**Инструмент разработки мобильных приложений: Visual Studio, JDK, Android Studio, установка, компоненты, настройка, структура проекта, запуск приложения.**

**Visual Studio**

**Установка**

Перейдите на официальный сайт Visual Studio и скачайте установочный файл.

Запустите установщик и следуйте инструкциям мастера установки.

Выберите рабочую нагрузку "Mobile development with .NET" (Разработка мобильных приложений с .NET), которая включает Xamarin.

**Компоненты**

Xamarin: фреймворк для кроссплатформенной разработки мобильных приложений.

.NET Core: платформа для создания серверной части приложений.

NuGet: система управления пакетами для .NET.

**Настройка**

Откройте Visual Studio и создайте новый проект.

Выберите шаблон для мобильного приложения (например, Xamarin.Forms App).

Установите необходимые пакеты с помощью NuGet Package Manager

Настройте проект, указав целевые платформы (Android, iOS) и версии SDK.

**Структура проекта**

MainActivity.cs: файл, содержащий главную активность приложения для Android.

App.xamlи App.xaml.cs: файлы, определяющие общие настройки и инициализацию приложения.

Resources: директория, содержащая ресурсы, такие как изображения, макеты, стили.

XAML файлы: файлы разметки интерфейса пользователя.

**Запуск приложения**

Подключите физическое устройство или настройте эмулятор Android.

Выберите целевое устройство в Visual Studio.

Нажмите кнопку "Run" (Запустить) или используйте сочетание клавиш Ctrl+F5 для запуска приложения.

JDK (Java Development Kit)

**Установка**

Перейдите на официальный сайт Oracle и скачайте последнюю версию JDK.

Установите JDK, следуя инструкциям мастера установки.

Настройте переменные среды:

Добавьте путь к директории установки JDK в переменную среды PATH.

Создайте переменную среды JAVA\_HOME, указывающую на директорию установки JDK.

**Компоненты**

javac: компилятор, преобразующий исходный код Java в байт-код

java: интерпретатор, выполняющий байт-код Java.

JRE: среда выполнения Java-приложений.

**Настройка**

Убедитесь, что JDK корректно установлен и доступен через командную строку.

Настройте переменные среды PATH и JAVA\_HOME.

**Структура проекта**

src: директория для хранения исходного кода Java.

bin: директория для хранения скомпилированных классов.

lib: директория для хранения внешних библиотек.

**Запуск приложения**

Напишите исходный код Java и сохраните его в файле с расширением .java.

Скомпилируйте код с помощью команды javac MyApp.java.

Запустите приложение с помощью команды java MyApp.

**Android Studio**

**Установка**

Перейдите на официальный сайт Android Studio и скачайте установочный файл.

Запустите установщик и следуйте инструкциям мастера установки.

Установите необходимые компоненты, такие как Android SDK, эмуляторы и другие инструменты разработки.

**Компоненты**

Android SDK: набор инструментов и библиотек для разработки Android-приложений.

Emulator: виртуальные устройства для тестирования приложений.

Gradle: система сборки для автоматизации процесса сборки и управления зависимостями.

**Настройка**

Создайте новый проект в Android Studio.

Выберите шаблон проекта (например, Empty Activity) и настройте параметры, такие как минимальная и целевая версия SDK.

Настройте файлы сборки Gradle для добавления зависимостей и плагинов.

**Структура проекта**

app: основной модуль приложения, содержащий весь код и ресурсы.

src: исходный код приложения.

main: основной код и ресурсы.

java: директория для файлов Java или Kotlin.

res: директория для ресурсов (макеты, изображения, строки).

build.gradle: файлы конфигурации сборки для проекта и модулей.

**Запуск приложения**

Подключите физическое устройство или настройте эмулятор Android.

Выберите целевое устройство в Android Studio.

Нажмите кнопку "Run" (Запустить) или используйте сочетание клавиш Shift+F10 для запуска приложения.

**\_\_\_\_\_**

**Android Studio** - это официальная интегрированная среда разработки (IDE) для Android. Она построена на базе IntelliJ IDEA и предоставляет все необходимые инструменты для проектирования, разработки, отладки и тестирования Android-приложений.

