

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ  
«ДОМ ЮНОШЕСКОГО ТЕХНИЧЕСКОГО ТВОРЧЕСТВА»  
ЦЕНТР ЦИФРОВОГО ОБРАЗОВАНИЯ «IT-КУБ» Г. САТКА

ПРИНЯТО на заседании  
педагогического совета  
ГБУ ДО «ДЮТТ Челябинской области»  
протокол № 135 от 15 июля 2023 г.

УТВЕРЖДАЮ:  
Директор ГБУ ДО «ДЮТТ  
Челябинской области»  
В. Н. Халамов  
Приказ № 35 от «28» июня 2023 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
«Разработка на языке Kotlin для платформы Android– Продвинутый уровень»

Направленность: техническая  
Уровень программы: продвинутый  
Срок освоения программы: 1 год  
Возрастная категория обучающихся: 14-18 лет

Автор-составитель:  
Гайнанов Максим Вячеславович,  
педагог дополнительного образования

г. Сатка  
2023

## Содержание

|   |    |
|---|----|
| Раздел 1. Комплекс основных характеристик программы           |    |
| 1.1 Пояснительная записка .....                               | 3  |
| 1.2.Сведения о программе.....                                 | 5  |
| 1.3. Цель и задачи программы .....                            | 7  |
| 1.4. Содержание программы .....                               | 8  |
| 1.5. Учебный план.....  | 31 |
| 1.6. Планируемые результаты .....                             | 34 |
| Раздел 2. Комплекс организационно-педагогических условий..... | 35 |
| 2.1 календарный учебный график .....                          | 35 |
| 2.2 Условия реализации программы .....                        | 35 |
| 2.3. Формы аттестации .....                                   | 35 |
| 2.4. Оценочные материалы .....                                | 36 |
| 2.5 Методические материалы .....                              | 39 |
| 2.6 Воспитательные компоненты .....                           | 40 |
| 2.7. Информационные ресурсы и литература .....                | 42 |
| Приложение.....   | 43 |
| Приложение 1.....   | 43 |
| Приложение 2.....   | 44 |
| Приложение 3.....   | 45 |

## Раздел 1. Комплекс основных характеристик программы

### 1.1 Пояснительная записка

Настоящее Положение о порядке разработки и реализации дополнительных общеобразовательных программ ГБУ ДО «ДОТТ Челябинской области» разработано на основании:

Программа разработана на основании:

- Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Федерального закона от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в Российской Федерации»;
- Распоряжения Правительства РФ от 12.11.2020 № 2945-р «Об утверждении плана мероприятий по реализации в 2021 — 2025 г. г. Стратегии развития воспитания в Российской Федерации на период до 2025 года»;
- Концепции развития дополнительного образования детей до 2030 года и плана мероприятий по ее реализации, утвержденной распоряжением Правительства РФ от 31.03.2022 № 678-р;
- Указа Президента Российской Федерации «Стратегия научно- технологического развития Российской Федерации» (редакция от 15.03.2021г. №143);
- Постановления Главного государственного санитарного врача РФ от 28.09.2020 N. 28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Паспорта приоритетного проекта «Доступное дополнительное образование для детей», утвержденного президиумом Совета при Президенте РФ по стратегическому развитию и приоритетным проектам 30 ноября 2016 г.;
- Приказ Министерства просвещения Российской Федерации от 27 июля 2022г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Методических рекомендаций по проектированию дополнительных общеразвивающих программ, разработанных Министерством образования и науки России совместно с ГАОУ ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования», АНО дополнительного профессионального образования «Открытое образование»;
- Письмо Минобрнауки РФ от 18.11.2015 г. № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
- Письмо Минобрнауки России от 29 марта 2016 г. № ВК-641/09 «О направлении методических рекомендаций (вместе с «Методическими рекомендациями по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей)»;
- Распоряжение Правительства ЧО № 901-рп от 20.09.2022 г. «Об утверждении регионального плана мероприятий на 2022 – 2024 годы по реализации Концепции развития дополнительного образования детей до 2030 года»;
- Приказ Министерства труда и социальной защиты РФ № 652-н от 21.09.2021 г «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;

- Приказ Минпросвещения России от 03.09.2019 N 467 (ред. от 21.04.2023) «Об утверждении Целевой модели развития региональных систем дополнительного образования детей» (Зарегистрировано в Минюсте России 06.12.2019 N 56722);
- Закона Челябинской области от 29.08.2013 № 515-ЗО «Об образовании в Челябинской области»;
- Устава ГБУ ДО «Дом юношеского технического творчества Челябинской области».

**Актуальность программы.** Программа разработки на языке Kotlin для платформы Android является актуальной по нескольким причинам:

1. Kotlin - это язык программирования, разработанный JetBrains, который стал "официальным" языком разработки Android в 2017 году. Он предоставляет разработчикам множество преимуществ по сравнению с Java, таких как короткий и лаконичный синтаксис, поддержка функционального программирования, улучшенная безопасность типов и многое другое. Kotlin позволяет более эффективно и быстро писать код, что является основным преимуществом в разработке мобильных приложений.

2. Огромное сообщество разработчиков Kotlin на платформе Android. За несколько лет с момента принятия Kotlin в качестве официального языка для Android, множество разработчиков перешли на Kotlin и создали большое сообщество энтузиастов, которые активно делятся опытом, решениями и разработчиками библиотек на Kotlin. Это делает процесс разработки более удобным и эффективным.

3. Поддержка Kotlin во множестве инструментов и библиотек. Kotlin полностью совместим с Java, поэтому практически все существующие инструменты и библиотеки на Java могут быть использованы и в проектах на Kotlin. Кроме того, многие инструменты разработки, такие как Android Studio, поддерживают Kotlin "из коробки" и предоставляют средства для автоматического преобразования кода Java в Kotlin.

4. Kotlin продолжает развиваться и улучшаться. Kotlin активно разрабатывается и поддерживается компанией JetBrains. В последние годы он получил множество новых функций и дополнений, и его функциональность и производительность постоянно улучшаются. Это означает, что разработчики могут быть уверены в том, что язык Kotlin будет развиваться и соответствовать современным требованиям разработки мобильных приложений.

В целом, Разработка на языке Kotlin для платформы Android– Продвинутый уровень является актуальной и востребованной, и участие в программе позволит разработчикам освоить современные технологии и инструменты для создания качественных мобильных приложений.

**Педагогическая целесообразность.** Программа "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" имеет ряд педагогических преимуществ и целесообразна по нескольким причинам:

1. Заинтересованность обучающихся: Мобильные приложения - это тема, которая актуальна и интересна для большинства современных обучающихся. Разработка мобильных приложений и игр является популярной и захватывающей деятельностью, которая может вызывать положительные эмоции и мотивацию для изучения и углубления в эту область.

2. Практическое применение знаний: Программа "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" ориентирована на практическую разработку мобильных проектов. Обучающиеся имеют возможность применить полученные знания и навыки в создании собственных игр и приложений. Это помогает им понять, как применять теоретические знания на практике и улучшить свои навыки программирования.

3. Развитие творческого мышления: Разработка на языке Kotlin для платформы Android– Продвинутый уровень требует творческого подхода и воображения для создания уникальных и интересных проектов. В процессе обучения обучающиеся будут сталкиваться с различными задачами и проблемами, которые потребуют от них поиска нестандартных решений и развития творческого мышления.

4. Командная работа: Разработка мобильного приложения часто требует командной работы. Программа позволяет учиться работать в команде, обмениваться знаниями и навыками, а также эффективно совместно решать задачи и достигать общих целей.

5. Подготовка к профессиональной деятельности: Программа "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" предоставляет базовые знания и навыки в области разработки мобильных проектов. Это может стать отличной отправной точкой для дальнейшей профессиональной деятельности в области мобильных технологий и программирования.

В целом, программа "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" имеет педагогическую целесообразность, так как она сочетает в себе актуальность темы, практическое применение знаний и развитие ключевых навыков, необходимых для успешной работы в области мобильной разработки.

**Отличительные особенности** программы заключаются в том, что она учитывает новые технологические уклады, которые требуют новый способ мышления и тесного взаимодействия при постоянном повышении уровня междисциплинарности проектов, а также использует новые формы диагностики и подведения итогов реализации программы, выполняемые в формате защиты проектов и участия во Всероссийском конкурсе мобильных приложений.

**Адресат программы** – школьники 14-18 лет, проявляющие интерес к информационным технологиям, и имеющие фундаментальные знания по математике, информатике и английскому языку. На обучение принимаются все желающие, без предварительной подготовки, по заявлению родителей или лиц, их заменяющих. Набор в объединение производится по желанию обучающихся и их родителей.

Особенности развития детей среднего школьного возраста: формируется самосознание – представление о себе самом, самооценивание умственных, моральных, волевых качеств. Происходит соотношение себя с идеалом, появляется возможность самовоспитания. Возрастает волевая регуляция. Ведущая деятельность – учебно-профессиональная. Стремление приобрести профессию – основной мотив познавательной деятельности. Возрастает концентрация внимания, объем памяти, сформировалось абстрактно-логическое мышление. Появляется умение самостоятельно разбираться в сложных вопросах. Формируется собственное мировоззрение, как целостная система взглядов, знаний, убеждений, своей жизненной философии. Стремление к самоуправлению, стремление заново осмыслить все окружающее, происходит жизненное определение человека.

**Срок реализации и объем программы** определяется содержанием программы и составляет 1 год (144 академических часа).

**Направленность:** техническая.

**Язык реализации программы:** русский.

**Особенности реализации программы:** модульный принцип.

**Уровень освоения программы:** базовый.

**Форма обучения** – очная, с возможностью применения дистанционных технологий.

**Формы организации:** в группе 12 человек.

**Режим занятий:** 4 академических часа в неделю.

1 раз - 2 часа (академический час – 45 мин.). Через каждые 45 минут занятия следует 15-минутный перерыв, согласно требованиям, СанПиН.

**Форма организации занятий:** групповое, индивидуально-групповое.

**Методы обучения:** наглядный, практический, объяснительно- иллюстративный.

## 1.2. Сведения о программе

|                    |   |
|--------------------|---|
| Название программы | Разработка на языке Kotlin для платформы Android– Продвинутый уровень (14-18 лет) |
|--------------------|---|

|  |  |
|--|--|
| Возраст обучающихся  | 14-18 лет  |
| Длительность программы (в часах)                                     | 144 часа   |
| Количество занятий в неделю  | 2 часа 2 раза в неделю (академический час – 45 мин)  |
| Цель, задачи   | Цель: развитие интереса обучающихся к информационным и телекоммуникационным технологиям; реализация их творческих идей в области мобильной разработки.<br>Задачи направлены на достижение цели и включают в себя обучающие, развивающие, воспитательные.   |
| Краткое описание программы   | Программа "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" предлагает изучение и практическое применение языка программирования Kotlin для создания мобильных приложений для платформы Android.<br>В ходе программы научатся работать над практическими проектами, решать задачи и создавать свои собственные мобильные приложения на платформе Android с использованием языка Kotlin. Они также будут ознакомлены с современными инструментами разработки и методологиями разработки мобильных приложений.<br>Цель программы - обеспечить участникам навыки и практический опыт разработки приложений на языке Kotlin для платформы Android, чтобы они могли успешно создавать и поддерживать высококачественные мобильные приложения. |
| Первичные знания, необходимые для освоения программы                 | Базовые знания, полученные при изучении школьной программы.  |
| Результат освоения   | приобретение навыков и знаний для самостоятельной разработки мобильных приложений с использованием языка Kotlin. Они смогут создавать пользовательские интерфейсы, работать с данными, интегрировать сторонние сервисы и библиотеки, оптимизировать приложения.  |
| Перечень соревнований, в которых учащиеся смогут принять участие     | Конференция «Юные техники – инженеры»<br>Фестиваль по IT-технологиям<br>Ярмарка проектов<br>“Играойя” (г.Сатка)<br>Всероссийский конкурс мобильных приложений  |
| Перечень основного оборудования, необходимого для освоения программы | Планшет, ноутбук, WEB-камера, наушники, моноблочное интерактивное устройство, напольная мобильная стойка для интерактивных досок или универсальное настенное крепление, сетевой фильтр   |

|  |   |
|--|---|
| <p>Преимущества данной программы (отличия от других подобных курсов)</p> | <p>Данная программа формирует профессиональные компетенции, которые позволят обучающимся в будущем успешно конкурировать в области мобильной разработки.</p> <p>Программа плотно связана с массовыми мероприятиями в научно-технической сфере для обучающихся (турнирами, состязаниями, конференциями), что позволяет, не выходя за рамки учебного процесса, принимать активное участие в конкурсах различного уровня</p> |
|--|---|

### 1.3. Цель и задачи программы

**Целью** программы является развитие интереса обучающихся к информационным, телекоммуникационным технологиям и реализация их творческих идей в области мобильной разработки.

#### **Задачи:**

##### Личностные:

- Формирование умения самостоятельной деятельности.
- Формирование умения работать в команде.
- Формирование коммуникативных навыков.
- Формирование навыков анализа и самоанализа.
- Формирование целеустремленности и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

##### Предметные:

- Формирование представления о программном обеспечении и сетевом оборудовании организаций.
- Формирование представления об устройстве персонального компьютера и принципе его работы.
- Формирование представления о принципах работы сетей.
- Формирование умений по работе с различным программным обеспечением.

##### Метапредметные:

- Формирование умения ориентироваться в системе знаний.
- Формирование умения выбирать наиболее эффективные способы решения задач на компьютере в зависимости от конкретных условий.
- Формирование приемов проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат своей деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы.
- Формирование умения распределения времени.
- Формирование умений успешной самопрезентации.

