**Введение**

Управление личными финансами зачастую понимается людьми как экономия денежных средств. В свою очередь управление личными финансами – является распределением денег в зависимости от целей человека. Все большую актуальность в наше время имеет неправильное управление личными финансами. Люди не задумываются или же вовсе не контролируют свою расходы и доходы. Причиной этого, отчасти, являются рекламы, акции в магазинах, общественное мнение. Под влиянием этих факторов человек зачастую совершает ненужные траты, не следит за своими расходами и тратит деньги не обдуманно.

Управление личными финансами помогает выбрать план согласно вашим целям, учит с умом тратить деньги и организовывать семейный бюджет.

Управление личными финансами служит следующим целям:

- понять какие денежные средства используется неэффективно;

- выявить причины недостатка денег и найти способ их решения;

- проанализировать структуру расходов с целью повышения их эффективности.

Целью дипломной работы является создание мобильного приложения для управления личными финансами, которое будет простым в использовании и обеспечивает хранение в электронном виде данных о доходах и расходах.

Декомпозиция цели может дать список предполагаемых задач исследования:

1. **Программные решения, используемые для разработки мобильного приложения для управления финансами.**
   1. **Среда разработки Android Studio.**

Android Studio (рисунок 1.1) — является интегрированной средой разработки, с помощью которой пользователю доступны инструменты для создания приложений на платформе Android OS. Android Studio основана на базе IntelliJ IDEA от компании JetBrains. Android Studio доступна для Windows, macOS и GNU/Linux.