**1. Установка**

Требования к системе: Убедитесь, что ваш компьютер соответствует минимальным системным требованиям Android Studio. Обычно это Windows, macOS или Linux, с достаточным объемом оперативной памяти и свободного места на диске.

Скачивание: Скачайте последнюю версию Android Studio с официального сайта: https://developer.android.com/studio

Установка: Запустите скачанный установщик и следуйте инструкциям на экране. Во время установки вам будет предложено установить компоненты, такие как Android SDK.

**2. Компоненты**

Android Studio включает в себя несколько ключевых компонентов:

IDE (Integrated Development Environment): Основной интерфейс для написания кода, проектирования UI, отладки и управления проектом.

Android SDK (Software Development Kit): Набор инструментов, библиотек и документации, необходимых для разработки приложений под Android. SDK содержит:

Платформы Android: Конкретные версии Android (например, Android 13, Android 14), включая API, библиотеки и системные образы.

Build Tools: Инструменты для компиляции, сборки и упаковки вашего приложения.

Android Emulator: Эмулятор Android-устройств, позволяющий тестировать приложения на компьютере без необходимости использования физического устройства.

ADB (Android Debug Bridge): Инструмент командной строки для взаимодействия с Android-устройством или эмулятором.

Gradle: Система сборки, которая автоматизирует процесс компиляции, упаковки и развертывания вашего приложения.

SDK Manager: Инструмент для управления установленными версиями Android SDK, Build Tools и других компонентов.

AVD Manager (Android Virtual Device Manager): Инструмент для создания и управления виртуальными устройствами (эмуляторами).

**3. Настройка Android Studio**

SDK Manager:

Откройте SDK Manager (Tools -> SDK Manager).

Установите необходимые платформы Android (например, последнюю версию и версию, на которую вы ориентируетесь для поддержки более старых устройств).

Установите необходимые Build Tools (рекомендуется последняя стабильная версия).

В разделе “SDK Tools” убедитесь, что установлены:

Android SDK Build-Tools

Android SDK Command-line Tools (latest)

Android Emulator

Android SDK Platform-Tools

AVD Manager:

Откройте AVD Manager (Tools -> AVD Manager).

Создайте виртуальное устройство, выбрав аппаратную конфигурацию (например, Pixel 6), версию Android и другие параметры.

**4. Структура проекта Android Studio**

Когда вы создаете новый проект в Android Studio, создается определенная структура каталогов и файлов. Вот основные элементы:

app/: Содержит основной код вашего приложения.

src/: Содержит исходный код.

main/: Основной набор исходного кода.

java/: Содержит Java или Kotlin код.

res/: Содержит ресурсы приложения (изображения, макеты, строки, стили, темы и т.д.).

drawable/: Изображения и другие графические ресурсы.

layout/: XML-файлы, описывающие пользовательский интерфейс (UI).

values/: XML-файлы, содержащие значения (строки, цвета, размеры и т.д.).

mipmap/: Иконки приложения для разных разрешений экрана.

AndroidManifest.xml: Файл манифеста, содержащий метаданные о вашем приложении (название, иконка, разрешения, компоненты и т.д.).

test/: Юнит-тесты.

androidTest/: Инструментальные тесты.

build.gradle: Файл конфигурации Gradle для модуля приложения.

gradle/: Содержит обертки Gradle.

build.gradle: Файл конфигурации Gradle для всего проекта.

settings.gradle: Определяет, какие модули включены в проект.

**5. Запуск приложения**

Подключите устройство (если используете физическое устройство): Убедитесь, что на вашем Android-устройстве включена отладка по USB (Developer Options). Подключите устройство к компьютеру через USB. Разрешите отладку USB на устройстве, когда появится запрос.

Выберите устройство: В Android Studio в выпадающем списке устройств выберите ваше подключенное устройство или запущенный эмулятор.

Запустите приложение: Нажмите кнопку “Run” (зеленый треугольник) или выберите “Run ‘app’” в меню “Run”.