## 1.4. Содержание программы

### Модуль 1. Архитектурные подходы в Android

#### 1.1. Модель разработки приложений на платформе Android

##### Теория:

- Введение в разработку приложений на платформе Android и основы языка Kotlin.
- Изучение модели разработки приложений на платформе Android, такой как Model-View-Controller (MVC), Model-View-Presenter (MVP) и Model-View-ViewModel (MVVM).
- Понимание основных принципов каждой модели разработки и их применение в контексте платформы Android.
- Ознакомление с различными паттернами программирования и архитектурами, используемыми в разработке на платформе Android, такими как Clean Architecture, SOLID принципы и другие.
- Изучение основных компонентов приложений на платформе Android и их связи в каждой модели разработки, таких как активности, фрагменты, сервисы и другие.
- Обзор локального хранения данных на платформе Android, такого как SharedPreferences, SQLite и Room Persistence Library.
- Изучение подходов к тестированию приложений на платформе Android с использованием модели разработки.

##### Практика:

- Создание нового проекта на платформе Android с использованием языка Kotlin и выбор модели разработки для приложения.
- Разделение кода на соответствующие компоненты (модель, представление, контроллер или вью-модель) в соответствии с выбранной моделью разработки.
- Реализация функциональности приложения в каждом компоненте, соблюдая правила и принципы выбранной модели разработки.
- Тестирование и отладка приложения с использованием инструментов и подходов, соответствующих выбранной модели разработки.
- Оптимизация и улучшение функциональности и структуры кода в приложении в соответствии с выбранной моделью разработки.

#### 1.2. Изучение основ архитектурного подхода MVVM

##### Теория:

- Введение в архитектурный подход Model-View-ViewModel (MVVM) и его применение при разработке приложений на платформе Android.
- Ознакомление с основными компонентами MVVM: модель (Model), представление (View) и модель представления (ViewModel).
- Понимание основных принципов и преимуществ MVVM, таких как разделение ответственностей между компонентами, упрощение тестирования и обновления пользовательского интерфейса.
- Изучение примеров использования MVVM в различных сценариях разработки приложений на платформе Android.

##### Практика:

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разделение компонентов приложения согласно архитектурному подходу MVVM: создание модели, представления и модели представления.
- Настройка связей и взаимодействия между компонентами MVVM, например, с помощью привязки данных (data binding) и обработчиков событий.
- Разработка функциональности приложения с использованием MVVM, например, получение и обработка данных в модели представления, отображение данных в представлении.
- Тестирование и отладка приложения на основе архитектурного подхода MVVM.
- Оптимизация и улучшение функциональности и производительности приложения с использованием принципов MVVM, например, оптимизация работы с данными, добавление асинхронных операций и другие.



### **1.3.Создание архитектуры приложения на основе MVVM**

#### **Теория:**

- Ознакомление с основами языка Kotlin и его использованием для разработки на платформе Android.
- Изучение основ архитектуры MVVM (Model-View-ViewModel) и ее применение при разработке приложений на платформе Android.
- Понимание роли каждого компонента архитектуры MVVM: модели (Model), представления (View) и модели представления (ViewModel).
- Изучение основных принципов связи между компонентами архитектуры MVVM и методов обновления данных.
- Обзор инструментов и библиотек, которые можно использовать для реализации архитектуры MVVM в приложении на платформе Android, таких как Jetpack's ViewModel, LiveData, Data Binding и другие.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка модели (Model) приложения, которая будет отвечать за получение и обработку данных.
- Создание представления (View) приложения, которое будет отображать пользовательский интерфейс и отправлять запросы на обновление данных.
- Разработка моделей представления (ViewModel), которые будут содержать бизнес-логику, обрабатывать данные из моделей и предоставлять обновленные данные для представления.
- Связывание компонентов архитектуры MVVM с использованием инструментов и библиотек, таких как Jetpack's ViewModel, LiveData и Data Binding.
- Тестирование и отладка приложения с реализованной архитектурой MVVM.
- Оптимизация и улучшение функциональности и производительности приложения с использованием принципов и инструментов архитектуры MVVM.

## **Модуль 2. Потоки данных**

### **2.1.Работа с потоками данных и их типами**

#### **Теория:**

- Ознакомление с основами языка Kotlin и его возможностями для работы с потоками данных.
- Изучение основных типов потоков данных, таких как Observable, Flowable, Single, Completable и Maybe.
- Понимание различий между этими типами потоков данных и их использование для различных сценариев разработки.
- Изучение операторов для работы с потоками данных, таких как map, flatMap, filter, reduce и другие.
- Понимание принципов использования асинхронных операций и управления многопоточностью при работе с потоками данных.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка функциональности, используя потоки данных различных типов, таких как Observable, Flowable, Single и другие.
- Использование операторов для обработки и преобразования данных в потоках.
- Работа с асинхронными операциями и управление многопоточностью при работе с потоками данных.
- Тестирование и отладка функциональности, реализованной с использованием потоков данных.
- Оптимизация и улучшение производительности приложения с использованием потоков данных и их типов.

## **2.2.Использование LiveData для обновления пользовательского интерфейса**

### **Теория:**

- Ознакомление с основами языка Kotlin и его использованием для разработки на платформе Android.
- Изучение основ архитектуры компонентов Jetpack, включая LiveData.
- Понимание роли и принципов работы LiveData при обновлении пользовательского интерфейса.
- Изучение основных методов и функций LiveData, таких как observe, setValue и getValue.
- Понимание лучших практик использования LiveData для обновления пользовательского интерфейса.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация пользовательского интерфейса, включая компоненты, которые будут обновляться с помощью LiveData.
- Создание ViewModel, которая будет содержать LiveData для обновления данных в пользовательском интерфейсе.
- Использование метода observe LiveData для прослушивания изменений данных и обновления пользовательского интерфейса в соответствии с этими изменениями.
- Реализация методов setValue и getValue для обновления данных в LiveData и получения текущего значения данных.
- Тестирование и отладка приложения с использованием LiveData для обновления пользовательского интерфейса.
- Оптимизация и улучшение функциональности и производительности приложения с использованием LiveData для обновления пользовательского интерфейса.

## **2.3.Тестирование потоков данных**

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка потока данных, который будет тестироваться, например, использование LiveData, RxJava или Coroutines для работы с асинхронными операциями.
- Написание модульных тестов для проверки правильности работы потока данных, используя инструменты и фреймворки для тестирования на платформе Android.
- Интеграция тестов в процесс сборки и выполнение тестов с использованием инструментов разработки, таких как Android Studio.
- Анализ результатов тестирования и исправление ошибок, если таковые будут обнаружены.
- Оптимизация и улучшение работы потока данных на основе полученных результатов и обратной связи от тестирования.

## **Модуль 3. Работа с базами данных**

### **3.1.Основные принципы работы с базам данных**

#### **Теория:**

- Ознакомление с основами работы с базами данных на платформе Android.
- Изучение основных принципов и понятий в работе с базами данных, таких как таблицы, записи, индексы и запросы.
- Рассмотрение различных типов баз данных, которые можно использовать на платформе Android, включая SQLite, Room и Firebase.
- Понимание преимуществ и недостатков каждого типа базы данных и выбор наиболее подходящего для конкретного приложения.
- Ознакомление с основами работы с SQLite, который является стандартным встроенным решением для работы с базами данных на платформе Android.

- Изучение основ работы с Room, библиотекой для работы с базами данных, которая предоставляет удобные абстракции и возможности для работы с базами данных на платформе Android.

- Понимание основных шагов и подходов к созданию и использованию баз данных на платформе Android.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Изучение и настройка необходимых зависимостей, таких как SQLite, Room или Firebase, для работы с базами данных.

- Создание и настройка базы данных в приложении, включая определение таблиц, структуры и связей между ними.

- Написание функций и методов для взаимодействия с базой данных, включая добавление, изменение, удаление и запросы данных.

- Тестирование и отладка функциональности работы с базами данных в приложении.

- Оптимизация и улучшение работы с базами данных в приложении, включая оптимизацию запросов, индексов и других параметров.

### **3.2.Создание базы данных в приложении на Kotlin**

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Реализация модели данных для работы с базой данных, включая создание таблиц и определение структуры данных.

- Создание класса для работы с базой данных, включая создание и обновление базы данных, а также выполнение операций вставки, обновления и удаления данных.

- Импорт SQLiteOpenHelper для управления базой данных в приложении.

- Реализация методов для выборки данных из базы данных, используя операторы SQL.

- Тестирование функциональности базы данных в приложении, включая проверку правильности выполнения операций вставки, обновления и удаления данных.

- Оптимизация и улучшение функциональности и производительности базы данных в приложении на основе полученных результатов и обратной связи.

### **3.3.Изучение системы SQLite**

**Теория:**

- Введение в систему SQLite и ее использование для хранения данных в приложениях на платформе Android.

- Основные принципы работы с базами данных на основе SQLite, включая создание таблиц, добавление и удаление данных, выполнение запросов и обновление данных.

- Изучение основных методов и функций, используемых для работы с базами данных на основе SQLite в Kotlin.

- Понимание принципов организации данных в SQLite и использование индексов и ограничений для оптимизации производительности.

- Обзор основных принципов безопасности и защиты данных в базах данных на основе SQLite.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Реализация функциональности, которая будет взаимодействовать с базой данных на основе SQLite, например, сохранение и загрузка данных в базу данных.

- Создание класса-помощника для работы с базой данных на основе SQLite и выполнение операций, таких как создание таблиц, добавление и удаление данных, выполнение запросов и обновление данных.

- Тестирование и отладка функциональности связанной с базой данных на основе SQLite.

- Оптимизация и улучшение производительности базы данных на основе SQLite, включая использование индексов и ограничений по датам.
- Реализация механизмов безопасности, таких как шифрование и защита данных в базе данных на основе SQLite.

## **Модуль 4. Работа с сетью**

### **4.1. Работа с сетевыми запросами/Использование Retrofit для работы с сетью**

#### **Теория:**

- Введение в работу с сетевыми запросами на платформе Android и их важность для разработки приложений.
- Ознакомление с протоколом HTTP и основными методами запросов, такими как GET, POST, PUT, DELETE.
- Изучение основных инструментов и библиотек для работы с сетевыми запросами на платформе Android, включая Retrofit.
- Понимание принципов работы Retrofit и его основных компонентов, таких как интерфейсы API, аннотации и конвертеры данных.
- Обзор основных функций и методов Retrofit, таких как создание запросов, обработка ответов, загрузка и отправка данных, управление временем ожидания и прочее.
- Рассмотрение лучших практик и советов по работе с сетевыми запросами на платформе Android с использованием Retrofit.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация функциональности, которая будет взаимодействовать с внешними API с использованием Retrofit.
- Создание интерфейсов API с помощью Retrofit, объявляющих методы запросов и аннотаций для указания URL и параметров запроса.
- Использование Retrofit для создания запросов, обратной обработки ответов, загрузки и отправки данных.
- Тестирование и отладка функциональности запросов с использованием Retrofit.
- Оптимизация и улучшение производительности сетевых запросов с использованием Retrofit, включая управление временем ожидания, кэширование и другие возможности библиотеки.
- Расширение функциональности приложения, используя Retrofit для работы с различными функциями API, такими как загрузка изображений, аутентификация и другие.

### **4.2. Изучение REST API**

#### **Теория:**

- Введение в концепцию REST (Representational State Transfer) и его использование для обмена данными между клиентскими и серверными приложениями.
- Ознакомление с основными принципами работы с REST API, такими как HTTP-методы (GET, POST, PUT, DELETE), параметры запроса, заголовки и коды состояния.
- Изучение форматов данных, используемых в REST API, таких как JSON и XML, и их обработка в приложениях на платформе Android с использованием языка Kotlin.
- Основы аутентификации и авторизации при работе с REST API, включая токены доступа и методы безопасности.
- Обзор популярных библиотек и инструментов для работы с REST API на платформе Android, таких как Retrofit, OkHttp и Gson.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация функциональности, которая будет взаимодействовать с REST API, например, получение данных с сервера, отправка данных на сервер и обновление данных на сервере.

- Настройка клиентского запроса с использованием методов HTTP, параметров запроса и заголовков в соответствии с документацией REST API.
- Получение и обработка ответа от сервера, включая парсинг данных в формате JSON или XML.
- Тестирование и отладка функциональности, связанной с REST API.
- Оптимизация и улучшение производительности при работе с REST API, включая использование кэширования и оптимизацию запросов.

## **Модуль 5. Продвинутое UI**

### **5.1. Изучение новых инструментов и элементов пользовательского интерфейса**

#### **Теория:**

- Обзор новых инструментов и элементов пользовательского интерфейса, которые доступны для разработки на платформе Android, такие как Material Design, MotionLayout, ConstraintLayout и другие.
- Понимание основных принципов дизайна пользовательского интерфейса и их применение при разработке приложений на платформе Android.
- Изучение методов и техник для создания анимации и переходов между экранами в пользовательском интерфейсе с использованием новых инструментов и элементов.
- Ознакомление с возможностями адаптивного дизайна и создание пользовательского интерфейса, который будет прекрасно выглядеть на разных устройствах и разрешениях экрана.
- Рассмотрение принципов доступности пользовательского интерфейса и его создание с учетом потребностей пользователей со сниженными возможностями.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация пользовательского интерфейса с использованием новых инструментов и элементов, таких как Material Design, MotionLayout и ConstraintLayout.
- Создание анимации и переходов между экранами в пользовательском интерфейсе с использованием MotionLayout и других инструментов.
- Тестирование и отладка функциональности и внешнего вида пользовательского интерфейса на разных устройствах и разрешениях экрана.
- Оптимизация и улучшение пользовательского интерфейса, включая адаптивный дизайн и учет принципов доступности пользователей со сниженными возможностями.

