Какие методы обязательно нужно переопределить в наследнике SQLiteOpenHelper (onCreate, onUpgrade)? Для чего служит метод onDowngrade?
50. Как получить доступ к базе данных для чтения и записи?
51. Какие два основных способа выполнения SQL-запросов к SQLite?
52. Что представляет собой класс Cursor, и какие методы позволяют перемещаться по результатам запроса?
53. Какова роль ContentValues при вставке или обновлении данных в SQLite?
54. Что такое SQLiteDatabase.insert(), update(), delete() и как они отличаются от написания прямых SQL-команд?
55. Как выполнить сложный запрос с JOIN и агрегацией (например, подсчет суммы) в SQLite на Android?
56. Для чего используется класс SimpleCursorAdapter, и какие у него недостатки по сравнению с CursorAdapter?
57. Как RecyclerView взаимодействует с данными из SQLite через CursorAdapter или Room?
58. Что такое ORM (Object-Relational Mapping) и как библиотека Room упрощает работу с SQLite по сравнению с прямыми запросами?
59. Как обеспечить безопасность данных при работе с SQLite (избежать SQL-инъекций)?
60. Перечислите три основные категории сенсоров в Android (движение, положение, окружающая среда) и приведите примеры для каждой.
61. Как получить список всех доступных сенсоров на устройстве с помощью SensorManager.getSensorList()?
62. Как проверить, есть ли на устройстве конкретный датчик (например, акселерометр или датчик приближения), и что вернёт getDefaultSensor() при его отсутствии?
63. Какие шаги необходимо выполнить, чтобы зарегистрировать слушатель событий сенсора (SensorEventListener)?
64. Почему не рекомендуется использовать максимальную частоту для всех сенсоров? Какие проблемы с батареей и производительностью это вызывает?
65. Как правильно отключать сенсоры в методах жизненного цикла Activity (onPause / onResume), чтобы избежать разряда батареи и утечек памяти?
66. Какие данные предоставляет акселерометр (три оси), и как их использовать для определения тряски устройства (shake)?
67. В чём разница между акселерометром и гироскопом? Для каких задач применяется каждый из них?
68. Как получить данные от датчика магнитного поля (магнитометра) и объединить их с акселерометром для определения ориентации в пространстве?
69. Что такое датчик приближения (proximity sensor), и как его типично используют?
70. Как работать с датчиком освещённости (light sensor) и динамически изменять яркость интерфейса или тему приложения?

Модуль «Контрольная работа» (100 баллов):

Темы заданий контрольной работы:

1. Эволюция мобильных платформ и инструментов разработки.
2. Элементы управления пользовательского интерфейса мобильных приложений.
3. Навигация и многоконность в Android: от Activity до диалоговых окон.
4. Классификация сенсоров по принципу измерения и назначению.

Для выполнения контрольной работы обучающиеся получают в библиотеке печатный экземпляр или скачивают в ЛК электронное издание учебно-методического пособия: «Методические указания по выполнению контрольной работы по дисциплине «Программирование мобильных устройств».

Модуль посвящен выполнению задания на контрольную работу. Максимальное количество баллов за выполненную и защищенную контрольную работу составляет 100.

Контрольная работа выполняется в процессе освоения компетенции (ПК-2: Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности), а итоговая оценка за её защиту характеризует заключительный этап их освоения.

Критерии оценивания контрольной работы приведены в таблице.

Таблица – Критерии оценивания контрольной работы

№ п/п	Оцениваемые параметры	Оценка в баллах
1.	Корректность реализации программного кода: – Программа компилируется и выполняется без ошибок. – Все функции работают без ошибок, включая краевые случаи. – Переходы между экранами корректны, стек кнопки «Назад» работает логично, диалоги не теряются. – Данные сенсоров получаются корректно, частота обновления оптимальна.	40 10 10 10 10
2.	Эффективность и выбор элементов управления: – Соответствие типа элемента типу данных. – Элементы перестраиваются при повороте, клавиатура не перекрывает поля ввода, все видно. – Обратная связь и состояние элементов реализованы полностью. – Есть подтверждение деструктивных действий, валидация ввода с подсказками.	30 8 8 6 8
3.	Оформление результатов контрольной работы: – контрольная работа оформлена в полном соответствии с ЕСКД; – контрольная работа оформлена с незначительными отклонениями от ЕСКД; – оформление контрольной работы не соответствует требованиям ЕСКД.	15 15 10 5
4.	Качество ответов на вопросы в процессе защиты контрольной работы: – Студент может объяснить выбор мобильной платформы и языка разработки. – Отвечает на вопросы о сложности операций, преимуществах/недостатках реализованных структур. – Вносит изменения в код по требованию преподавателя. – Допускает грубые ошибки при ответе или не может четко ответить на	15 5 5 5 0

№ п/п	Оцениваемые параметры	Оценка в баллах
	вопросы.	