Android Studio скомпилирует ваше приложение, установит его на выбранное устройство и запустит.

**Visual Studio** - это мощная интегрированная среда разработки (IDE) от Microsoft, которая позволяет создавать широкий спектр приложений, включая мобильные приложения. Она поддерживает разработку мобильных приложений для различных платформ, таких как Android, iOS и Windows, используя разные технологии, включая Xamarin, .NET MAUI (Multipurpose App UI), Cordova и другие.

**1. Установка Visual Studio:**

Загрузка: Загрузите последнюю версию Visual Studio с официального сайта Microsoft: https://visualstudio.microsoft.com/

Выбор версии: Visual Studio предлагает несколько версий:

Community: Бесплатная версия для индивидуальных разработчиков, студентов, разработчиков с открытым исходным кодом и небольших команд.

Professional: Платная версия с расширенными функциями для профессиональных разработчиков и команд.

Enterprise: Платная версия с самыми полными возможностями для крупных предприятий.

Запуск установщика: Запустите скачанный установочный файл.

Выбор рабочих нагрузок: Это ключевой шаг! Выберите рабочие нагрузки, необходимые для разработки мобильных приложений. В зависимости от выбранной технологии, вам могут потребоваться:

.NET Multi-platform App UI development (или .NET MAUI development): Для разработки кроссплатформенных приложений с единой кодовой базой для Android, iOS, Windows и macOS. (Рекомендуется для новых проектов)

Mobile development with .NET (Xamarin): Для разработки кроссплатформенных приложений с использованием C# и Xamarin. (Хорошо для поддержки существующих Xamarin проектов).

Universal Windows Platform development: Для разработки приложений для Windows, включая мобильные устройства.

ASP.NET and web development: Может быть полезно, если ваше мобильное приложение взаимодействует с веб-сервисами или API.

Development with C++: Может понадобиться, если нужно использовать нативный код (например, для производительности или доступа к специфичным API).

Выбор отдельных компонентов: После выбора рабочих нагрузок, установщик предложит выбрать отдельные компоненты. Просмотрите список и убедитесь, что у вас установлены необходимые SDK, эмуляторы и инструменты для выбранных вами платформ. Например:

Android SDK и эмуляторы: Необходимы для разработки и тестирования приложений для Android.

iOS Build Agent: Нужен для сборки iOS приложений, требует наличия компьютера Mac с установленным Xcode.

.NET SDK: Версия .NET SDK, которая соответствует вашей выбранной рабочей нагрузке.

Git for Windows: Рекомендуется для контроля версий.

Установка: Нажмите кнопку “Установить” и дождитесь завершения процесса установки. Это может занять некоторое время, в зависимости от скорости вашего интернета и выбранных компонентов.

**2. Компоненты Visual Studio для разработки мобильных приложений:**

Редактор кода: Предоставляет инструменты для написания, редактирования и отладки кода.

Интеллектуальные возможности: Поддержка IntelliSense (автозавершение кода, подсказки), рефакторинга и других функций, облегчающих процесс разработки.

Отладчик: Позволяет находить и исправлять ошибки в коде.

Дизайнер пользовательского интерфейса (UI): Визуальные инструменты для создания пользовательского интерфейса (XAML, XML). Может отличаться в зависимости от выбранной технологии.

Эмуляторы и симуляторы: Позволяют тестировать приложения на виртуальных устройствах без необходимости использования физических устройств.

Система сборки: Автоматизирует процесс компиляции, линковки и создания пакетов приложений.

Инструменты для профилирования: Помогают оптимизировать производительность приложений.

Интеграция с системами контроля версий: Поддержка Git, SVN и других систем контроля версий.

Пакет SDK: Включает библиотеки, инструменты и документацию, необходимые для разработки для определенной платформы (Android SDK, iOS SDK, Windows SDK).

**3. Настройка Visual Studio:**

Вход в учетную запись Microsoft: Рекомендуется войти в свою учетную запись Microsoft для синхронизации настроек и использования облачных сервисов.