### **5.2. Создание анимации с использованием ConstraintLayout и MotionLayout**

#### **Теория:**

- Введение в анимацию на платформе Android и ее роль в создании интерактивного и привлекательного пользовательского интерфейса.
- Основы работы с ConstraintLayout и его возможности для создания сложной анимации с использованием ConstraintSet и TransitionManager.
- Изучение основ MotionLayout, который является расширением ConstraintLayout и позволяет создавать более сложные и динамические анимации.
- Ознакомление с ключевыми понятиями MotionLayout, такими как ConstraintSet, KeyFrameSet, KeyAttribute и многими другими.
- Обзор различных видов анимаций, которые могут быть созданы с использованием ConstraintLayout и MotionLayout, такие как переходы, перемещения, масштабирование, изменение альфа-канала и многие другие.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация пользовательского интерфейса, включая элементы, которые будут анимированы с использованием ConstraintLayout и MotionLayout.
- Настройка ConstraintSet и TransitionManager для создания анимации с использованием ConstraintLayout.

- Создание MotionScene и настройка ConstraintSet, KeyFrameSet и KeyAttribute для создания более сложной и динамической анимации с использованием MotionLayout.
- Тестирование и отладка анимации, включая реагирование на пользовательский ввод или изменение данных.
- Оптимизация и улучшение производительности анимации, включая использование оптимизированных констрейнтов и ограничений.

### **5.3. Работа со светлой и темной темами**

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация пользовательского интерфейса, включая различные элементы, такие как кнопки, текстовые поля и фон, с использованием светлой и темной темы.
- Настройка стилей и атрибутов для элементов пользовательского интерфейса, чтобы они соответствовали светлой и темной теме.
- Настройка ресурсов, таких как изображения и цвета, для работы как в светлой, так и в темной теме.
- Использование механизма автоматического переключения между светлой и темной темами в зависимости от системных настроек или пользовательских предпочтений.
- Тестирование и отладка пользовательского интерфейса в светлой и темной теме.
- Оптимизация и улучшение внешнего вида и функциональности пользовательского интерфейса в светлой и темной теме.

## **Модуль 6. Использование библиотек и инструментов**

### **6.1. Промежуточная Аттестация. Тестирование**

#### **Теория: Тестирование**

### **6.2. Изучение библиотек Dagger 2 и Koin**

#### **Теория:**

- Введение в понятие собственных библиотек и их роль в разработке на платформе Android.
- Ознакомление с основными концепциями и принципами разработки собственных библиотек на языке Kotlin.

#### **Практика:**

- Установка необходимых инструментов и зависимостей для работы с Kotlin и Android.
- Ознакомление с базовыми концепциями и принципами Dependency Injection (DI) и Inversion of Control (IoC).
- Изучение основных компонентов и аннотаций в Dagger 2 и Koin для реализации DI в приложении Android.
- Разработка простого приложения Android с использованием Dagger 2 и Koin для инъекции зависимостей и управления объектами.
- Создание модулей, компонентов и провайдеров в Dagger 2 и Koin для описания зависимостей и их предоставления.
- Использование аннотаций и методов в Dagger 2 и Koin для связывания зависимостей и их использования в разных частях приложения.
- Тестирование и отладка приложения с использованием Dagger 2 и Koin для проверки правильной работы внедрения зависимостей.
- Создание более сложного приложения, используя различные функции и возможности Dagger 2 и Koin для DI и IoC.

### **6.3. Работа с инструментом Android Profiler для оптимизации приложения**

#### **Теория:**

- Введение в концепцию Android Profiler и его роль в оптимизации приложений на платформе Android.
- Ознакомление с основными возможностями и функциональностью Android Profiler, такими как мониторинг использования ресурсов, профилирование процессов, анализ потоков выполнения кода и т.д.
- Изучение основных показателей и метрик, которые можно анализировать с помощью Android Profiler, таких как использование ЦП и памяти, временные затраты на выполнение операций, использование сети и т.д.
- Рассмотрение различных сценариев использования Android Profiler для оптимизации производительности, устранения утечек памяти, снижения использования энергии и общего улучшения работы приложения.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Интеграция Android Profiler в проект с помощью инструментов разработки, таких как Android Studio и SDK.
- Использование Android Profiler для мониторинга использования ресурсов приложением, включая использование ЦП, памяти, сети и энергии.
- Профилирование процессов выполнения кода приложением и идентификация узких мест и узловых точек для дальнейшей оптимизации.
- Анализ временных затрат на выполнение операций в приложении с помощью Android Profiler и определение возможностей для улучшения производительности.
- Устранение утечек памяти и оптимизация использования памяти в приложении с помощью инструментов Android Profiler.
- Тестирование и отладка приложения после проведения оптимизаций с использованием Android Profiler.

#### **6.4.Создание собственных библиотек**

**Теория:**

- Введение в понятие собственных библиотек и их роль в разработке на платформе Android.
- Ознакомление с основными концепциями и принципами разработки собственных библиотек на языке Kotlin.
- Изучение различных подходов и инструментов для создания и развития собственных библиотек на платформе Android.
- Рассмотрение методов публикации и распространения собственных библиотек для использования в других проектах.
- Обзор лучших практик и рекомендаций по разработке и поддержке собственных библиотек на платформе Android.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Определение функциональности и набора классов, которые будут включены в собственную библиотеку.
- Разработка кода для реализации функциональности каждого класса в собственной библиотеке.
- Тестирование и отладка функциональности собственной библиотеки внутри проекта.
- Создание документации и примеров использования собственной библиотеки для других разработчиков.
- Публикация собственной библиотеки в репозитории или маркетплейсе для использования в других проектах.
- Управление и обновление собственной библиотеки в соответствии с требованиями и обратной связью от пользователей.

### **Модуль 7. Многопоточность**

## **7.1. Работа с многопоточностью и её применение в приложении**

### **Теория:**

- Введение в понятие многопоточности и ее роль в разработке приложений на платформе Android.
- Ознакомление с основными концепциями и принципами многопоточности, такими как процессы, потоки, синхронизация и взаимодействие между потоками.
- Изучение подходов и инструментов для работы с многопоточностью на платформе Android, таких как классы Thread и Handler, AsyncTask, ThreadPoolExecutor и т.д.
- Рассмотрение различных сценариев применения многопоточности в разработке приложений на платформе Android, таких как параллельная загрузка данных, асинхронные операции с сетью, обработка задач в фоне и т.д.
- Обзор лучших практик и рекомендаций по использованию многопоточности для оптимизации производительности и повышения отзывчивости приложений на платформе Android.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка функциональности, которая будет выполняться в отдельном потоке или в фоновом режиме с помощью многопоточности.
- Использование классов Thread и Handler, AsyncTask или других подходов для создания и управления потоками выполнения кода.
- Реализация параллельной загрузки данных из сети, обработку асинхронных задач, а также работу с базой данных или файловой системой в фоновом режиме.
- Обеспечение синхронизации и взаимодействия между потоками при необходимости с использованием механизмов синхронизации, таких как мониторы, блокировки или семафоры.
- Тестирование и отладка функциональности, использующей многопоточность, в приложении.

## **7.2. Изучение принципа работы Coroutines**

### **Теория:**

- Введение в понятие корутин (Coroutines) и их роль в разработке на платформе Android.
- Ознакомление с основными принципами и концепциями работы с корутинами на языке Kotlin.
- Изучение основных возможностей и функциональности корутин, таких как асинхронное выполнение задач, управление потоками выполнения, отмена операций и т.д.
- Рассмотрение различных подходов к использованию корутин в приложениях на платформе Android, включая создание и запуск корутин, использование синтаксиса сопрограмм и обработку ошибок.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импортирование необходимых зависимостей и библиотек для работы с корутинами.
- Создание корутин для выполнения асинхронных операций в приложении, таких как загрузка данных с сервера или выполнение длительных вычислений.
- Управление потоками выполнения с помощью корутин, таких как запуск корутин в отдельных потоках или ограничение числа одновременно выполняющихся корутин.
- Отмена операций с помощью корутин, включая отмену корутин при закрытии приложения или по требованию пользователя.
- Обработка ошибок и исключений при работе с корутинами, включая понимание механизмов обработки и распространения исключений.

## **7.3. Создание собственных Coroutines**

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.



- Импорт необходимых библиотек и настройка окружения для работы с корутинами.
- Создание собственной корутины для выполнения асинхронной задачи.
- Определение сценария, в котором будет использоваться собственная корутина, например, загрузка данных из сети или обращение к базе данных.
- Разработка кода для выполнения задачи с использованием собственной корутины.
- Тестирование и отладка функциональности с использованием собственной корутины.
- Оптимизация и улучшение собственной корутины, если это необходимо, для повышения производительности и эффективности кода.

## **Модуль 8. Проверка качества кода**

### **8.1.Изучение принципов написания чистого и эффективного кода**

#### **Теория:**

- Введение в концепцию чистого и эффективного кода и его роль в разработке на платформе Android.
- Ознакомление с основными принципами и практиками написания чистого и эффективного кода на языке Kotlin.
- Изучение различных методов и подходов для повышения читаемости, поддерживаемости и производительности кода.
- Рассмотрение принципов SOLID, DRY, KISS и других паттернов и архитектурных принципов, которые помогают создать чистый и эффективный код.
- Обзор инструментов и техник для автоматической проверки качества кода, таких как статический анализаторы, инспекторы и модульные тесты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка функциональности приложения, следуя принципам чистого и эффективного кода.
- Использование правильной структуры проекта, такой как разделение на модули или пакеты, для облегчения поддержки и читаемости кода.
- Реализация модульных тестов для проверки корректности и надежности функциональности.
- Использование инструментов статического анализа кода для обнаружения потенциальных проблем и улучшений в коде.
- Оптимизация и улучшение кода с помощью рефакторинга, сокращения дублирования кода, оптимизации производительности и других методов.

### **8.2.Использование статических анализаторов кода и утилит для проверки качества кода**

#### **Теория:**

- Введение в понятие статического анализа кода и его роль в разработке на платформе Android.
- Ознакомление с основными понятиями и принципами работы статических анализаторов кода, таких как линтеры, плагины и другие инструменты.
- Изучение различных инструментов для статического анализа кода на платформе Android, таких как Android Lint, Detekt, Ktlint и другие.
- Рассмотрение различных аспектов, которые могут быть оценены с помощью статического анализа, таких как стиль кодирования, устранение потенциальных ошибок, оптимизация производительности и другие.
- Общие рекомендации по использованию статических анализаторов кода и утилит для проверки качества кода на платформе Android.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Интеграция выбранного статического анализатора кода в проект с помощью инструментов разработки, таких как Android Studio и плагины статического анализа.
- Прогон кода через статический анализатор и анализ полученных результатов на наличие стилистических ошибок, потенциальных проблем или неэффективного кода.
- Устранение обнаруженных ошибок или проблем, вносимых статическим анализатором, и повторный запуск анализа для проверки исправлений.
- Тестирование и отладка проекта после проведения проверки качества кода с использованием статического анализатора.

### **8.3. Работа над оптимизацией производительности кода**

#### **Теория:**

- Введение в понятие оптимизации производительности и ее роль в разработке на платформе Android.
- Ознакомление с основными принципами оптимизации производительности кода на языке Kotlin.
- Изучение различных подходов и инструментов для оптимизации производительности кода на платформе Android.
- Рассмотрение оптимизации производительности в различных аспектах разработки, таких как скорость выполнения операций, использование памяти, энергозатраты и т.д.
- Обзор лучших практик и рекомендаций по оптимизации производительности кода на платформе Android.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Идентификация узких мест и проблем, связанных с производительностью кода в проекте.
- Анализ и определение возможных решений для оптимизации производительности.
- Реализация оптимизированного кода, включая использование эффективных алгоритмов, оптимизированных структур данных и т.д.
- Тестирование и отладка оптимизированного кода для проверки его производительности.
- Использование инструментов и профилировщиков производительности для измерения и анализа производительности оптимизированного кода.
- Оценка и сравнение производительности оптимизированного кода с исходной версией.

## **Модуль 9. Работа со звуком и видео**

### **9.1. Изучение принципов работы с аудио и видео в Android приложениях**

#### **Теория:**

- Введение в понятие работы с аудио и видео в Android приложениях.
- Ознакомление с основными принципами работы с аудио и видео файлами, воспроизведения и записи звука, видео.
- Изучение различных форматов аудио и видео файлов и их особенностей в рамках платформы Android.
- Рассмотрение основных классов и API, предоставляемых платформой Android для работы с аудио и видео, таких как MediaPlayer, AudioRecord, MediaRecorder и т.д.
- Ознакомление с принципами работы с аудио и видео стримами, потоковой передачей данных.
- Рассмотрение способов обработки аудио и видео данных, таких как применение эффектов, фильтров, кодеков и т.д.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация функциональности воспроизведения аудио и видео файлов, записи звука, видео с использованием соответствующих классов и API.

- Использование различных форматов аудио и видео файлов и проверка их совместимости с платформой Android.
- Реализация различных эффектов и фильтров для обработки аудио и видео данных, таких как изменение тембра голоса, добавление эффектов эхо, реверберации и т.д.
- Разработка функциональности потокового воспроизведения и передачи аудио и видео данных через сеть.
- Тестирование и отладка функциональности работы с аудио и видео в приложении для проверки ее правильности и производительности.

