|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №1**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-21-23 | Муравьев А. О. |
| **Проверил:**  Старший преподаватель кафедры МОСИТ | Шешуков Л.С. |

Москва 2024 г.

**СОДЕРЖАНИЕ**

[1. Цель работы 3](#_Toc190960888)

[2. Теоретическое введение 3](#_Toc190960889)

[3. Код программы 4](#_Toc190960890)

[4. Результаты работы программы 12](#_Toc190960891)

[5. Выводы по работе 14](#_Toc190960892)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc190960893)

# Цель работы

Цель данной работы — ознакомиться с основами проектирования и разработки мобильных приложений под Android, а также научиться использовать файлы разметки (XML) и базовые компоненты пользовательского интерфейса: текстовые поля, кнопки, поля ввода и изображения.

В ходе выполнения задания будет создан проект с несколькими файлами разметки и различными элементами интерфейса, что позволит отработать навыки применения стандартных Android-компонентов и их расположения в макете.

# Теоретическое введение

Мобильные приложения представляют собой программы, разработанные для использования на смартфонах или планшетах под управлением таких операционных систем, как Android и iOS. Для создания Android-приложений обычно применяют языки Kotlin или Java, а пользовательский интерфейс определяется при помощи XML-файлов разметки.

В Android файлы разметки располагаются в папке res/layout и содержат описание структуры пользовательского интерфейса, отделяя визуальную часть приложения от логической. Основными контейнерами для расположения элементов на экране выступают LinearLayout, RelativeLayout, ConstraintLayout и FrameLayout, каждый из которых имеет собственные особенности компоновки.

В стандартный набор UI-компонентов Android входят:

- TextView – отображает текст,

- EditText – предоставляет поле для ввода текста пользователем,

- Button – кнопка, реагирующая на нажатие,

- ImageView – используется для вывода изображений.

Каждый из перечисленных элементов имеет ряд атрибутов (например, layout\_width, layout\_height, text, background, src), которые позволяют изменять их внешний вид и настраивать поведение.

При работе с пользовательским интерфейсом в Android рекомендуются следующие принципы:

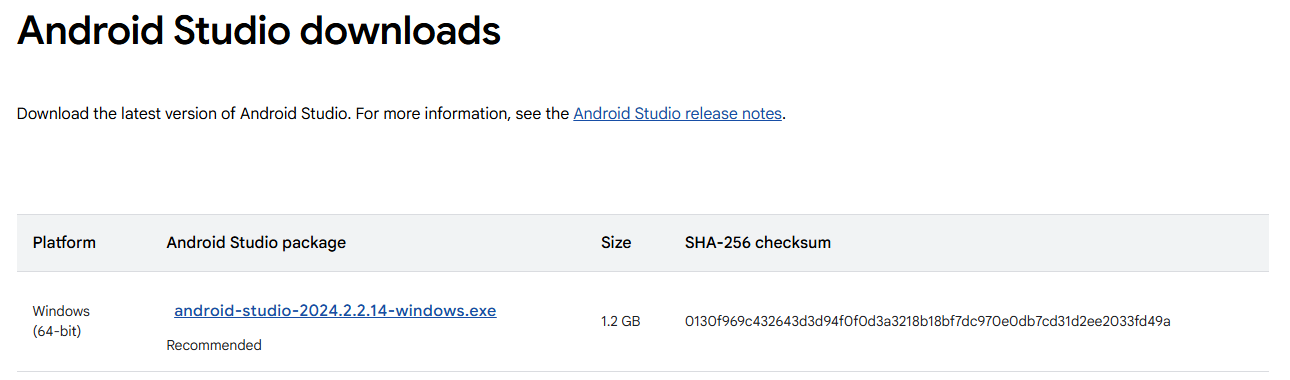
- использовать ConstraintLayout для более гибкого расположения компонентов,

- применять стили и темы для единообразного оформления приложения,

- реализовывать обработчики событий, обеспечивая взаимодействие пользователя с интерфейсом.

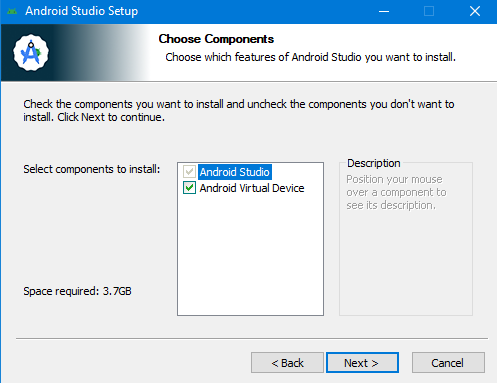
# Подготовка среды разработки

В качестве среды разработки будет использоваться Android Studio версии 2024.2.2.14 для Windows. Соответствующую версию можно скачать на официальном сайте Android Developers (рис. 1).

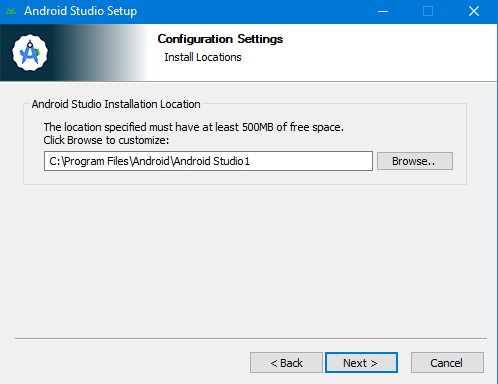
  
Рисунок 1 – Ссылка для скачивания установщика Android Studio

После скачивания установочного файла его необходимо запустить и настроить процесс установки.