Настройка тем и цветовых схем: Персонализируйте внешний вид Visual Studio, чтобы сделать его более удобным для вас.

Установка расширений: Visual Studio поддерживает расширения, которые добавляют новые функции и инструменты. Найдите и установите расширения, которые могут быть полезны для вашей работы. Примеры: Resharper (платный), Productivity Power Tools.

Настройка горячих клавиш: Настройте горячие клавиши для часто используемых команд, чтобы ускорить процесс разработки.

Настройка эмуляторов: Убедитесь, что эмуляторы настроены правильно и работают корректно. Создайте эмуляторы с разными разрешениями экрана и версиями Android/iOS.

**4. Структура проекта:**

Структура проекта в Visual Studio может варьироваться в зависимости от выбранной технологии разработки мобильных приложений. Вот пример структуры проекта для .NET MAUI:

MyMauiApp/

├── MyMauiApp.csproj // Файл проекта (основной файл)

├── App.xaml // XAML файл приложения

├── App.xaml.cs // C# код для App.xaml

├── MainPage.xaml // XAML файл главной страницы

├── MainPage.xaml.cs // C# код для MainPage.xaml

├── Resources/ // Ресурсы приложения (изображения, шрифты и т.д.)

│ ├── Images/

│ │ └── ...

│ ├── Raw/

│ │ └── ...

│ ├── Styles/ // Стили CSS

│ │ └── ...

│ └── ...

├── Platforms/ // Специфичные для платформы файлы

│ ├── Android/ // Файлы для Android

│ │ ├── MainActivity.cs

│ │ ├── Resources/

│ │ │ └── ...

│ │ └── ...

│ ├── iOS/ // Файлы для iOS

│ │ ├── AppDelegate.cs

│ │ ├── Info.plist

│ │ └── ...

│ └── Windows/ // Файлы для Windows

│ ├── ...

├── Dependencies/ // Зависимости проекта (NuGet пакеты)

├── Properties/ // Свойства проекта

├── obj/ // Объектные файлы (сгенерированные файлы сборки)

├── bin/ // Скомпилированные файлы приложения

└── ...

**Основные компоненты:**

Файл проекта (.csproj): Содержит информацию о проекте, включая зависимости, настройки сборки и целевые платформы.

XAML файлы: Описывают пользовательский интерфейс.

C# файлы: Содержат логику приложения.

Ресурсы: Изображения, звуки, шрифты и другие статические файлы.

Платформы: Содержит платформо-специфичный код и настройки.

Dependencies: NuGet пакеты, используемые в проекте.

**5. Запуск приложения:**

Выбор целевой платформы: Выберите платформу (Android, iOS, Windows) и устройство/эмулятор, на котором вы хотите запустить приложение. Это делается в выпадающем списке рядом с кнопкой “Start” (или “Запуск”).

Сборка проекта: Visual Studio автоматически собирает проект перед запуском. Вы можете выполнить сборку вручную, выбрав пункт “Build” -> “Build Solution” в меню.

Запуск приложения: Нажмите кнопку “Start” (или “Запуск”) или нажмите клавишу F5. Visual Studio развернет приложение на выбранном устройстве/эмуляторе и запустит его.

Отладка: Во время выполнения приложения вы можете использовать отладчик Visual Studio для просмотра переменных, установки точек останова и пошагового выполнения кода.

**JDK (Java Development Kit)** является необходимым инструментом для разработки мобильных приложений на Android, так как Android SDK использует Java в качестве основного языка программирования (хотя сейчас также широко используется Kotlin). JDK предоставляет среду выполнения Java (JRE) и инструменты, необходимые для компиляции и запуска Java кода.

**1. Установка JDK:**

Скачать JDK:

Перейдите на сайт Oracle: https://www.oracle.com/java/technologies/javase-downloads.html

Найдите последнюю версию JDK (например, Java SE 21).

Выберите версию, подходящую для вашей операционной системы (Windows, macOS, Linux) и архитектуры (x64, x86).

Примите лицензионное соглашение и скачайте установщик. Может потребоваться учетная запись Oracle.