## **9.2.Создание мультимедийных приложений**

### **Теория:**

- Изучение основных концепций и возможностей языка Kotlin для разработки на платформе Android.
- Понимание основных компонентов мультимедийных приложений на платформе Android, таких как воспроизведение аудио и видео, работа с графикой и изображениями, использование микрофона и камеры и т.д.
- Знакомство с основными библиотеками и инструментами для работы с мультимедиа на платформе Android, такими как ExoPlayer, MediaCodec, Glide и т.д.
- Изучение различных подходов и техник для разработки мультимедийных приложений, включая анимацию, взаимодействие с пользователем, обработку мультимедийных данных и т.д.
- Рассмотрение практических примеров и сценариев разработки мультимедийных приложений на платформе Android.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Создание пользовательского интерфейса для мультимедийного приложения, включая компоненты для воспроизведения аудио и видео, работы с изображениями и графикой, обработки мультимедийных данных и т.д.
- Разработка логики приложения, включая управление воспроизведением, обработку пользовательских действий, взаимодействие с микрофоном и камерой и т.д.
- Интеграция основных библиотек и инструментов для работы с мультимедиа на платформе Android, таких как ExoPlayer для воспроизведения аудио и видео, Glide для работы с изображениями и т.д.
- Тестирование и отладка мультимедийного приложения для проверки его функциональности и работоспособности.
- Оптимизация и улучшение производительности мультимедийного приложения, включая оптимизацию кода, управление памятью и энергопотреблением, улучшение интерфейса пользователя и т.д.

## **9.3.Работа с плеерами и кодеками**

### **Теория:**

- Введение в работу с плеерами и кодеками на платформе Android.
- Обзор основных понятий и технологий, связанных с плеерами и кодеками, таких как медиаформаты, контейнеры, кодирование и декодирование аудио и видео и т.д.
- Изучение основных API и библиотек для работы с плеерами и кодеками на платформе Android, таких как MediaCodec, MediaPlayer, ExoPlayer и т.д.
- Ознакомление с принципами работы плееров и кодеков, таких как буферизация, синхронизация, обработка ошибок и другие аспекты проигрывания медиафайлов.
- Рассмотрение основных проблем и сложностей, связанных с работой с плеерами и кодеками на платформе Android, включая поддержку различных форматов и кодеков, обработку разных типов видео и аудио потоков, обработку ошибок и т.д.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Импорт и настройка необходимых библиотек и зависимостей для работы с плеерами и кодеками.
- Реализация базового функционала плеера, такого как загрузка и воспроизведение медиафайлов, управление воспроизведением, отображение видео- и аудиоданных и т.д.
- Использование соответствующих API и методов для работы с плеерами и кодеками, таких как MediaCodec для кодирования и декодирования медиафайлов, MediaPlayer или ExoPlayer для проигрывания медиафайлов и другие API и методы.
- Реализация дополнительных функций и возможностей плеера, таких как управление аудиодорожками, субтитрами, настройки качества видео и аудио и другие функции.
- Тестирование и отладка плеера, включая проверку корректности воспроизведения, обработку ошибок и другие аспекты работы плеера с медиафайлами и кодеками.

## **Модуль 10. Bluetooth и NFC**

### **10.1. Основы работы с Bluetooth и NFC в Android /Изучение NFC-технологий и их применения**

#### **Теория:**

- Введение в работу с Bluetooth и NFC на платформе Android.
- Обзор основных понятий и принципов работы с Bluetooth и NFC, их различия и применение.
- Изучение основных API и методов для работы с Bluetooth и NFC на платформе Android, таких как BluetoothAdapter, BluetoothDevice, NfcAdapter и т.д.
- Ознакомление с возможностями Bluetooth и NFC, такими как обмен данными между устройствами, определение доступных устройств, обнаружение тегов NFC и другие функции.
- Рассмотрение основных проблем и сложностей, связанных с работой с Bluetooth и NFC на платформе Android, включая обработку ошибок, управление соединениями, безопасность и другие аспекты работы с этими технологиями.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с Bluetooth и NFC.
- Реализация базового функционала работы с Bluetooth, такого как поиск доступных устройств, установка соединения, обмен данными и управление соединениями.
- Реализация базового функционала работы с NFC, такого как чтение и запись данных на теги NFC, обмен данными между устройствами и другие функции.
- Использование соответствующих API и методов для работы с Bluetooth и NFC на платформе Android, таких как BluetoothAdapter для управления Bluetooth-соединениями, BluetoothDevice для обмена данными с Bluetooth-устройствами, NfcAdapter для работы с NFC и другие API и методы.
- Тестирование и отладка функционала работы с Bluetooth и NFC, включая проверку корректности соединений, обмен данных и другие аспекты работы с этими технологиями.

### **10.2. Разработка приложений для работы с Bluetooth и NFC**

#### **Теория:**

- Введение в разработку приложений для работы с Bluetooth и NFC на платформе Android.
- Обзор основных понятий и технологий, связанных с Bluetooth и NFC, таких как обнаружение устройств, установление соединений, обмен данными и т.д.
- Изучение основных API и библиотек для работы с Bluetooth и NFC на платформе Android, таких как BluetoothAdapter, BluetoothDevice, NfcAdapter и т.д.
- Ознакомление с принципами работы Bluetooth и NFC, таких как процесс парирования устройств, передача данных, режимы работы и другие аспекты коммуникации.

- Рассмотрение основных проблем и сложностей, связанных с разработкой приложений для работы с Bluetooth и NFC на платформе Android, включая поддержку разных версий Bluetooth и NFC, обработку ошибок, безопасность и т.д.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с Bluetooth и NFC.
- Реализация функционала для работы с Bluetooth, такого как поиск доступных устройств, установление соединения, передача данных и другие операции.
- Использование соответствующих API и методов для работы с Bluetooth, таких как BluetoothAdapter для включения и отключения Bluetooth, BluetoothDevice для обнаружения и управления устройствами и другие API и методы.
- Реализация функционала для работы с NFC, такого как чтение и запись данных на теги NFC, обмен данными между устройствами через NFC и другие операции.
- Использование соответствующих API и методов для работы с NFC, таких как NfcAdapter для инициализации NFC, NdefMessage для чтения и записи данных на теги NFC и другие API и методы.
- Тестирование и отладка приложения, включая проверку корректности работы с Bluetooth и NFC, обработку ошибок и другие аспекты коммуникации.

## **Модуль 11. Работа с Firebase**

### **11.1. Изучение сервиса Firebase и его использование для разработки приложений**

**Теория:** Firebase - это сервис разработки приложений, предоставляемый компанией Google. С его помощью можно создавать приложения для мобильных, веб- и серверных платформ. Firebase включает в себя множество инструментов и сервисов для разработки и управления приложениями, включая аутентификацию пользователей, хранение и синхронизацию данных, аналитику и многое другое.

**Практика:**

1. Создайте новый проект Android Studio с поддержкой языка Kotlin.
2. Подключите сервис Firebase к вашему проекту. Включите необходимые модули Firebase, такие как Firebase аутентификация, Firebase Realtime Database и Firebase Cloud Messaging.
3. Настройте аутентификацию пользователей с использованием Firebase. Реализуйте функции регистрации, входа и выхода пользователей, используя методы Firebase аутентификации.
4. Используйте Firebase Realtime Database для хранения и синхронизации данных в реальном времени. Создайте модели данных и настройте их связь с Firebase Database. Работайте с данными: добавляйте, изменяйте и удаляйте их, и наблюдайте за их синхронизацией с другими подключенными клиентами.
5. Используйте Firebase Cloud Messaging для отправки уведомлений на мобильные устройства. Настройте прием уведомлений на вашем Android-устройстве и отправьте тестов

### **11.2. Работа с базой данных Firebase**

**Практика:**

1. Создайте новый проект Android Studio и настройте его для разработки на Kotlin.
2. Создайте проект Firebase и настройте его для работы с базой данных Firebase Realtime Database или Firestore.
3. Добавьте зависимости Firebase в файле build.gradle вашего проекта.
4. Инициализируйте Firebase в вашем приложении.
5. Создайте класс для доступа к базе данных, включая получение, запись и удаление данных.

6. Используйте методы доступа к базе данных для получения данных из базы данных и отображения их в пользовательском интерфейсе вашего приложения. Используйте также методы для записи данных в базу данных, обновления или удаления данных по мере необходимости.

### **11.3. Работа с Push уведомлениями на Firebase**

**Теория:** Firebase Cloud Messaging (FCM) - это сервис для отправки push-уведомлений на устройства Android и веб-браузеры. Он позволяет разработчикам отправлять сообщения одному или нескольким устройствам, используя один интерфейс API. FCM поддерживает различные типы сообщений, включая текстовые сообщения, уведомления с изображениями, данные пакеты и многое другое.

#### **Практика:**

1. Создайте проект в консоли Firebase и настройте проект для использования Firebase Cloud Messaging.
2. Добавьте Firebase Cloud Messaging библиотеку в проект Android Studio, следуя инструкциям документации Firebase.
3. Настройте манифест вашего приложения для подключения к Firebase Cloud Messaging. Убедитесь, что у вас есть разрешение на использование интернета в манифесте приложения.
4. Регистрация устройства в Firebase Cloud Messaging. Получите токен устройства, который будет использоваться для отправки уведомлений на это устройство.
5. Настройте серверную часть для отправки push-уведомлений. Это может быть отдельное приложение сервера, которое будет обрабатывать запросы на отправку уведомлений через Firebase SDK или API FCM.
6. Отправьте тестовое уведомление с сервера на зарегистрированное устройство. Удостоверьтесь, что уведомление корректно отображается на устройстве.
7. Разработайте логику вашего приложения для работы с push-уведомлениями. Обработайте получение уведомлений на устройстве и выполните необходимые действия, когда уведомление прибывает.

## **Модуль 12. Работа с AR и VR**

### **12.1. Создание AR и VR приложений**

#### **Теория:**

- Введение в создание AR (Augmented Reality) и VR (Virtual Reality) приложений на платформе Android.
- Обзор основных понятий и технологий, связанных с AR и VR, таких как трекинг, рендеринг, визуализация и т.д.
- Изучение основных API и библиотек для работы с AR и VR на платформе Android, таких как ARCore, ARKit, Google VR SDK и т.д.
- Ознакомление с принципами работы AR и VR, таких как распознавание маркеров, позиционирование объектов, виртуальное взаимодействие и другие аспекты создания иммерсивного визуального опыта.
- Рассмотрение основных проблем и сложностей, связанных с разработкой AR и VR приложений, включая обработку пользовательского ввода, оптимизацию производительности, поддержку разных устройств и т.д.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с AR и VR.

- Реализация функционала для работы с AR, такого как распознавание маркеров, создание и позиционирование виртуальных объектов, взаимодействие с пользователем и другие операции.
- Использование соответствующих API и методов для работы с AR, таких как ARCore или ARKit для трекинга маркеров, Sceneform или Unity для создания и визуализации 3D-объектов и другие API и методы.
- Реализация функционала для работы с VR, такого как визуализация 360-градусных изображений и видео, взаимодействие с пользователем через VR-гарнитуру и другие операции.
- Использование соответствующих API и методов для работы с VR, таких как Google VR SDK или Unity для визуализации и взаимодействия с контентом VR и другие API и методы.
- Тестирование и отладка приложения, включая проверку корректности работы с AR и VR, обработку пользовательского ввода и другие аспекты визуального восприятия.

## **12.2.Изучение основ использования ARCore и Vuforia в Android**

### **Теория:**

- Введение в разработку приложений с использованием ARCore и Vuforia на платформе Android.
- Обзор основных понятий и технологий дополненной реальности (AR), таких как трекинг, отображение 3D графики, обнаружение поверхностей и другие возможности ARCore и Vuforia.
- Изучение основных API и библиотек для работы с ARCore и Vuforia на платформе Android, таких как ARCore SDK и Vuforia SDK.
- Ознакомление с принципами работы ARCore и Vuforia, таких как отображение виртуальных объектов на реальных поверхностях, распознавание и отслеживание маркеров, обработка жестов и другие аспекты разработки AR-приложений.
- Рассмотрение основных проблем и сложностей, связанных с разработкой приложений с использованием ARCore и Vuforia, включая поддержку разных устройств, оптимизацию производительности, обработку ошибок и т.д.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с ARCore и Vuforia.
- Реализация функционала для работы с ARCore, такого как обнаружение и отслеживание поверхностей, размещение виртуальных объектов на реальных поверхностях, обработка жестов и другие возможности ARCore.
- Использование соответствующих API и методов ARCore для работы с AR-приложениями, таких как ARSession для инициализации AR, ARFrame для получения данных о камере и др.
- Реализация функционала для работы с Vuforia, такого как распознавание маркеров, отображение виртуальных объектов на маркерах, обработка жестов и другие возможности Vuforia.
- Использование соответствующих API и методов Vuforia для работы с AR-приложениями, таких как TrackerManager для инициализации трекера, ImageTarget для создания маркеров и др.
- Тестирование и отладка приложения, включая проверку корректности работы с ARCore и Vuforia, обработку ошибок и другие аспекты разработки AR-приложений.

## **12.3.Создание взаимодействия AR-элементов и виртуальной реальности и реального мира**

### **Теория:**

- Введение в создание взаимодействия AR-элементов с виртуальной реальностью и реальным миром на платформе Android.