Студент, набравший по контрольной работе:

- от 0 до 40 баллов, получает оценку «неудовлетворительно»;
- от 41 до 60 баллов, получает оценку «удовлетворительно»;
- от 61 до 80 баллов, получает оценку «хорошо»;
- от 81 до 100 баллов, получает оценку «отлично».

2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

2.1 Порядок и методика проведения текущего контроля и промежуточной аттестации

Предусмотрены следующие виды контроля:

- текущий контроль по каждому модулю – в форме написания контрольной работы для оценки теоретических знаний;
- текущий контроль по каждому модулю – в форме отчетов по лабораторным работам и практическим занятиям для оценки практических навыков;
- промежуточная аттестация по дисциплине – в форме зачета.

Текущий контроль успеваемости в форме защиты лабораторных и практических проводится по окончании каждой лабораторной работы (практического занятия) в форме индивидуального собеседования по выполненным исследованиям или расчетам. Проводится с целью контроля закрепления теоретической части материала и степени отработки студентом практических навыков исследования на аппаратуре.

Контрольные работы выполняются в виде короткого письменного ответа на один вопрос, изученный на предыдущей лекции в начале каждой последующей лекции. Ответ на вопрос дается в течение 5-10 минут. Таким образом, после лекционного курса каждого модуля формируется общая оценка за теоретические знания.

С целью повышения качества обучения за счет побуждения студентов к активной текущей учебной работе, четкого и оперативного контроля всего хода учебного процесса, снижения роли случайных и субъективных факторов при оценивании учебной деятельности студентов в образовательном процессе реализована модульно-рейтинговая система.

Правила ее использования прописаны в «Положении об МРС».

Набранные обучающимся в течение текущего контроля баллы могут быть переведены в оценку (по дисциплине предусмотрен зачет):

- «не зачтено» - от 0 до 40 от максимального количества баллов;
- «зачтено» - от 41 до 100 максимального количества баллов.

Соотношения максимального количества баллов, полученных студентом по блокам модулей, показаны в Таблицах.

Таблица - Распределение баллов по блокам модулей дисциплины «Программирование мобильных устройств» для очной формы обучения (ОФО)

Модуль	Всего баллов (Максимальное значение)	Теоретический блок (Распределение баллов по средствам оценивания)	Практический блок (Распределение баллов по средствам оценивания)
Курс 3, Семестр 6			

Модуль 1	50	10	40=20+20
Модуль 2	50	10	40=20+20
Зачет	100	100	

Таблица - Распределение баллов по блокам модулей дисциплины «Программирование мобильных устройств» для заочной формы обучения (ЗФО)

Модуль	Всего баллов (Максимальное значение)	Теоретический блок (Распределение баллов по средствам оценивания)	Практический блок (Распределение баллов по средствам оценивания)
Курсы 3,4			
Модуль 1	50	20	30=20+10
Модуль 2	50	20	30=10+20
Зачет	100	100	

Как правило, теоретический блок оценивается по результатам контрольной работы. Практический блок оценивается по результатам выполнения лабораторных работ и заданий на практических занятиях.

На зачете производится оценка тех компетенций, которые должны быть в той или иной форме освоены в процессе изучения. Рекомендуется формировать вопросы в билетах к зачету таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины.

2.2 Методика проведения зачета в группах очной и заочной форм

Зачет по дисциплине «Программирование мобильных устройств» у студентов очной и заочной форм обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 2-3 вопроса, скомпонованные таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины. Количество билетов должно быть не менее числа студентов в группе. Студенты готовят письменные ответы на вопросы.

3 Промежуточная аттестация

Промежуточная аттестация проводится в форме зачета:

- для очной формы обучения (ОФО) на третьем курсе в 6 семестре;
- для заочной формы обучения (ЗФО) на четвертом курсе, сессия 2.

Зачет проводится по расписанию экзаменационной сессии в письменном виде. Количество вопросов в билете к зачету – 2. Проверка ответов и объявление результатов производится в день зачета. Результаты аттестации заносятся в зачетную ведомость и зачетную книжку студента.

Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

Образец билета к зачету

	МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»	Утверждаю Зав. кафедрой «ИВТ» _____ Соколов С.В. « ____ » _____ 20 ____ г.
Направление подготовки: 09.03.01. Информатика и вычислительная техника Курсы: 3 Дисциплина: Программирование мобильных устройств		
Билет №1 1. Какие интегрированные среды разработки (IDE) являются основными при создании нативных приложений для iOS и Android? 2. Какие механизмы безопасности (шифрование, песочница, SELinux) используются в Android для защиты данных приложения? Приведите примеры. ст. преподаватель кафедры ИВТ _____ Осмоловская Н.С. « ____ » _____ 20 ____ г.		

Один комплект отпечатанных билетов к зачету, подписанных преподавателем кафедры и утвержденных заведующим кафедрой хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

4 Тестовые задания для проведения оценки сформированности компетенций (ПК-2)

Данная дисциплина формирует у обучающегося ПК-2 компетенцию, объективно оценить которую можно при ответе студента на тест или при выполнении практического или лабораторного задания.

Один комплект тестовых вопросов с указанием правильных ответов хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

Тестовые задания позволяют оценить уровень сформированности компетенции ПК-2.

Задания распределены по блокам в соответствии с уровнем сложности. Каждый блок содержит номер задания, и текст задания.

Базовый уровень содержит примерно 50 % заданий.

Он формируется из заданий с выбором одного или нескольких верных ответов из предложенных или воспроизведения фактического материала (терминология, факты, классификация, параметры и др.);

Повышенный уровень в среднем составляет 35 % заданий.

В нем используются задания на сопоставление, сравнение, установление последовательности;

Высокий уровень в среднем составляют – 15 % заданий.

В задании используются: решения нетиповых задач, алгоритмы, доказательства, задания с развернутым ответом – определения, перечисление, изображение схемы, структуры и др.

Количество заданий по каждой компетенции равно 20. Всего 40 заданий.

Ключи к ответам представлены в отдельной таблице.

ПК-2: Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности

БЛОК А (базовый уровень) – задание с выбором одного или нескольких верных ответов из предложенных.

Инструкция по тестам Блока А: Прочитайте текст и выберите один или несколько правильных ответов.

№ задания	Тексты заданий
1	Какая операционная система была одной из первых массовых мобильных ОС с графическим интерфейсом и стилусом? 1. iOS 2. Symbian 3. Palm OS 4. Android
2	Какая современная мобильная ОС, кроме Android и iOS, разработана компанией Huawei? 1. Tizen 2. HarmonyOS 3. KaiOS 4. Ubuntu Touch
3	Что из перечисленного является современной средой разработки для мобильных приложений под Android? 1. Xcode 2. Visual Studio Code (без плагинов) 3. Android Studio 4. Eclipse (старая версия)
4	Что значит «песочница» в мобильной операционной системе? 1. Игрушка для детей 2. Изоляция приложений друг от друга для безопасности 3. Режим полёта 4. Экран блокировки
5	Как сделать так, чтобы элемент интерфейса (например, кнопка) растянулась на всю ширину экрана? 1. match_parent 2. wrap_content 3. fix_size 4. max_width
6	Где лучше отключать датчики (например, перестать слушать гироскоп), чтобы экономить заряд батареи? 1. При запуске приложения 2. При нажатии на кнопку 3. В конце работы приложения 4. Когда экран уходит в фон (onPause)
7	Какой датчик в телефоне определяет, как быстро вращается устройство? 1. Датчик света 2. Гироскоп 3. Акселерометр 4. Датчик приближения
8	Что такое стек экранов (обратный стек)? 1. Очередь сообщений

№ задания	Тексты заданий
	2. Список открытых экранов для кнопки «Назад» 3. Набор картинок 4. График загрузки процессора
9	Как называется момент жизненного цикла экрана, когда он становится виден пользователю (но ещё не полностью готов к действиям)? 1. onCreate 2. onPause 3. onStop 4. onStart
10	Для чего нужен адаптер при показе списка данных (например, списка контактов)? 1. Чтобы подключаться к интернету 2. Чтобы связывать данные с элементами списка 3. Чтобы устанавливать картинки 4. Чтобы закрывать приложение

БЛОК Б (повышенный уровень) – задание закрытого типа на установление соответствия
Инструкция: Прочитайте текст и установите соответствие.