Альтернативно, можно использовать OpenJDK, который является бесплатным и с открытым исходным кодом. Например, AdoptOpenJDK (теперь Eclipse Temurin) или другие дистрибутивы. Ссылка: https://adoptium.net/

Установить JDK:

Запустите скачанный установщик.

Следуйте инструкциям на экране. Рекомендуется установить JDK в стандартную директорию.

Запомните директорию, в которую была установлена JDK, она понадобится для настройки переменных среды. Например, C:\Program Files\Java\jdk-21 (путь может отличаться в зависимости от версии и системы).

**2. Компоненты JDK:**

Основные компоненты JDK включают:

javac: Компилятор Java. Преобразует исходный код Java (.java файлы) в байт-код (.class файлы).

java: Интерпретатор Java. Запускает байт-код Java.

jar: Инструмент для создания и управления JAR (Java Archive) файлами.

javadoc: Инструмент для генерации документации API из комментариев в исходном коде Java.

JRE (Java Runtime Environment): Среда выполнения Java, необходимая для запуска Java-приложений. Входит в состав JDK.

Другие утилиты: jdb (отладчик), jprofiler (профилировщик), keytool (управление ключами), и т.д.

**3. Настройка JDK:**

После установки необходимо настроить переменные среды операционной системы, чтобы система могла найти инструменты JDK.

JAVA\_HOME: Указывает на директорию установки JDK.

PATH: Добавляет директорию bin из директории JDK в переменную PATH, чтобы можно было запускать команды javac и java из командной строки.

Настройка в Windows:

Найти путь к JDK: Откройте проводник и найдите директорию установки JDK (например, C:\Program Files\Java\jdk-21).

Открыть системные свойства: Правой кнопкой мыши на “Этот компьютер” (или “Мой компьютер”) -> “Свойства” -> “Дополнительные параметры системы”.

Переменные среды: В окне “Свойства системы” нажмите кнопку “Переменные среды”.

Создать JAVA\_HOME: В разделе “Системные переменные” нажмите “Создать…”. В поле “Имя переменной” введите JAVA\_HOME, а в поле “Значение переменной” укажите путь к директории JDK (например, C:\Program Files\Java\jdk-21).

Редактировать PATH: В разделе “Системные переменные” найдите переменную PATH и нажмите “Изменить…”.

Если список значений представлен отдельными строками: Добавьте новую строку с путем к директории bin JDK (например, %JAVA\_HOME%\bin).

Если список значений представлен одной строкой, разделенной точками с запятой: Добавьте в конец строки ;%JAVA\_HOME%\bin.

Применить изменения: Нажмите “ОК” во всех окнах.

Проверить установку: Откройте командную строку (cmd) и введите java -version. Должна отобразиться информация о установленной версии Java. Также попробуйте javac -version.

Настройка в macOS/Linux:

Найти путь к JDK: Откройте терминал и введите which javac. Если JDK уже установлена, команда вернет путь к компилятору. Если нет, установите JDK через менеджер пакетов (например, brew install openjdk на macOS с использованием Homebrew).

Редактировать файл .bash\_profile или .zshrc: Откройте файл .bash\_profile (если используете Bash) или .zshrc (если используете Zsh) в текстовом редакторе (например, nano ~/.bash\_profile). Файл находится в вашей домашней директории.

Добавить переменные среды: Добавьте следующие строки в файл:

export JAVA\_HOME=$(/usr/libexec/java\_home)

export PATH=$JAVA\_HOME/bin:$PATH

bash

Если which javac выдает путь, отличный от того, что возвращает /usr/libexec/java\_home, используйте этот путь напрямую для JAVA\_HOME. Например:

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home

export PATH=$JAVA\_HOME/bin:$PATH

bash

На Linux может потребоваться указать путь к JDK, например:

export JAVA\_HOME=/usr/lib/jvm/java-21-openjdk-amd64

export PATH=$JAVA\_HOME/bin:$PATH

bash

Сохранить файл и перезагрузить конфигурацию: Сохраните файл и перезагрузите конфигурацию терминала, выполнив команду source ~/.bash\_profile или source ~/.zshrc.