- Обзор основных понятий и технологий, связанных с AR и VR, таких как трекинг объектов, создание визуальных элементов на мониторе или смартфоне, создание и управление виртуальной реальностью и другие аспекты AR и VR.
- Изучение основных API и библиотек для работы с AR и VR на платформе Android, таких как ARCore, Google VR SDK и другие.
- Ознакомление с принципами работы AR и VR, таких как распознавание поверхностей, масштабирование и вращение виртуальных объектов, управление камерой и другие аспекты взаимодействия AR-элементов с реальным миром и виртуальной реальностью.
- Рассмотрение основных проблем и сложностей, связанных с созданием взаимодействия AR-элементов с виртуальной реальностью и реальным миром на платформе Android, включая обработку ввода пользователя, синхронизацию между виртуальной и реальной средой, оптимизацию производительности и другие аспекты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с AR и VR.
- Реализация функционала для работы с AR, такого как распознавание поверхностей, расположение и отображение виртуальных объектов на реальные поверхности, управление камерой и другие операции.
- Использование соответствующих API и методов для работы с AR, таких как ARCore для трекинга поверхностей, Sceneform или OpenGL для отображения виртуальных объектов и другие API и методы.
- Реализация функционала для работы с VR, такого как создание виртуальной реальности, управление положением и поворотом камеры, отображение виртуальной среды и другие операции.
- Использование соответствующих API и методов для работы с VR, таких как Google VR SDK для создания виртуальной реальности, Cardboard SDK для работы с устройствами виртуальной реальности и другие API и методы.
- Тестирование и отладка приложения, включая проверку корректности взаимодействия AR-элементов с виртуальной реальностью и реальным миром, обработку пользовательского ввода и другие аспекты AR и VR.

### **Модуль 13. Работа с камерой**

#### **13.1. Работа с камерой в Android**

##### **Теория:**

- Введение в работу с камерой в Android: основные понятия, такие как доступ к камере устройства, получение изображения с камеры, настройка параметров камеры и другие аспекты.
- Обзор основных API и библиотек для работы с камерой в Android, таких как Camera API, Camera2 API и другие.
- Изучение способов работы с камерой в Android, включая различные режимы работы (фото, видео), использование различных камер (передняя, задняя), настройку параметров камеры (разрешение, фокус, экспозиция) и другие возможности.
- Ознакомление с основными проблемами и сложностями, связанными с работой с камерой в Android, такими как управление жизненным циклом камеры, обработка полученных изображений, обработка ошибок и другие аспекты.

##### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с камерой в Android.
- Реализация функционала для работы с камерой в Android, такого как отображение полученного изображения с камеры на экране устройства, съемка фотографий, запись видео и другие операции.



- Использование соответствующих API и методов для работы с камерой в Android, таких как Camera API для доступа к камере устройства, MediaRecorder API для записи видео, SurfaceTexture для отображения изображения на экране и другие API и методы.
- Реализация функционала для настройки параметров камеры в Android, таких как разрешение изображения, фокусировка, экспозиция и другие опции.
- Тестирование и отладка приложения, включая проверку корректности работы с камерой, настройку параметров камеры и другие аспекты.

### **13.2.Создание приложений для работы с камерой**

#### **Теория:**

- Введение в создание приложений для работы с камерой на платформе Android с использованием языка Kotlin.
- Обзор основных понятий и компонентов, связанных с камерой на Android устройствах, таких как доступ к камере устройства, предварительный просмотр, захват изображений или видео, обработка полученных данных и другие аспекты работы с камерой.
- Изучение основных API и библиотек для работы с камерой на платформе Android, таких как Camera API, Camera2 API и другие.
- Ознакомление с принципами работы камеры на устройствах Android, таких как управление параметрами камеры, автофокусировка, экспозиция, обработка изображений или видео и другие аспекты работы с камерой.
- Рассмотрение основных проблем и сложностей, связанных с созданием приложений для работы с камерой на платформе Android, включая обработку ввода пользователя, обработку изображений или видео, оптимизацию производительности и другие аспекты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с камерой.
- Реализация функционала для работы с камерой, такого как доступ к камере устройства, предварительный просмотр изображений или видео, захват изображений или видео, обработка полученных данных и другие операции.
- Использование соответствующих API и методов для работы с камерой, таких как Camera API или Camera2 API для доступа к камере устройства, обработки изображений или видео и других операций.
- Реализация пользовательского интерфейса для работы с камерой, такого как отображение предварительного просмотра изображений или видео, кнопки для захвата изображений или видео и другие элементы интерфейса.
- Тестирование и отладка приложения, включая проверку корректности работы с камерой, обработку пользовательского ввода и другие аспекты работы с камерой.

## **Модуль 14. Работа с GPS**

### **14.1.Изучение принципов работы с GPS**

#### **Теория:**

- Введение в принципы работы с GPS (Global Positioning System) на платформе Android.
- Обзор основных понятий и компонентов, связанных с GPS на устройствах Android, таких как получение текущих координат, определение положения, получение информации о скорости и высоте, отслеживание перемещения и другие аспекты работы с GPS.
- Изучение основных API и библиотек для работы с GPS на платформе Android, таких как Location API, Google Play Services API и другие.
- Ознакомление с принципами работы GPS на устройствах Android, таких как получение данных о местоположении, определение точности измерений, управление энергопотреблением и другие аспекты работы с GPS.

- Рассмотрение основных проблем и сложностей, связанных с работой с GPS на платформе Android, включая обработку полученных данных, оптимизацию производительности, учет различных условий окружающей среды (например, плохая связь, преграды) и другие аспекты.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с GPS.
- Реализация функционала для работы с GPS, такого как получение текущих координат, определение положения, получение информации о скорости и высоте, отслеживание перемещения и другие операции.
- Использование соответствующих API и методов для работы с GPS, таких как Location API или Google Play Services API для получения данных о местоположении, определения точности измерений и других операций.
- Реализация пользовательского интерфейса для отображения данных о местоположении, например, текущие координаты, скорость, высота и другие информационные элементы.
- Тестирование и отладка приложения, включая проверку корректности работы с GPS, обработку пользовательского ввода и другие аспекты работы с GPS.

#### **14.2.Создание приложений для работы с GPS**

**Теория:**

- Введение в создание приложений для работы с GPS на платформе Android с использованием языка Kotlin.
- Обзор основных понятий и компонентов, связанных с GPS на устройствах Android, таких как получение географического положения, отслеживание перемещений, определение расстояния и другие аспекты работы с GPS.
- Изучение основных API и библиотек для работы с GPS на платформе Android, таких как Location API, Google Play Services API и другие.
- Ознакомление с принципами работы GPS на устройствах Android, таких как получение координат географического положения, определение точности позиционирования, обновление положения и другие аспекты работы с GPS.
- Рассмотрение основных проблем и сложностей, связанных с созданием приложений для работы с GPS на платформе Android, включая обработку ввода пользователя, определение текущего положения, отслеживание перемещений, оптимизацию производительности и другие аспекты.

**Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка необходимых библиотек и зависимостей для работы с GPS.
- Реализация функционала для работы с GPS, такого как получение текущего географического положения, отслеживание перемещений, определение расстояния между точками и другие операции.
- Использование соответствующих API и методов для работы с GPS, таких как Location API или Google Play Services API для получения географического положения, обновления положения и других операций.
- Реализация пользовательского интерфейса для работы с GPS, такого как отображение текущего географического положения, кнопки для начала отслеживания перемещений и другие элементы интерфейса.
- Тестирование и отладка приложения, включая проверку корректности работы с GPS, обработку пользовательского ввода и другие аспекты работы с GPS.

#### **14.3.Изучение методов определения местоположения и работа с картами**

**Теория:**

- Введение в методы определения местоположения и работу с картами на платформе Android с использованием языка Kotlin.

- Обзор основных понятий и компонентов, связанных с определением местоположения на устройствах Android, таких как GPS, Wi-Fi, сотовая связь и другие методы определения местоположения.

- Изучение основных API и библиотек для работы с определением местоположения на платформе Android, таких как Google Play Services Location API, Android Location API и другие.

- Ознакомление с принципами работы карт на устройствах Android, таких как отображение местоположения на карте, перемещение карты, поиск мест на карте и другие операции работы с картами.

- Рассмотрение основных проблем и сложностей, связанных с работой с определением местоположения и картами на платформе Android, включая обработку ввода пользователя, обработку данных о местоположении, оптимизацию производительности и другие аспекты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Импорт и настройка необходимых библиотек и зависимостей для работы с определением местоположения и картами.

- Реализация функционала для определения местоположения, такого как получение данных о местоположении устройства, обработка полученных данных и другие операции.

- Использование соответствующих API и методов для определения местоположения, таких как Google Play Services Location API или Android Location API для доступа к данным о местоположении и других операций.

- Реализация функционала для работы с картами, такого как отображение местоположения на карте, перемещение карты, поиск мест на карте и другие операции работы с картами.

- Использование соответствующих API и методов для работы с картами, таких как Google Maps API для отображения карты и других операций работы с картами.

- Тестирование и отладка приложения, включая проверку корректности работы с определением местоположения и картами, обработку пользовательского ввода и другие аспекты работы.

## **Модуль 15. Работа с фрагментами и библиотеками**

### **15.1. Изучение создания фрагментов в Android**

#### **Теория:**

- Введение в создание фрагментов в Android на платформе Android с использованием языка Kotlin.

- Обзор основных понятий и компонентов, связанных с фрагментами в Android, таких как жизненный цикл фрагментов, передача данных между фрагментами, управление фрагментами в Activity и другие аспекты работы с фрагментами.

- Изучение основных классов и методов для работы с фрагментами, таких как Fragment, FragmentManager, FragmentTransaction и другие.

- Ознакомление с принципами работы фрагментов, таких как создание и добавление фрагментов в Activity, управление жизненным циклом фрагментов, обработка пользовательского ввода в фрагментах, передача данных между фрагментами и другие аспекты работы с фрагментами.

- Рассмотрение основных проблем и сложностей, связанных с созданием фрагментов на платформе Android, включая обработку ввода пользователя, управление жизненным циклом фрагментов, обеспечение связи между Activity и фрагментами, оптимизацию производительности и другие аспекты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Импорт и настройка необходимых библиотек и зависимостей для работы с фрагментами.

- Реализация функционала с использованием фрагментов, такого как создание и добавление фрагментов в Activity, управление жизненным циклом фрагментов, обработка пользовательского ввода в фрагментах, передача данных между фрагментами и другие операции.

- Использование соответствующих классов и методов для работы с фрагментами, таких как Fragment, FragmentManager, FragmentTransaction и другие классы и методы.

- Реализация пользовательского интерфейса с использованием фрагментов, такого как отображение фрагментов в различных макетах, взаимодействие между фрагментами и другие аспекты интерфейса.

- Тестирование и отладка приложения, включая проверку корректности работы с фрагментами, обработку пользовательского ввода и другие аспекты работы с фрагментами.

## **15.2. Работа с библиотеками и их внедрение в проект**

### **Теория:**

- Введение в работу с библиотеками на платформе Android с использованием языка Kotlin.

- Обзор основных понятий и принципов работы с библиотеками, таких как импорт и настройка библиотек, использование функций и классов из библиотеки, управление зависимостями и другие аспекты работы с библиотеками.

- Изучение основных типов библиотек на платформе Android, таких как библиотеки для работы с графикой, сетью, базами данных, UI элементами и другие.

- Ознакомление с реализацией процесса внедрения библиотек в проект на платформе Android, такого как добавление зависимости через файлы Gradle или Maven, импорт функций и классов из библиотеки в код проекта и другие аспекты внедрения библиотек.

- Рассмотрение основных проблем и сложностей, связанных с работой с библиотеками и их внедрением в проект на платформе Android, включая совместимость библиотек, обновление библиотек, управление зависимостями и другие аспекты.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.

- Определение необходимых библиотек для проекта и их добавление в проект с использованием файлов Gradle или Maven.

- Импорт и настройка библиотек в проекте, включая добавление зависимостей, настройку файлов конфигурации и другие задачи.

- Использование функций и классов из импортированных библиотек в коде проекта, включая вызов функций, создание объектов классов и использование других возможностей библиотеки.

- Тестирование и отладка приложения, включая проверку корректности работы с библиотеками и обработку возможных ошибок или проблем.

## **15.3. Изучение внедрения зависимостей в проект**

### **Теория:**

- Введение в внедрение зависимостей в проект на платформе Android с использованием языка Kotlin.

- Обзор основных понятий и принципов внедрения зависимостей, таких как инверсия управления, внедрение через конструктор, методы и свойства, разрешение зависимостей и другие аспекты внедрения зависимостей.

- Изучение основных фреймворков и библиотек для внедрения зависимостей на платформе Android, таких как Dagger, Koin, Hilt и другие.

- Ознакомление с основными принципами и паттернами внедрения зависимостей, такими как Dependency Injection Container, Singleton, Provider и другими паттернами внедрения зависимостей.

- Рассмотрение основных проблем и сложностей, связанных с внедрением зависимостей в проект на платформе Android, включая управление жизненным циклом зависимостей, разрешение зависимостей, обработку контекста и другие аспекты.

### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Импорт и настройка выбранного фреймворка или библиотеки для внедрения зависимостей.
- Определение зависимостей, необходимых для проекта, и их конфигурация в выбранном фреймворке или библиотеке.
- Реализация функционала, использующего внедренные зависимости, такого как доступ к базе данных, сетевым службам или другим компонентам приложения.
- Тестирование и отладка приложения, включая проверку корректности внедрения зависимостей, обработку пользовательского ввода и другие аспекты работы с внедренными зависимостями.

## **Модуль 16. Работа с виджетами**

### **16.1.Создание виджетов для Android**

#### **Теория:**

- Введение в разработку на языке Kotlin для платформы Android.
- Обзор основных понятий и принципов разработки виджетов для Android, таких как разметка XML, обработка событий, создание пользовательских интерфейсов и другие аспекты разработки виджетов.
- Ознакомление с основными элементами пользовательского интерфейса Android, такими как кнопки, текстовые поля, изображения и другие виджеты.
- Изучение основных фреймворков и библиотек для создания виджетов на платформе Android, таких как Android Jetpack, Material Design и другие.
- Рассмотрение основных проблем и сложностей, связанных с разработкой виджетов, включая управление жизненным циклом виджетов, обработку событий, взаимодействие с другими компонентами приложения и другие аспекты.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Разработка разметки XML для виджетов, включая размещение виджетов на экране, определение их внешнего вида и свойств.
- Реализация логики обработки событий и взаимодействия виджетов с другими компонентами приложения, такими как база данных, сетевые службы или другие виджеты.
- Тестирование и отладка разработанных виджетов, включая проверку корректности отображения, обработку пользовательского ввода и другие аспекты работы с виджетами.