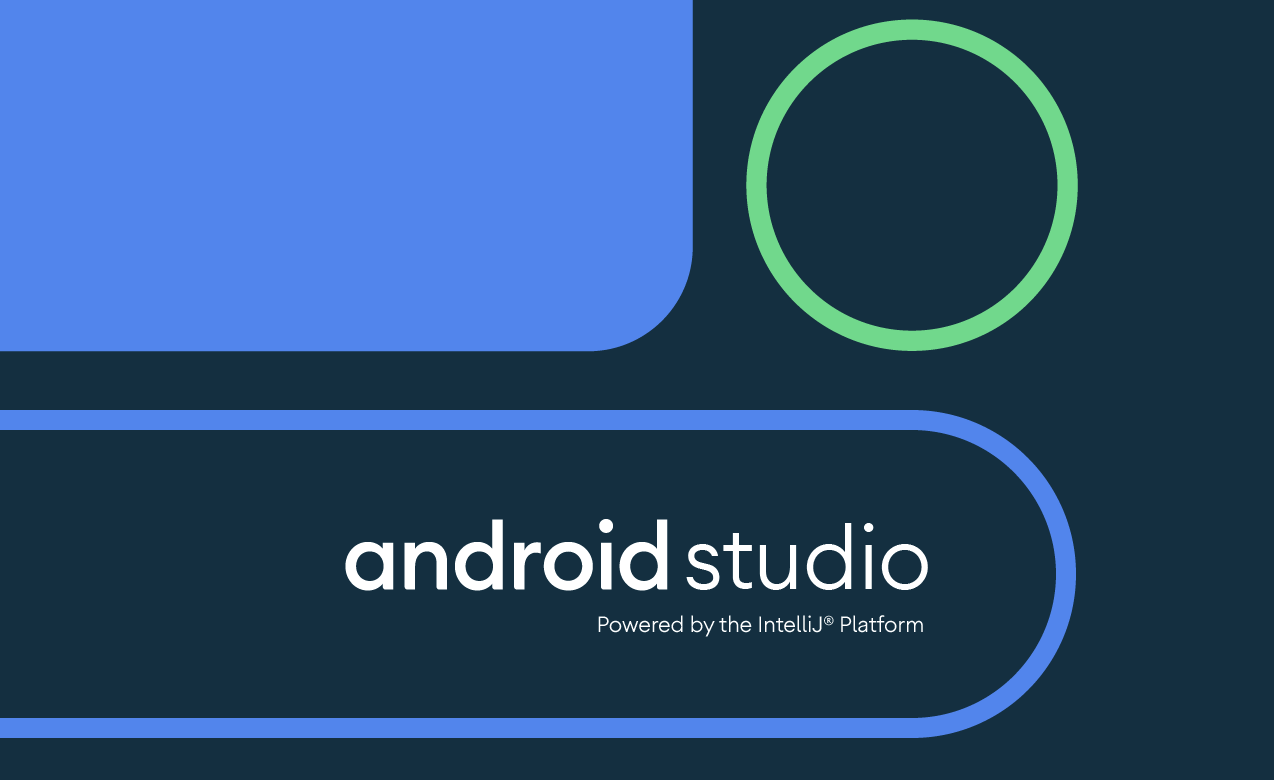


Рисунок 1.1 Android studio

Среда разработки Android Studio является бесплатной, то есть ее можно спокойно загрузить и сразу начать работать. Среда IDE не адаптирована под русский язык, поэтому интерфейс на английском. В Android Studio присутствуют макеты для создания пользовательского интерфейса, с которых можно начать разработку приложения. Также содержатся инструменты разработки решений для смартфонов и планшетов, а также поддерживается разработка приложений для Android Wear и Android TV.

В среде разработки Android Studio можно работать как одному разработчику, так и группе разработчиков с GIT или другими подобными системами управления версиями. Решения для Android разрабатываются в Android Studio с использованием Java, Kotlin или C++. В процесс разработки заложен концепт непрерывной интеграции, который позволяет сразу обнаружить имеющиеся проблемы.

Определять состояние файла с прикладными программами позволяют средства оценки производительности. С помощью визуализации графики можно проверить соответствие приложения ориентиру Google. А также с помощью инструмента для визуализации памяти разработчик может определять, когда его приложение использует много оперативной памяти и когда произойдет "сборка мусора". Инструменты для анализа батареи дают нам узнать, какая нагрузка приходится устройство.

Локализация приложений существенно проще с функцией SDK, которая также входит в перечень достоинств Android Studio.

Каждый проект в Android Studio содержит один или несколько модулей с файлами исходного кода и файлами ресурсов. Типы модулей включают:

1. Модули приложений Android
2. Модули библиотеки
3. Модули Google App Engine

По умолчанию Android Studio отображает файлы вашего проекта в представлении проекта Android, как показано на рисунке 1.2. Это представление организовано по модулям для обеспечения быстрого доступа к ключевым исходным файлам вашего проекта.

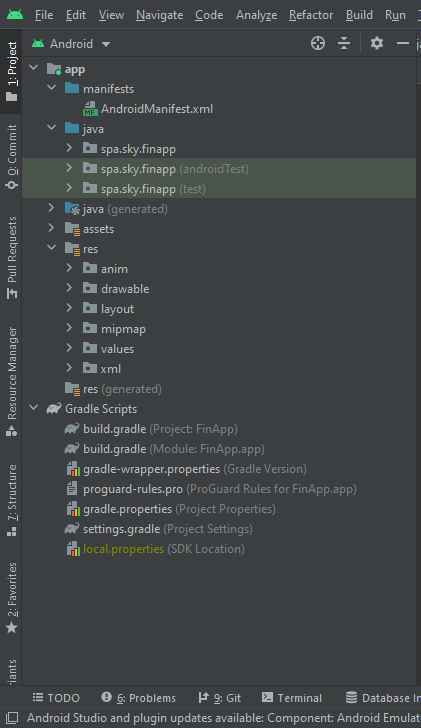


Рисунок 1.2 Файлы проекта в Android Studio

Все файлы сборки видны на верхнем уровне в Gradle Scripts, и каждый модуль приложения содержит следующие папки:

* + 1. manifests: содержит AndroidManifest.xml файл.
    2. java: содержит файлы с исходным кодом на языке Java, включая тестовый код JUnit.
    3. res: содержит все ресурсы, не связанные с кодом, такие как макеты XML, строки пользовательского интерфейса и растровые изображения.