№ задания	Тексты заданий
11	Сопоставьте мобильную операционную систему с её ключевой характеристикой 1. Symbian 2. Palm OS 3. Windows Mobile 4. Android А. Первая ОС от Google с открытым исходным кодом Б. Использовала стилус, популярна на КПК в 90-х В. Долгое время лидер в смартфонах Nokia Г. Интерфейс как у настольной Windows, кнопка «Пуск»
12	Сопоставьте компонент Android-приложения с его назначением 1. Activity 2. Service 3. BroadcastReceiver 4. ContentProvider А. Выполнение фоновых задач без интерфейса Б. Представление одного экрана (окна) В. Хранение данных и предоставление доступа к ним Г. Реагирование на системные события (зарядка, SMS)
13	Сопоставьте тип сенсора с тем, что он измеряет 1. Акселерометр 2. Гироскоп 3. Магнитометр 4. Датчик освещённости А. Угловая скорость вращения Б. Уровень освещённости вокруг В. Ускорение (включая гравитацию) Г. Направление магнитного поля (компас)
14	Сопоставьте способ хранения данных в Android с типом данных (назначением) 1. SharedPreferences 2. SQLite

№ задания	Тексты заданий
	3. Внутреннее хранилище (файлы) 4. Внешнее хранилище (SD-карта) А. Реляционная база данных (таблицы, SQL) Б. Небольшие настройки типа «ключ-значение» В. Скрытые файлы приложения, доступные только ему Г. Общедоступные файлы (фото, документы)

БЛОК Б (повышенный уровень) – задание закрытого типа на установление последовательности.

Инструкция: Прочитайте текст и установите правильную последовательность.

№ задания	Тексты заданий
15	Расположите следующие мобильные ОС в порядке их появления на рынке (от самой ранней к самой поздней): 1. Palm OS 2. Android 3. Symbia 4. iOS
16	Установите последовательность передачи и обработки события касания (touch event) от оборудования до приложения: 1. Вызов onTouchEvent() у View 2. Аппаратное прерывание от сенсорного экрана 3. Передача события в конкретный View (целевой элемент) 4. Вызов зарегистрированного onTouchListener(), если есть
17	Расположите шаги запроса разрешения (runtime permission) в Android в правильной последовательности (начиная с самого первого): 1. Вызвать requestPermissions() 2. Проверить наличие разрешения через checkSelfPermission() 3. Обработать результат в onRequestPermissionsResult() 4. Проверить, нужно ли объяснение через shouldShowRequestPermissionRationale()

БЛОК В (высокий уровень) – задание открытого типа с развернутым ответом.

Инструкция: Прочитайте текст и запишите обоснованный ответ.

№ задания	Тексты заданий
18	Энергопотребление – это количество электрической энергии (в миллиампер-часах, мА·ч), которое приложение потребляет от аккумулятора за единицу времени. Чем выше энергопотребление, тем _____ устройство.
19	Устройство, которое обнаруживает какое-либо воздействие из окружающей среды, реагирует на это воздействие и преобразует его в сигнал, удобный для дальнейшей обработки (чаще всего в электрический), называется _____.
20	Механизм в Android для передачи сообщений между компонентами приложения (Activity, Service, BroadcastReceiver) или между разными приложениями – это...

Система оценивания выполнения тестовых заданий по компетенции ПК-2.

Номер задания	Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания/характеристика правильности ответа)
Задания 1-10	Задания с выбором одного или нескольких верных ответов из предложенных считается верным, если правильно указана цифра или цифры, означающие верные ответы.	Совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.
Задания 11-14	Задания закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Полное совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.
Задания 15-17	Задания закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр	Полное совпадение с верным ответом оценивается 1 баллом; если допущены ошибки или ответ отсутствует - 0 баллов.
Задания 18-20	Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте.	Полный правильный ответ на задание оценивается 3 баллами; если допущена одна ошибка/неточность/ответ правильный, но не полный - 1 балл, если допущено более одной ошибки/ответ неправильный/ ответ отсутствует- 0 баллов.

Методика проведения тестирования

Максимальное время выполнения теста - 60 минут.

Количество тестовых заданий в тесте не менее 60.

Для проведения контроля остаточных знаний компьютер формирует в случайном порядке 60 тестовых заданий.

При прохождении тестового контроля знаний студент должен выбрать один или несколько правильных ответов в каждом тестовом задании.

В случае правильного ответа на 49 и более тестовых заданий (81-100% правильных ответов) студент подтверждает освоение компетенций с уровнем «отлично». При ответе на 48-37 тестовых заданий (61-80% правильных ответов) студент подтверждает освоение компетенций с уровнем «хорошо». При ответе на 36-25 тестовых заданий (41-60% правильных ответов) студент подтверждает освоение компетенций с уровнем «удовлетворительно». При ответе менее чем на 25 тестовых заданий (0-40% правильных ответов) студент не подтверждает необходимый уровень знаний и оценивается «неудовлетворительно».

Методические рекомендации по освоению дисциплины

1. Методические рекомендации преподавателю

Перед началом изучения дисциплины преподаватель должен ознакомить студентов с рабочей программой и оценочными материалами по дисциплине, с видами учебной и самостоятельной работы, перечнем литературы и интернет-ресурсов, с формами текущей и промежуточной аттестации, с критериями оценки качества знаний для итоговой оценки по дисциплине.