### **16.2.Изучение работы с системными виджетами и создание собственных**

#### **Теория:**

- Введение в разработку на платформе Android с использованием языка Kotlin.
- Ознакомление с основами работы с системными виджетами, их предназначением и возможностями.
- Изучение основных классов и методов, используемых для создания и настройки системных виджетов в Android.
- Обзор различных типов системных виджетов, таких как TextView, Button, ImageView, ProgressBar, ListView и другие, и их использование в приложениях.
- Рассмотрение различных способов создания собственных пользовательских виджетов, включая создание кастомных View и использование XML-макетов.
- Изучение основных методов и событий, используемых для взаимодействия с системными виджетами, таких как обработка пользовательского ввода, обновление данных и другие операции.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Настройка пользовательского интерфейса с помощью системных виджетов.

- Имплементация функционала системных виджетов, такого как отображение текста, кнопки для выполнения операций, изображений и других виджетов.
- Изучение различных способов взаимодействия с системными виджетами, включая обработку событий, изменение свойств виджетов и другие операции.
- Создание собственных пользовательских виджетов, добавление и настройка их в приложении с использованием кастомных View или XML-макетов.
- Тестирование и отладка приложения, включая проверку корректности работы с системными виджетами и пользовательскими виджетами.

### **16.3. Оптимизация работы с батареями при работе с виджетами**

#### **Теория:**

- Обзор понятия виджетов на платформе Android и их роли в пользовательском интерфейсе.
- Рассмотрение основных виджетов, которые потребляют большое количество энергии при работе, таких как анимации, изображения высокого разрешения и другие сложные компоненты.
- Изучение принципов работы с батареями на платформе Android, включая энергопотребление, управление энергосбережением и другие аспекты оптимизации работы с батареями.
- Рассмотрение основных методов оптимизации работы с батареями при использовании виджетов, таких как использование эффективных ресурсов, ограничение частоты обновления, минимизация использования сетевых запросов и другие методы.
- Изучение специфических функций и возможностей платформы Android, которые позволяют оптимизировать работу с батареями при использовании виджетов, таких как использование JobScheduler, адаптация к изменению состояния батареи и другие функции.

#### **Практика:**

- Создание нового проекта на платформе Android с использованием языка Kotlin.
- Реализация пользовательского интерфейса с использованием виджетов, которые потребляют большое количество энергии, таких как анимации или изображения высокого разрешения.
- Анализ и измерение энергопотребления при работе с виджетами с помощью инструментов разработчика Android.
- Использование различных методов оптимизации работы с батареями для уменьшения энергопотребления при работе с виджетами.
- Тестирование и отладка приложения, включая анализ энергопотребления на разных устройствах и реакцию приложения на изменение состояния батареи.

## **Модуль 17. Создание индивидуального проекта**

### **17.1. Разработка индивидуального проекта на Kotlin для платформы Android**

**Теория:** Разработка индивидуального проекта мобильного приложения или игры на платформе по выбору учащегося является заключительным этапом обучения по программе "Разработка на языке Kotlin для платформы Android– Продвинутый уровень".

**Практика:** Определение целей и требований проекта. Обучающиеся должны определить цель проекта и требования к нему. Например, если проект заключается в разработке приложения, то обучающиеся должны определить, какую проблему приложение будет решать, как он будет использоваться, кто будет использовать приложение и т.д.

### **17.2. Работа с преподавателем по разработке и созданию проекта**

**Теория:** В процессе обучения в программе "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" обучающимся доступно множество ресурсов и инструментов для создания своих проектов. Однако, помощь и поддержка со стороны

преподавателя могут оказаться важными для решения сложных задач и получения советов по оптимизации проекта.

**Практика:** Использовать все доступные ресурсы, такие как книги и документация, чтобы получить информацию о технологиях и методах, используемых в разработке мобильных приложений.

### 17.3. Аттестация по итогам прохождения программы

**Практика:** Защита проектов

#### 1.5. Учебный план

| № п/п  | Название модуля, темы   | Кол-во часов |                       |                      | Формы аттестации/контроля |
|--|---|--------------|-----------------------|----------------------|---------------------------|
|  |   | Всего часов  | Теоретические занятия | Практические занятия |                           |
| <b>Модуль 1. Архитектурные подходы в Android</b> |   | <b>8</b>     | <b>3</b>              | <b>5</b>             |                           |
| 1.1  | Модель разработки приложений на платформе Android                     | 2            | 1                     | 1                    |                           |
| 1.2  | Изучение основ архитектурного подхода MVVM                            | 2            | 1                     | 1                    |                           |
| 1.3  | Создание архитектуры приложения на основе MVVM                        | 4            | 1                     | 3                    |                           |
| <b>Модуль 2. Потоки данных</b>                   |   | <b>8</b>     | <b>2</b>              | <b>6</b>             |                           |
| 2.1  | Работа с потоками данных и их типами                                  | 2            | 1                     | 1                    |                           |
| 2.2  | Использование LiveData для обновления пользовательского интерфейса    | 4            | 1                     | 3                    |                           |
| 2.3  | Тестирование потоков данных   | 2            | 0                     | 2                    |                           |
| <b>Модуль 3. Работа с базами данных</b>          |   | <b>8</b>     | <b>2</b>              | <b>6</b>             |                           |
| 3.1  | Основные принципы работы с базами данных                              | 2            | 1                     | 1                    |                           |
| 3.2  | Создание базы данных в приложении на Kotlin                           | 2            | 0                     | 2                    |                           |
| 3.3  | Изучение системы SQLite   | 4            | 1                     | 3                    |                           |
| <b>Модуль 4. Работа с сетью</b>                  |   | <b>8</b>     | <b>2</b>              | <b>6</b>             |                           |
| 4.1  | Работа с сетевыми запросами/Использование Retrofit для работы с сетью | 4            | 1                     | 3                    |                           |
| 4.2  | Изучение REST API   | 4            | 1                     | 3                    |                           |
| <b>Модуль 5. Продвинутое UI</b>                  |   | <b>8</b>     | <b>2</b>              | <b>6</b>             |                           |
| 5.1  | Изучение новых инструментов и элементов пользовательского интерфейса  | 2            | 1                     | 1                    |                           |
| 5.2  | Создание анимации с использованием ConstraintLayout и MotionLayout    | 4            | 1                     | 3                    |                           |

|   |  |           |          |          |
|---|--|-----------|----------|----------|
| 5.3   | Работа со светлой и темной темами  | 2         | 0        | 2        |
| <b>Модуль 6. Использование библиотек и инструментов</b> |  | <b>10</b> | <b>3</b> | <b>7</b> |
| 6.1   | Промежуточная Аттестация. Тестирование   | 2         | 0        | 2        |
| 6.2   | Изучение библиотек Dagger 2 и Koip   | 2         | 1        | 1        |
| 6.3   | Работа с инструментом Android Profiler для оптимизации приложения                  | 2         | 1        | 1        |
| 6.4   | Создание собственных библиотек   | 4         | 1        | 3        |
| <b>Модуль 7. Многопоточность</b>                        |  | <b>10</b> | <b>2</b> | <b>8</b> |
| 7.1   | Работа с многопоточностью и её применение в приложении                             | 4         | 1        | 3        |
| 7.2   | Изучение принципа работы Coroutines  | 4         | 1        | 3        |
| 7.3   | Создание собственных Coroutines  | 2         | 0        | 2        |
| <b>Модуль 8. Проверка качества кода</b>                 |  | <b>8</b>  | <b>3</b> | <b>5</b> |
| 8.1   | Изучение принципов написания чистого и эффективного кода                           | 2         | 1        | 1        |
| 8.2   | Использование статических анализаторов кода и утилит для проверки качества кода    | 4         | 1        | 3        |
| 8.3   | Работа над оптимизацией производительности кода                                    | 2         | 1        | 1        |
| <b>Модуль 9. Работа со звуком и видео</b>               |  | <b>8</b>  | <b>3</b> | <b>5</b> |
| 9.1   | Изучение принципов работы с аудио и видео в Android приложениях                    | 2         | 1        | 1        |
| 9.2   | Создание мультимедийных приложений   | 4         | 1        | 3        |
| 9.3   | Работа с плеерами и кодеками   | 2         | 1        | 1        |
| <b>Модуль 10. Bluetooth и NFC</b>                       |  | <b>8</b>  | <b>2</b> | <b>6</b> |
| 10.1  | Основы работы с Bluetooth и NFC в Android /Изучение NFC-технологий и их применения | 4         | 1        | 3        |
| 10.2  | Разработка приложений для работы с Bluetooth и NFC                                 | 4         | 1        | 3        |
| <b>Модуль 11. Работа с Firebase</b>                     |  | <b>8</b>  | <b>2</b> | <b>6</b> |
| 11.1  | Изучение сервиса Firebase и его использование для разработки приложений            | 2         | 1        | 1        |
| 11.2  | Работа с базой данных Firebase   | 2         | 0        | 2        |
| 11.3  | Работа с Push уведомлениями на Firebase  | 4         | 1        | 3        |



|   |  |            |           |            |
|---|--|------------|-----------|------------|
| <b>Модуль 12. Работа с AR и VR</b>                    |  | <b>8</b>   | <b>3</b>  | <b>5</b>   |
| 12.1  | Создание AR и VR приложений  | 2          | 1         | 1          |
| 12.2  | Изучение основ использования ARCore и Vuforia в Android                        | 2          | 1         | 1          |
| 12.3  | Создание взаимодействия AR-элементов и виртуальной реальности и реального мира | 4          | 1         | 3          |
| <b>Модуль 13. Работа с камерой</b>                    |  | <b>8</b>   | <b>2</b>  | <b>6</b>   |
| 13.1  | Работа с камерой в Android   | 4          | 1         | 3          |
| 13.2  | Создание приложений для работы с камерой                                       | 4          | 1         | 3          |
| <b>Модуль 14. Работа с GPS</b>                        |  | <b>8</b>   | <b>3</b>  | <b>5</b>   |
| 14.1  | Изучение принципов работы с GPS  | 2          | 1         | 1          |
| 14.2  | Создание приложений для работы с GPS   | 4          | 1         | 3          |
| 14.3  | Изучение методов определения местоположения и работа с картами                 | 2          | 1         | 1          |
| <b>Модуль 15. Работа с фрагментами и библиотеками</b> |  | <b>8</b>   | <b>3</b>  | <b>5</b>   |
| 15.1  | Изучение создания фрагментов в Android   | 2          | 1         | 1          |
| 15.2  | Работа с библиотеками и их внедрение в проект                                  | 4          | 1         | 3          |
| 15.3  | Изучение внедрения зависимостей в проект                                       | 2          | 1         | 1          |
| <b>Модуль 16. Работа с виджетами</b>                  |  | <b>10</b>  | <b>3</b>  | <b>7</b>   |
| 16.1  | Создание виджетов для Android  | 2          | 1         | 1          |
| 16.2  | Изучение работы с системными виджетами и создание собственных                  | 4          | 1         | 3          |
| 16.3  | Оптимизация работы с батареей при работе с виджетами                           | 4          | 1         | 3          |
| <b>Модуль 17. Создание индивидуального проекта</b>    |  | <b>10</b>  | <b>2</b>  | <b>8</b>   |
| 17.1  | Разработка индивидуального проекта на Kotlin для платформы Android             | 4          | 1         | 3          |
| 17.2  | Работа с преподавателем по разработке и созданию проекта                       | 4          | 1         | 3          |
| 17.3  | Аттестация по итогам прохождения программы                                     | 2          | 0         | 2          |
| <b>ВСЕГО</b>  |  | <b>144</b> | <b>42</b> | <b>102</b> |

## 1.6. Планируемые результаты

Планируемые результаты программы "Разработка на языке Kotlin для платформы Android– Продвинутый уровень" включают:

1. Приобретение фундаментальных знаний о языке программирования Kotlin и его особенностях для разработки мобильных приложений на платформе Android.
2. Умение создавать пользовательские интерфейсы для мобильных приложений с использованием элементов и компонентов Android.
3. Навыки работы с данными, включая методы получения, обработки и хранения данных в приложении на платформе Android.
4. Умение интегрировать сторонние библиотеки и сервисы в мобильные приложения с помощью языка Kotlin.
5. Знание методов оптимизации производительности мобильных приложений и умение отлаживать и исправлять ошибки.
6. Навыки тестирования мобильных приложений и развертывания их на устройствах и в магазинах приложений.
7. Умение создавать качественные мобильные приложения на платформе Android, отвечающие требованиям и ожиданиям пользователей.

В результате программы участники будут готовы самостоятельно разрабатывать и поддерживать мобильные приложения на платформе Android с использованием языка Kotlin, а также продолжать углублять свои знания и навыки в этой области.

Метапредметные результаты:

- формирование навыков самоорганизации;
- формирование навыков сотрудничества: работа в коллективе, в команде, микро-группе;
- воспитание бережного отношения к технике;
- воспитание самостоятельности, инициативности;
- развитие навыков анализа и оценки получаемой информации.
- Личностные:
- развитие личностных качеств (активность, инициативность, воля, любознательность и т. п.);
- развитие внимания, памяти, восприятия, образного мышления;
- развитие логического и пространственного воображения;
- развитие творческих способностей и фантазии;
- развитие мотивации к познанию и творчеству;
- формирование положительных черт характера: трудолюбия, аккуратности, собранности, усидчивости, отзывчивости;
- развитие мотивации к профессиональному самоопределению.