Проверить установку: Откройте новый терминал и введите java -version. Должна отобразиться информация о установленной версии Java. Также попробуйте javac -version.

**4. Структура проекта Android:**

Android проекты обычно создаются и управляются с помощью Android Studio, официальной IDE для разработки Android приложений. Структура проекта выглядит следующим образом:

MyApplication/

├── .gradle/ # Каталог с файлами Gradle

├── .idea/ # Каталог с файлами IntelliJ IDEA (Android Studio)

├── app/ # Каталог с исходным кодом приложения

│ ├── build.gradle # Файл Gradle для модуля app

│ ├── libs/ # Каталог для локальных библиотек

│ └── src/ # Каталог с исходным кодом, ресурсами и манифестом

│ ├── androidTest/ # Каталог для instrumentation tests

│ ├── main/ # Основной исходный код

│ │ ├── java/ # Каталог с исходным кодом Java/Kotlin

│ │ │ └── com/example/myapplication/

│ │ │ ├── MainActivity.java # Основной класс Activity

│ │ │ └── ...

│ │ ├── res/ # Каталог с ресурсами (layouts, drawables, strings, etc.)

│ │ │ ├── drawable/ # Изображения

│ │ │ ├── layout/ # Макеты интерфейса пользователя

│ │ │ │ └── activity\_main.xml # Макет для MainActivity

│ │ │ ├── mipmap-hdpi/ # Иконки приложения для разных разрешений экрана

│ │ │ ├── values/ # Значения ресурсов (strings, colors, dimensions)

│ │ │ │ ├── colors.xml

│ │ │ │ ├── strings.xml

│ │ │ │ └── styles.xml

│ │ │ └── ...

│ │ └── AndroidManifest.xml # Файл манифеста приложения

│ └── test/ # Каталог для unit tests

├── build.gradle # Файл Gradle для всего проекта

├── gradle.properties # Файл настроек Gradle

├── gradlew # Скрипт Gradle для Linux/macOS

├── gradlew.bat # Скрипт Gradle для Windows

├── settings.gradle # Файл настроек проекта Gradle

└── ...

app/src/main/java: Содержит исходный код Java/Kotlin.

app/src/main/res: Содержит ресурсы приложения (макеты, изображения, строки, цвета и т.д.).

AndroidManifest.xml: Файл манифеста, содержащий метаданные приложения (название, иконки, разрешения, компоненты и т.д.).

build.gradle: Файлы конфигурации сборки Gradle, определяющие зависимости, версии SDK и другие параметры сборки.

**5. Запуск приложения:**

Используя Android Studio:

Подключите Android устройство или запустите эмулятор.

Нажмите кнопку “Run” (зеленый треугольник) в Android Studio.

Выберите устройство или эмулятор для запуска приложения.

Из командной строки:

Соберите APK файл: Откройте терминал и перейдите в директорию проекта Android. Выполните команду ./gradlew assembleDebug (для debug сборки) или ./gradlew assembleRelease (для release сборки). APK файл будет создан в директории app/build/outputs/apk.

Установите APK файл на устройство:

Используя adb (Android Debug Bridge): Убедитесь, что adb установлен и настроен. Выполните команду adb install app/build/outputs/apk/debug/app-debug.apk.

Вручную: Скопируйте APK файл на устройство и установите его, используя файловый менеджер. Необходимо разрешить установку приложений из неизвестных источников в настройках безопасности устройства.

\_\_\_\_\_

JDK (Java Development Kit) - это набор инструментов, необходимых для разработки Java-приложений, включая мобильные приложения на Android.

Android Studio - это интегрированная среда разработки (IDE), специально созданная для разработки приложений под Android. Она предоставляет удобные инструменты для создания, отладки и тестирования мобильных приложений Android.

**Установка JDK:**

Перейдите на официальный сайт Oracle (https://www.oracle.com/java/technologies/javase-jdk11-downloads.html) и загрузите последнюю версию JDK для вашей операционной системы.