При проведении лекций, преподаватель:

- 1) формулирует тему и цель занятия, объявляет учебные вопросы;
- 2) излагает основные теоретические положения;
- 3) с помощью технических средств обучения и/или под запись дает определения основных понятий, расчетных формул;
- 4) проводит примеры из отечественного и зарубежного опыта, дает текущие статистические данные для наглядного и образного представления изучаемого материала;
- 5) в конце занятия выдает вопросы для самостоятельного изучения.

На занятиях лабораторного цикла следует обратить внимание на соответствие выбираемых студентом средств выполнения решаемым в работе задачам.

Каждая лабораторная работа должна быть оформлена и защищена в соответствии с требованиями. Защита производится после оформления отчета по работе.

Во время выполнения заданий в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Для оценки полученных знаний и освоения учебного материала по каждому разделу и в целом по дисциплине преподаватель использует формы текущего контроля и промежуточной аттестации знаний обучающихся.

2. Методические рекомендации студентам

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с рабочей программой, оценочными материалами, учебной, научной и методической литературой, имеющейся в ЭИОС СКФ МТУСИ,

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

2.1 Методические указания для обучающихся по лекционным занятиям

Важно сосредоточить внимание на содержании лекции. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине.

Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на

которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы.

Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

2.2 Методические указания для обучающихся по подготовке к лабораторным работам

Лабораторные занятия по любой учебной дисциплине проводятся в соответствии с рабочей программой этой учебной дисциплины в установленные расписанием часы.

Инструктаж по технике безопасности, правилам пожарной безопасности и правилам внутреннего распорядка в лаборатории проводится преподавателем на первом лабораторном занятии. На этом же занятии студентам сообщаются:

- программа всего предстоящего лабораторного цикла;
- условия взаимодействия студентов с преподавателем в процессе выполнения лабораторных работ;
- условия контроля самостоятельной работы студентов, включая правила оформления отчетов по лабораторным работам и их последующей защиты;
- другая необходимая информация.

Организация лабораторных работ включает:

- самостоятельную внеаудиторную подготовку студента к выполнению каждой отдельной лабораторной работы в соответствии с ее программой в рамках часов, выделенных на самостоятельную проработку материала в программе дисциплины;
- входной контроль преподавателем степени подготовленности каждого студента к выполнению лабораторных работ;
- выполнение программы лабораторных работ в полном объеме;
- оформление отчета и его защиту каждым студентом в установленные сроки;
- формирование преподавателем рейтингов каждого из студентов по результатам выполнения и защиты им отдельных лабораторных работ и их циклов (используется Модульно-рейтинговая система).

Оценка качества выполнения лабораторных работ каждым студентом производится преподавателем отдельно за подготовку к работе, ее выполнение и защиту.

2.3 Методические указания для обучающихся по подготовке к практическим занятиям

Целью практических занятий является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации:

- изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т.д.;
- при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы.

После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.

Методические указания по выполнению лабораторных и практических работ по дисциплине «Программирование мобильных устройств» для студентов очной формы обучения, размещенные на сайте СКФ МТУСИ в разделе «Методические материалы».

2.4 Методические указания для обучающихся по подготовке к контрольной работе

В ходе выполнения контрольной работы студент решает следующие задачи:

1. Эволюция мобильных платформ и инструментов разработки.
2. Элементы управления пользовательского интерфейса мобильных приложений.
3. Навигация и многоконность в Android: от Activity до диалоговых окон.
4. Классификация сенсоров по принципу измерения и назначению.

Порядок выполнения контрольной приведен в методических материалах по её выполнению. Критерии оценки контрольной работы приведены в оценочных материалах дисциплины.

3. Методические рекомендации для обучающихся по самостоятельной работе

Самостоятельная работа студентов является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих курсов (систем), а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам.

Самостоятельная работа во внеаудиторное время включает в себя:

- повторение лекционного материала;
- подготовки к лабораторным работам (практическим занятиям);
- изучения учебной и научной литературы;
- решения задач, выданных на практических занятиях;
- подготовки к тестированию;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях.
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов по отдельным вопросам изучаемой дисциплины.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж (консультацию) с определением цели задания, его содержания, сроков выполнения, основных требований к результатам работы, критериев оценки, форм контроля и перечня источников и литературы.

Обычно постановку задачи обучаемым на проведение самостоятельной работы преподаватель осуществляет на одном из занятий, предшествующем данному.

Методику самостоятельной работы все обучаемые выбирают индивидуально.