## Раздел 2. Комплекс организационно-педагогических условий

### 2.1 календарный учебный график

| Год обучения | Всего учебных недель | Количество учебных часов | Режим занятий           |
|--------------|----------------------|--------------------------|-------------------------|
| 1 год        | 36                   | 144                      | 1 раз в неделю – 2 часа |

### 2.2 Условия реализации программы

#### Материально-техническое обеспечение:

Для реализации учебных занятий используется следующее оборудование и материалы:

- Ноутбуки, оснащенные выходом в Интернет;
- Интерактивная доска;
- Планшеты;

#### Информационное обеспечение:

- операционная система Monjaro;
- Интернет-источники;
- поддерживаемые браузеры (для работы LMS): Yandex Browser, Chrome, Chrome Mobile, Firefox, Opera;
- варианты демонстрационных программ, материалы по терминологии ПО;
- инструкции по настройке оборудования;
- учебная и техническая литература
- методические пособия, разрабатываемые преподавателем с учётом конкретных условий;
- техническая библиотека объединения, содержащая справочный материал, учебную и техническую литературу.
- обязательным является инструктаж по технике безопасности и беседы о здоровьесберегающем поведении в процессе работы на компьютере, интенсивной интеллектуальной деятельности.

**Кадровое обеспечение:** Программа реализуется Гайнановым М.В., образование среднее профессиональное ГБПОУ «Саткинский горно-керамический колледж имени А.К. Савина», 30.06.2021 Техник-программист; Свидетельство Оператор электронно-вычислительных и вычислительных машин (4 разряд); Профессиональная переподготовка ГБУ ДПО «ЧИППКРО», 03.06.2022 Право на ведение профессиональной деятельности в сфере дополнительного образования.

### 2.3. Формы аттестации

Система контроля знаний и умений обучающихся представляется в виде:

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки.

Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они

активизируют, стимулируют работу обучающихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточная аттестация осуществляется в конце I полугодия учебного года.

Форма контроля: тестирование, решение практических задач средствами языка программирования.

Аттестация по итогам освоения программы осуществляется в конце учебного года.

Форма контроля: защита проекта.

Проект является одним из видов самостоятельной работы, предусмотренной в ходе обучения по программе. Педагог-наставник оказывает консультационную помощь в выполнении проекта.

Индивидуальный (групповой) проект оценивается формируемой комиссией. Состав комиссии (не менее 3-х человек): педагог-наставник, администрация учебной организации, приветствуется привлечение IT-профессионалов, представителей высших и других учебных заведений.

Компонентами оценки индивидуального (группового) проекта являются (по мере убывания значимости): качество индивидуального проекта, отзыв руководителя проекта, уровень презентации и защиты проекта. Если проект выполнен группой обучающихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

## **2.4. Оценочные материалы**

### **Промежуточная аттестация**

Тестирование проводится с обучающимися индивидуально, во время проведения занятия.

Перечень вопросов для тестирования:

1. Что такое Kotlin? Какие основные отличия Kotlin от Java для разработки на платформе Android?
2. Расскажите о базовых синтаксических конструкциях Kotlin, таких как переменные, функции, условные операторы и циклы.
3. Каким образом можно создать пользовательский интерфейс в приложении на платформе Android с использованием Kotlin?
4. Каким образом можно работать с данными в приложении на платформе Android с использованием Kotlin? Упомяните методы получения данных, работу с базами данных и сетевыми запросами.
5. Как можно интегрировать сторонние библиотеки и сервисы в приложение, разработанное на платформе Android с использованием Kotlin?
6. Каким образом можно оптимизировать производительность мобильного приложения на платформе Android с использованием Kotlin? Укажите некоторые методы и техники.
7. Как можно отлаживать и исправлять ошибки в приложении, разработанном на платформе Android с использованием Kotlin? Упомяните инструменты и методы.
8. Как проводить тестирование мобильного приложения, созданного на платформе Android с использованием Kotlin? Укажите основные этапы и методы тестирования.
9. Каким образом можно развернуть мобильное приложение на устройствах или в магазинах приложений после его разработки на платформе Android с использованием Kotlin?

10. Какие основные преимущества и недостатки Kotlin для разработки на платформе Android вы можете выделить по сравнению с другими языками программирования, такими как Java?

### **Практическая задача:**

Приложение будет отображать статистику активности в виде графиков и диаграмм.

Вот некоторые шаги, которые можно предпринять для разработки такого приложения на платформе Android с использованием Kotlin для детей и подростков:

1. Создайте новый проект в среде разработки Android Studio.
2. Определите модель данных для активности. Создайте класс Activity с необходимыми свойствами, такими как название активности, пройденное расстояние, длительность активности и количество сожженных калорий.
3. Создайте пользовательский интерфейс для приложения. Добавьте элементы управления, такие как текстовые поля и кнопки, с помощью XML-ресурсов.
4. Создайте активити (Activity) для ввода информации о физической активности. Свяжите нужные элементы интерфейса с кодом активити с помощью функции findViewById().
5. Реализуйте методы для сохранения информации о физической активности. Например, создайте метод для добавления активности в список и сохранения списка в файл или базу данных.
6. Используйте RecyclerView для отображения списка активностей в пользовательском интерфейсе. Создайте адаптер (Adapter) для связывания данных из списка активностей с элементами списка в RecyclerView.
7. Добавьте возможность просмотра статистики активности в приложении. Реализуйте графики или диаграммы, которые отражают суммарное пройденное расстояние или количество сожженных калорий в зависимости от даты активности.
8. Убедитесь, что приложение обрабатывает события пользователя, такие как нажатие на кнопку "Добавить активность" или выбор активности из списка.
9. Проверьте работоспособность приложения на эмуляторе Android или реальном устройстве. Используйте отладчик для исправления ошибок и отслеживания работы приложения.
10. Разработайте интерфейс приложения с учетом дизайна и удобства использования, адаптированного к детям и подросткам.

### **Критерии оценивания обучающихся**

#### **Описание правил проведения аттестации:**

1. Знание теории

По результатам ответов на вопросы определяется уровень теоретической подготовки.

Уровень подготовки определяется по количеству набранных баллов.

Без ответа – 0 баллов

Ответ неверный – 1 балл

Ответ частично неверный - 2 балла

Ответ развернутый и верный 3 балла

Максимальное количество баллов за ответ 3 балла.

Максимальное количество -30 баллов

#### **Критерии оценивания:**

Высокий уровень: 30-26 баллов;

Средний уровень: 25-11 баллов;

Низкий уровень: 10-1 баллов.

1. Знание практики

По результатам выполнения практического задания уровень практической подготовки.

Уровень подготовки определяется по количеству набранных баллов. Правильное выполнение задачи 5 баллов, при частичном выполнении от 1-4 баллов. Максимальное количество -5 баллов.

**Критерии оценивания:**

Высокий уровень: 5-4 баллов;

Средний уровень: 3- 2 баллов;

Низкий уровень: 2-1 баллов.

**Аттестация по итогам освоения программы.**

Разработка и защита проекта

**Исследовательский проект**

Тема проекта:

Творческое название (при наличии):

Основополагающий вопрос:

Авторы:

1.

2.

3.

...

Предметная область:

Краткая аннотация:

Проблемные вопросы учебной темы:

Темы исследования учащихся:

Этапы выполнения проекта:

На этапе презентации участники представляют проект на обсуждение.

Этап рефлексии отводится под обсуждение итогов проекта, оценки своих действий, формулирование выводов. Для оценивания проекта могут быть разработаны специальные оценочные листы. Ниже представлен пример оценочного листа:

Таблица

**Лист оценивания проекта**

| <b>Критерий оценивания</b>                      | <b>1-я группа</b> |
|---|-------------------|
| Актуальность темы                               |                   |
| Соответствие содержания проекта заявленной теме |                   |
| Техническая сложность                           |                   |
| Оригинальность                                  |                   |
| Дизайн  |                   |
| Уровень проработанности проекта                 |                   |
| Итоговое количество баллов                      |                   |

**Система оценки результатов освоения программы**

Предметом диагностики и контроля являются внешние образовательные продукты учащихся, а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам программы. Основой для оценивания деятельности учащихся являются результаты анализа его продукции, деятельности по ее созданию, уровень защиты проекта на конференции.

Оценке подлежит в первую очередь уровень достижения учеником минимально необходимых результатов, обозначенных в целях и задачах программы.

Оцениванию подлежат также те направления и результаты деятельности учащихся, которые определены в рабочей программе педагога и в индивидуальных образовательных маршрутах учащихся (при наличии таковых).

| Критерии оценки защиты проекта и уровня выполнения работы учащимся  | Оценка            |
|---|-------------------|
| Проект полный, оригинальный, обладает степенью новизны и практической пользы, не содержит ошибок. Учащийся способен обеспечить подачу проекта целевой аудитории, обобщить материал, сделать собственные выводы, выразить свое мнение, привести примеры, ответить на вопросы по теме проекта.      | отлично           |
| Проект полный, обладает оригинальностью, и практической пользой, не содержит значительных ошибок. Учащийся способен обеспечить подачу проекта целевой аудитории, сделать собственные выводы, ответить на вопросы по теме проекта. Собственное мнение по теме проекта недостаточно четко выражено. | хорошо            |
| Проект типовой, не содержит значительных ошибок. Не обладает лаконичностью. Есть ошибки в деталях и/или они просто отсутствуют. подача проекта сумбурная. Мнение по теме проекта сформировано частично. Затрудняется с ответами по теме проекта.  | удовлетворительно |

## 2.5 Методические материалы

Организация образовательного процесса в данной программе происходит в очной форме обучения, с возможностью применения дистанционных технологий, и групповой форме.

При реализации программы используются различные методы обучения:

- Объяснительно-иллюстративный (предъявление информации различными способами (объяснение, рассказ, беседа, инструктаж, демонстрация, работа с технологическими картами и др.);
- Проблемный (постановка проблемы и самостоятельный поиск её решения обучающимися);
- Репродуктивный (воспроизводство знаний и способов деятельности по аналогу);
- Метод проектов (технология организации образовательных ситуаций, в которых учащийся ставит и решает собственные задачи).

- методы обучения (словесный, наглядный, практический, объяснительно-иллюстративный, интегрированный, метод сравнения, репродуктивный, частично-поисковый, аналитический, дедуктивный, исследовательский, проблемный, игровой, дискуссионный, проектный и др.) и воспитания (убеждение, поощрение, упражнение, стимулирование, мотивация, метод положительного примера и др.);

- формы организации образовательного процесса: индивидуальная, индивидуально-групповая и групповая; выбор той или иной формы обосновывается с позиции профиля деятельности (технического).;

-формы организации учебного занятия – беседа, выставка, защита проектов, конкурс, мастер-класс, «мозговой штурм», наблюдение, олимпиада, открытое занятие, практическое занятие, презентация, семинар, соревнование, объяснение материала, моделирование и др.;

-образовательные (педагогические) технологии – технология индивидуализации обучения, технология группового обучения, технология коллективного взаимообучения, технология программированного обучения, технология модульного обучения, технология блочно-модульного обучения, технология дифференцированного обучения, технология разноуровневого обучения, технология развивающего обучения, технология проблемного обучения, технология дистанционного обучения, технология проектной деятельности, технология коллективной творческой деятельности, технология решения изобретательских задач, здоровьесберегающая технология.

## **2.6 Воспитательные компоненты**

Общей целью воспитания является формирование у обучающихся духовно-нравственных ценностей, способности к осуществлению ответственного выбора собственной индивидуальной образовательной траектории, способности к успешной социализации в обществе.

### **Задачи воспитания:**

- поддерживать и развивать традиции учреждения, коллективные творческие формы деятельности, реализовать воспитательные возможности ключевых дел, формировать у обучающихся чувство солидарности и принадлежности к образовательному учреждению;
- реализовывать воспитательный потенциал общеобразовательных общеразвивающих программ и возможности учебного занятия и других форм образовательных событий;
- развивать социальное партнерство как один из способов достижения эффективности воспитательной деятельности.
- организовывать работу с семьями обучающихся, их родителями или законными представителями, активно их включать в образовательный процесс, содействовать формированию позиции союзников в решении воспитательных задач;
- использовать в воспитании детей возможности занятий по дополнительным общеобразовательным общеразвивающим программам как источник поддержки и развития интереса к познанию и творчеству;
- содействовать приобретению опыта личностного и профессионального самоопределения на основе личностных проб в совместной деятельности и социальных практиках;
- формировать сознательное отношение обучающихся к своей жизни, здоровью, здоровому образу жизни, а также к жизни и здоровью окружающих людей.
- создавать инновационную среду, формирующую у детей и подростков изобретательское, креативное, критическое мышление через освоение дополнительных общеобразовательных общеразвивающих программ нового поколения в области инженерных и цифровых технологий;
- повышать разнообразие образовательных возможностей при построении индивидуальных образовательных траекторий (маршрутов) обучающихся;
- оптимизировать систему выявления, поддержки и развития способностей и талантов у детей и подростков, направленной на самоопределение и профессиональную ориентацию обучающихся.

### **Направления воспитательной работы**

- Основными направлениями воспитательной работы являются:
- воспитывать аккуратность и дисциплинированность при выполнении работы;
- способствовать формированию положительной мотивации к трудовой деятельности;
- развивать основы коммуникативных отношений внутри проектных групп и в коллективе в целом;
- воспитывать трудолюбие, уважение к труду;
- развивать навыки отношений делового сотрудничества, взаимоуважения.