Запустите загруженный инсталлятор и следуйте инструкциям по установке.

**Установка Android Studio:**

Перейдите на официальный сайт Android Studio (https://developer.android.com/studio) и загрузите последнюю версию Android Studio для вашей операционной системы.

Запустите загруженный инсталлятор и следуйте инструкциям по установке.

**Компоненты Android Studio:**

Android SDK (Software Development Kit) - коллекция инструментов и библиотек для разработки приложений под Android.

Android Virtual Device (AVD) - эмулятор Android-устройств для тестирования приложений.

Gradle - система сборки и управления зависимостями проекта.

Android Debugger - инструмент для отладки приложений под Android.

Android Profiler - инструмент для анализа производительности и использования ресурсов приложения.\

**Компоненты JDK:**

JDK включает следующие компоненты: среда выполнения Java, компилятор Java и API Java.

Настройка Android Studio:

Запустите Android Studio и следуйте инструкциям мастера установки.

Выберите путь для установки Android SDK и установите необходимые платформы Android.

Создайте или войдите в учетную запись разработчика Google для доступа к дополнительным функциям и сервисам.

**Настройка JDK:**

1. Откройте IntelliJ IDEA.

2. Перейдите в раздел «Файл» > «Структура проекта» > «SDKs».

3. Нажмите на знак «+» для добавления нового JDK.

4. Перейдите в каталог установки JDK, который вы установили ранее, и выберите его. IntelliJ загрузит JDK.

5. Убедитесь, что JDK выбран для вашего проекта: перейдите в раздел «Файл» > «Структура проекта» > «Проект».

6. В разделе «Project SDK» выберите JDK, который вы только что добавили.

7. Примените изменения и закройте окно.

**Структура проекта Android Studio:**

app/src/main/java - содержит исходный код приложения.

app/src/main/res - содержит ресурсы приложения, такие как макеты, изображения, строки, стили и т. д.

app/src/main/AndroidManifest.xml - файл манифеста приложения, содержащий информацию о приложении, его компонентах и разрешениях.

app/build.gradle - файл конфигурации проекта, в котором определяются зависимости, настройки версий и другие параметры.

**Структура проекта JDK**

1. Разделение исходного кода и тестового кода. Это позволяет лучше управлять зависимостями и обеспечивает чёткое различие между логикой приложения и кодом, используемым для проверки его корректности.

2. Следование стандартам именования пакетов. Это улучшает читаемость кода и облегчает навигацию в базе кода.

3. Использование Maven или Gradle для управления зависимостями. Это позволяет легко добавлять или обновлять зависимости, отслеживать транзитивные зависимости и управлять версиями конфликтов.

4. Применение принципов модульного проектирования. Это способствует улучшению ремонтопригодности и повышению повторного использования кода.

5. Документация кода. Это помогает понять проект.

**Запуск приложения Android studio:**

Подключите Android-устройство или запустите эмулятор Android Virtual Device (AVD).

В Android Studio откройте проект и выберите нужное устройство из списка доступных устройств.

Нажмите на кнопку "Запустить" (зеленый треугольник) или используйте комбинацию клавиш Shift + F10 для запуска приложения.

Запуск приложения JDK:

1. В стартовом окне выберите пункт «Create New Project».

2. В следующем окне выберите JDK, который будет использован в проекте. Нажмите кнопку «New...» и укажите путь к установленному ранее JDK.

3. В следующем окне выберите пункт «Create project from template» и выберите «Command Line App».

4. Укажите название проекта и Base Package проекта.

5. После завершения мастера нового проекта IDEA сгенерирует новый пустой проект.

6. Запустите проект. Для этого можно выбрать пункт «Run» — «Run „Main“», нажать комбинацию клавиш Shift+F10 или на зелёный треугольник справа в верхней части редактора.

\_\_\_\_\_

**Создать мобильное приложение «Калькулятор» с основными арифметическими операциями (+, - , \*, / ) над действительными числами. Операция выбирается из списка. Числа вводятся в двух полях, вычисления производятся при нажатии на кнопку «=».**