Достоинства Android Studio:

1. редактор кода, с которым довольно легко работать;
2. тестировать корректность работы проложений, их производительности на той или иной системе, можно непосредственно в эмуляторе;
3. рефакторинг уже готового кода;
4. достаточно большая библиотека с готовыми шаблонами и компонентами для разработки ПО;
5. разработка приложения для Android N – самой последней версии операционной системы;
6. предварительная проверка уже созданного приложения на предмет ошибок в нем;
7. ­­для неопытных или начинающих разработчиков специально создано руководство по использованию Android Studio, которое размещено на официальном сайте утилиты.
   1. **Программирование в Android Studio на Java.**

Почему именно Java? Java - язык, который широко используется для разработки мобильных приложений. Существуют и другие языки, которые можно использовать для разработки приложений Android, например C и C ++, с использованием Native Development Kit (NDK), но для разработки приложений рекомендуется использовать Java. В доступной среде IDE, например Android Studio, Java является языком по умолчанию для кодирования, тогда как C и C ++ находятся в опциях. Он также поддерживает собственный код, поэтому C и C ++ применимы к коду для мобильного приложения. SDK (Software Development Kit) используется для разработки под Android, который имеет кучу библиотек вместо JVM.

Java и XML (eXtensible Markup Language) используются при разработке мобильных приложений. Java используется в разработке под Android для написания внутренней логики или бизнес-логики. Это язык высокого уровня. XML используется для разработки пользовательского интерфейса. Java объектно-ориентированный язык, но не чисто объектно-ориентированный из-за наличия встроенных типов данных в нем. Он называется объектно-ориентированным, потому что поддерживает все функции объектно-ориентированного языка.

Java играет важную роль в разработке приложений для Android, поскольку бизнес-логика написана на Java. Знание Java заранее является плюсом для разработки приложений для Android. Обладая глубокими знаниями Java, вы можете добавлять в приложение новые функции.

Java - объектно-ориентированный язык, что является серьезной причиной использования этого языка для разработки приложений. Он безопасен благодаря использованию классов и объектов и не поддерживает указатель и любое другое ключевое слово для управления памятью вручную, что снижает вероятность нежелательного вмешательства в приложение. Для управления памятью он использует сборщик мусора, который удаляет объекты, не используемые в течение длительного времени.

* 1. **Классы и библиотеки IDE Android studio.**

Fragment manager  является классом, который отвечает за выполнение действий с фрагментами вашего приложения, например, добавление, удаление или замена их, а также добавление их в задний стек.

Каждая FragmentActivity и ее подклассы, такие как AppCompatActivity, имеют доступ к FragmentManager через метод getSupportFragmentManager (). Фрагменты также могут содержать один или несколько дочерних фрагментов. Внутри фрагмента вы можете получить ссылку на FragmentManager, который управляет дочерними элементами фрагмента через getChildFragmentManager (). Если вам нужен доступ к его хосту FragmentManager, вы можете использовать getParentFragmentManager ().

В приложении для управления личными финансами Fragment manager используется для смены одного фрагмента на другой. На рисунке 1.3 представлен фрагмент кода замены одного фрагмента на другой.

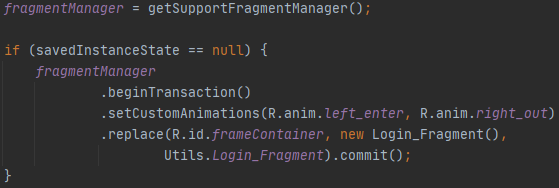


Рисунок 1.3 использование Fragment manager для смены фрагмента

Библиотека MPAndroidChart – библиотека диаграмм, которую можно перетаскивать и масштабировать, включая графики, гистограммы и круговые диаграммы.

Основные функции:

1. 8 различных типов диаграмм
2. масштабирование по двум осям
3. перетаскивание/ панорамирование
4. комбинированные графики
5. двойные (отдельные) оси
6. настраиваемые оси (ось X и ось Y)
7. выделение значений
8. можно сохранять диаграммы на SD-карту
9. предустановленные шаблоны цвета
10. анимации
11. линии предела
12. полностью настраивается (краска, шрифт, легенда, цвет, фон, жест, пунктирная линия, ...)
13. поддержка Grandle

Добавление библиотеки MPAndroidChart происходит путем настройки Grandle (рисунок 1.4).

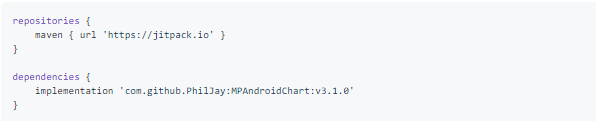


Рисунок 1.4 настройки Grandle