### **Работа с родителями**

Работа с родителями обучающихся в себя:



- организацию системы индивидуальной и коллективной работы (тематические беседы, собрания, индивидуальные консультации);
- содействие сплочению родительского коллектива и вовлечение родителей в жизнедеятельность детского объединения (организация и проведение открытых занятий, мероприятий в течение учебного года);

### Примерный перечень мероприятий

| Сроки  | Уровень проведения соревнований | Название соревнований, конкурсов, мероприятий  |
|--|---------------------------------|--|
| <b>1. Модуль « Воспитывающая среда»</b>  |                                 |  |
| 01.09.2023   | на уровне учреждения            | «День знаний»  |
| декабрь  | на уровне учреждения            | «КвантоЕлка»   |
| февраль  | муниципальный                   | Конкурс 3D моделей приуроченных к 23 февраля   |
| март   | муниципальный                   | конкурс по созданию видео открыток в среде "Подарок Маме"  |
| апрель   | муниципальный                   | конкурс рисунков ко дню Космонавтики   |
| май  | на уровне учреждения            | Организация выставки с достижениями детей  |
| <b>2. Модуль « Учебное занятие»</b>  |                                 |  |
| в течение года   | муниципальный                   | «Урок цифры»   |
| май  | муниципальный                   | «Урок Победы»  |
| декабрь-январь   | региональный                    | «Технологический диктант»  |
| февраль  | на уровне учреждения            | «День науки»   |
| <b>3.Модуль «Руководство детским объединением (направлением ) и взаимодействие с родителями»</b> |                                 |  |
| сентябрь, май  | на уровне учреждения            | Родительские собрание, мастер-классы   |
| <b>4.Модуль «Проектная деятельность»</b>   |                                 |  |
| декабрь, май   | на уровне учреждения            | «Ярмарка проектов»   |
| <b>5.Модуль «Профориентационная работа и наставничество»</b>                                     |                                 |  |
| апрель   | на уровне учреждения            | Дни открытых дверей в СУЗе   |
| <b>6.Модуль «Социальное партнерство и сетевое взаимодействие»</b>                                |                                 |  |
| ноябрь-май   | Региональный                    | Участие в конкурсе инженерных команд «Инженерные кадры России» и «Икаренок»  |
| сроки, указанные в проекте   | муниципальный                   | Проекты, совместно разрабатываемые и реализуемые обучающимися, педагогами с организациями-партнерами различной направленности            |
| <b>7.Модуль «Каникулы»</b>   |                                 |  |
| ноябрь, январь, март, июнь   | муниципальный                   | Онлайн-лагерь в дни школьных каникул   |
| <b>8.Модуль «Профилактика и безопасность»</b>  |                                 |  |
| сентябрь   | на уровне учреждения            | Проведение «Урока безопасности и навыков безопасного поведения в Интернете, информационной безопасности, повышение правовой грамотности» |
| сентябрь   | на уровне учреждения            | Проведение инструктажа по безопасности и охране жизни и здоровья   |

## 2.7. Информационные ресурсы и литература

### **Литература для педагогов:**

1. "Разработка мобильных приложений на платформе Android с использованием Kotlin: методические рекомендации для педагогов" - автор Баранова О.В. (доступно в электронном виде).

### **Литература для обучающихся:**

1. "Kotlin для начинающих: подробное руководство по изучению языка программирования Kotlin" - автор Миртл Ж.

2. "Программирование для Android на языке Kotlin" - автор Гасков А.

3. "Разработка мобильных приложений на платформе Android с использованием Kotlin" - автор Самохвалов А.

4. "Android разработка на языке Kotlin: руководство для начинающих" - автор Смит Д.

### **Электронные ресурсы:**

1. Официальный сайт Kotlin - <https://kotlinlang.org/>

2. Официальный сайт Android Developers - <https://developer.android.com/>

3. Kotlin на платформе Android: документация - <https://developer.android.com/kotlin/>

4. Kotlin Android Extensions: документация - <https://developer.android.com/topic/libraries/view-binding/>

5. Видеокурсы по Kotlin для Android-разработки на YouTube - пример: "Kotlin for Android Developers" от Филиппа Ляцко.

6. Kotlin Android Development Masterclass на Udemy - <https://www.udemy.com/course/kotlin-android/>

## Приложение

### Приложение 1

#### ОЦЕНОЧНЫЙ ЛИСТ ВХОДНОЙ ДИАГНОСТИКИ УЧАЩИХСЯ \_\_\_\_\_ учебный год

Входная диагностика учащихся объединения: Мобильная разработка

Наименование образовательной программы: Разработка на языке Kotlin для платформы Android– Продвинутый уровень(14-18 лет)

Фамилия, имя, отчество педагога: Гайнанов Максим Вячеславович

Дата проведения: \_\_\_\_\_

Форма проведения: Наблюдение, беседа

Форма оценки результатов: уровень (высокий, средний, низкий)

**3 балла (высокий уровень)** – высокий уровень развития компетенции. Обучающийся (его знания, умения) выделяются на общем фоне своей успешностью (оригинальностью, качеством).

**2 балла (средний уровень)** – промежуточный уровень.

**1 балл (низкий уровень)** – трудности в понимании заданий и учебного материала; низкий уровень развития компетенции, недостаточная активность

#### Результаты итоговой аттестации

| №  | Фамилия<br>ребенка | имя | 1 | 2 | 3 | Итого | Результат |
|----|--------------------|-----|---|---|---|-------|-----------|
| 1  |                    |     |   |   |   |       |           |
| 2  |                    |     |   |   |   |       |           |
| 3  |                    |     |   |   |   |       |           |
| 4  |                    |     |   |   |   |       |           |
| 5  |                    |     |   |   |   |       |           |
| 6  |                    |     |   |   |   |       |           |
| 7  |                    |     |   |   |   |       |           |
| 8  |                    |     |   |   |   |       |           |
| 9  |                    |     |   |   |   |       |           |
| 10 |                    |     |   |   |   |       |           |

|  |  |
|--|--|
| Всего аттестовано 12 воспитанников.<br>Из них по результатам аттестации:           |  |
| высокий уровень - __ чел.<br>средний уровень - __ чел.<br>низкий уровень - __ чел. |  |
| <b>Подпись педагога:</b> Гайнанов Максим Вячеславович                              |  |
| <b>Подписи членов аттестационной комиссии</b>                                      |  |
| А.В. Михайлов - руководитель ЦЦОД «IT-куб» г.Сатка;                                |  |
| Н.В. Кириченко – заведующий учебной частью ЦЦОД «IT-куб» г.Сатка;                  |  |
| Э. И. Макагон – методист ЦЦОД «IT-куб» г.Сатка                                     |  |

**ОЦЕНОЧНЫЙ ЛИСТ  
ПРОМЕЖУТОЧНОЙ ДИАГНОСТИКИ УЧАЩИХСЯ**  
\_\_\_\_\_ учебный год

Входная диагностика учащихся объединения: Мобильная разработка

Наименование образовательной программы: Разработка на языке Kotlin для платформы Android– Продвинутый уровень(14-18 лет)

Фамилия, имя, отчество педагога: Гайнанов Максим Вячеславович

Дата проведения: \_\_\_\_\_

Форма проведения: Решение практических задач

Форма оценки результатов: уровень (высокий, средний, низкий)

**3 балла (высокий уровень)** – высокий уровень развития компетенции. Обучающийся (его знания, умения) выделяются на общем фоне своей успешностью (оригинальностью, качеством).

**2 балла (средний уровень)** – промежуточный уровень.

**1 балл (низкий уровень)** – трудности в понимании заданий и учебного материала; низкий уровень развития компетенции, недостаточная активность

**Результаты итоговой аттестации**

| №  | Фамилия<br>ребенка | имя | 1 | 2 | 3 | Итого | Результат |
|----|--------------------|-----|---|---|---|-------|-----------|
| 1  |                    |     |   |   |   |       |           |
| 2  |                    |     |   |   |   |       |           |
| 3  |                    |     |   |   |   |       |           |
| 4  |                    |     |   |   |   |       |           |
| 5  |                    |     |   |   |   |       |           |
| 6  |                    |     |   |   |   |       |           |
| 7  |                    |     |   |   |   |       |           |
| 8  |                    |     |   |   |   |       |           |
| 9  |                    |     |   |   |   |       |           |
| 10 |                    |     |   |   |   |       |           |

|  |  |
|--|--|
| Всего аттестовано 12 воспитанников.<br>Из них по результатам аттестации:           |  |
| высокий уровень - __ чел.<br>средний уровень - __ чел.<br>низкий уровень - __ чел. |  |
| <b>Подпись педагога:</b> Гайнанов Максим Вячеславович                              |  |
| <b>Подписи членов аттестационной комиссии</b>                                      |  |
| А.В. Михайлов - руководитель ЦЦОД «IT-куб» г.Сатка;                                |  |
| Н.В. Кириченко – заведующий учебной частью ЦЦОД «IT-куб» г.Сатка;                  |  |
| Э. И. Макагон – методист ЦЦОД «IT-куб» г.Сатка                                     |  |

**План**  
**педагога Гайнанова Максима Вячеславовича**  
**по проведению аттестации по итогам освоения программы**  
**по дополнительной общеобразовательной общеразвивающей программе «Разработка**  
**на языке Kotlin для платформы Android– Продвинутый уровень»**

**Сроки проведения:**

**Вид аттестации:** аттестация по итогам освоения программы

**Цель итоговой аттестации:** оценка качества усвоения обучающимися содержания образовательной программы в конце учебного года.

**Форма проведения:** защита проектов

**Форма оценки, уровень усвоения программы:** высокий, средний, низкий.

**Правила проведения аттестации:** критерии оценки результата.

**Описание правил проведения аттестации:**

Модель реализации исследовательских проектов обучающихся

| № | Этапы реализации проекта                             | Примерные виды деятельности   |
|---|--|---|
| 1 | Организационный (подготовка). Текущая рефлексия      | Определение темы проекта. Разработка плана реализации. Обсуждение   |
| 2 | Планирование   | Корректировка маршрута. Совместные исследования   |
| 3 | Поиск  | Поиск информации в мультимедийной энциклопедии, справочнике, сети Интернет, электронном каталоге  |
| 4 | Промежуточные результаты и выводы. Текущая рефлексия | Обработка информации и полученных данных с использованием электронных шаблонов; создание отчета о проделанной работе, презентации, альбома и др. Обсуждение |
| 5 | Защита проекта. Рефлексия результатов                | Демонстрация отчета о проделанной работе, вручение грамот, дипломов. Обсуждение итогов  |

**Правила выбора проекта:**

1. Тема должна быть интересна обучающемуся, должна увлекать его. Исследовательская работа эффективна только на добровольной основе. Тема, навязанная, какой бы важной она ни казалась педагогу не даст должного эффекта. Вместо живого увлекательного поиска обучающийся будет чувствовать себя вовлеченным в очередное скучное мероприятие.
2. Тема должна быть выполнима, решение ее должно быть полезно участникам исследования. Натолкнуть обучающегося на ту идею, в которой он максимально реализуется как исследователь, раскроет лучшие стороны своего интеллекта, получит новые полезные знания, умения и навыки, – сложная, но необходимая задача для работы педагога.

3. Учитывая интересы обучающихся, необходимо держаться ближе к той сфере, в которой лучше всего разбираетесь, в которой чувствуете себя сильным. Увлечь другого может лишь тот, кто увлечен сам.
4. Тема должна быть оригинальной с элементами неожиданности, необычности. Оригинальность следует понимать как способность нестандартно смотреть на традиционные предметы и явления.
5. Тема должна быть такой, чтобы работа могла быть выполнена относительно быстро.
6. Тема должна быть доступной. Она должна соответствовать возрастным особенностям обучающихся. Это касается не только выбора темы исследования, но и формулировки и отбора материала для ее решения.
7. Сочетание желаний и возможностей. Выбирая тему, педагог должен учесть наличие требуемых средств и материалов – исследовательской базы. Ее отсутствие, невозможность собрать необходимые данные обычно приводят к поверхностному решению.
8. С выбором темы не стоит затягивать. Большинство обучающихся не имеют постоянных пристрастий, их интересы ситуативны. Поэтому, выбирая тему, действовать следует быстро, пока интерес не угас.

#### План рассказа о проекте

1. поприветствовать аудиторию. Представиться. Озвучить тему проекта.
2. Озвучить тему, актуальность, цели и задачи проекта.
3. Рассказать о выбранном наборе данных: источник, структура, размер.
4. Рассказать об использованных подходах, моделях и методах: причины выбора, структура, принцип работы.
5. Дать оценку качества работы модели по выбранным критериям.
6. Привести примеры работы модели.
7. В выводах озвучить, насколько достигнута поставленная цель и как усовершенствовать модель.
8. Поблагодарить за внимание.
9. Ответить на вопросы аудитории.

Общие критерии оценки проекта можно представить так:

#### Высокий уровень – (16-20 баллов)

1. Продукт отличается сложностью
2. Правильно поняты и сформулированы цель, задачи выполнения проекта;
3. Проект оформлен в соответствии с требованиями. Обучающийся владеет

специальными терминами и понятиями.

1. Проявлены творчество, инициатива;
2. Предъявленный продукт деятельности отличается высоким качеством

исполнения;

#### Средний уровень – (9-15 баллов):

1. Правильно поняты цель, задачи выполнения проекта;
2. Проект оформлен в соответствии с требованиями. Обучающийся владеет

специальными терминами и понятиями, но имеются 1-2 ошибки в этапах, в представлении продукта;

1. Самостоятельность проявлена на недостаточном уровне.

Низкий уровень – (8-1 балл)

Не набрано минимальное количество баллов, установленное комиссией, принимающей защиту; проект не выполнен или не завершен