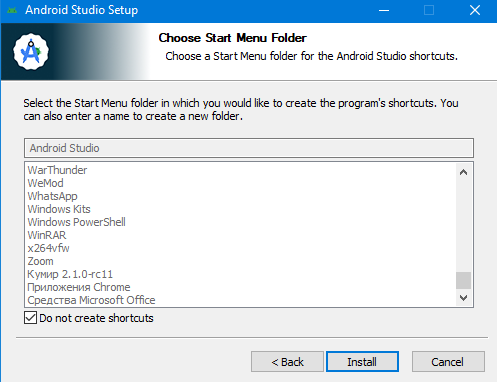
Сперва указываем компоненты к установке (рис. 2).

  
Рисунок 2 – Установка Android Studio, настройка установки компонентов

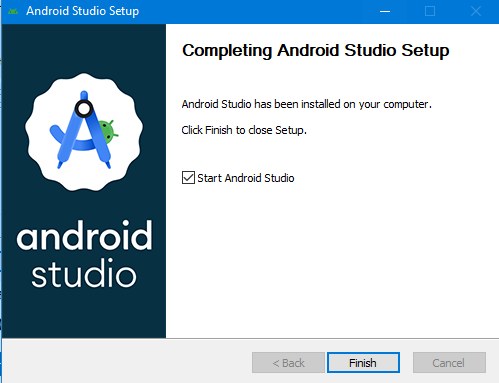
Далее указываем место установки Android Studio (рис. 3).

  
Рисунок 3 – Установка Android Studio, настройка места установки

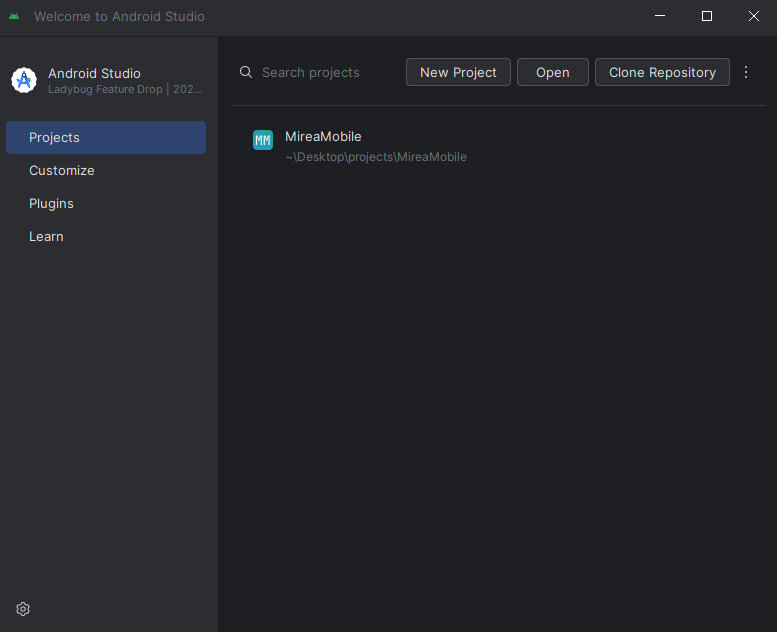
Отказываемся от создания папки Android Studio в меню Пуск и устанавливаем среду разработки (рис. 4).

  
Рисунок 4 – Установка Android Studio, создание папки в меню Пуск

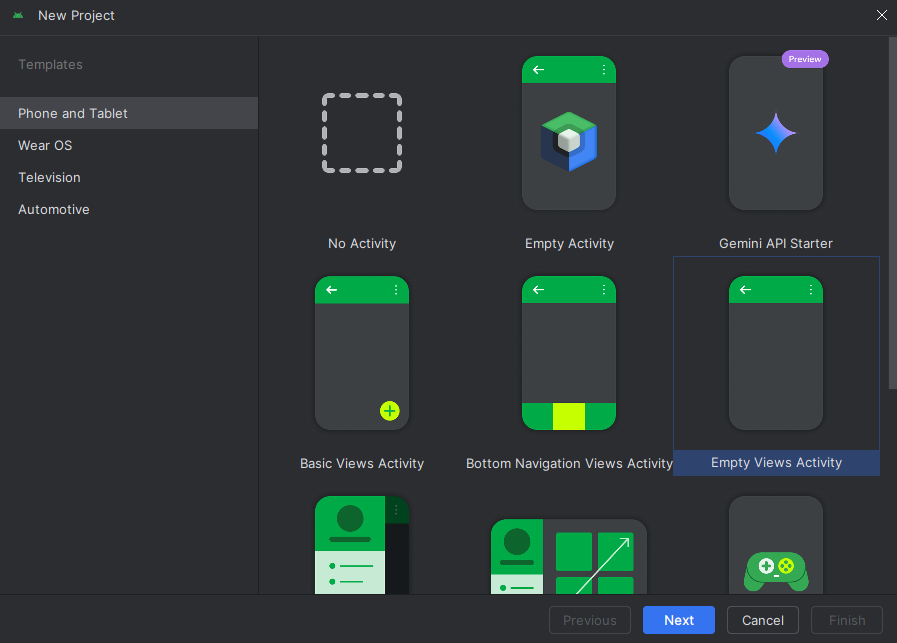
Далее установщик самостоятельно установит Android Studio на компьютер. После установки соглашаемся на запуск Android Studio и завершаем установку (рис. 5).

  
Рисунок 5 – Установка Android Studio, завершение установки

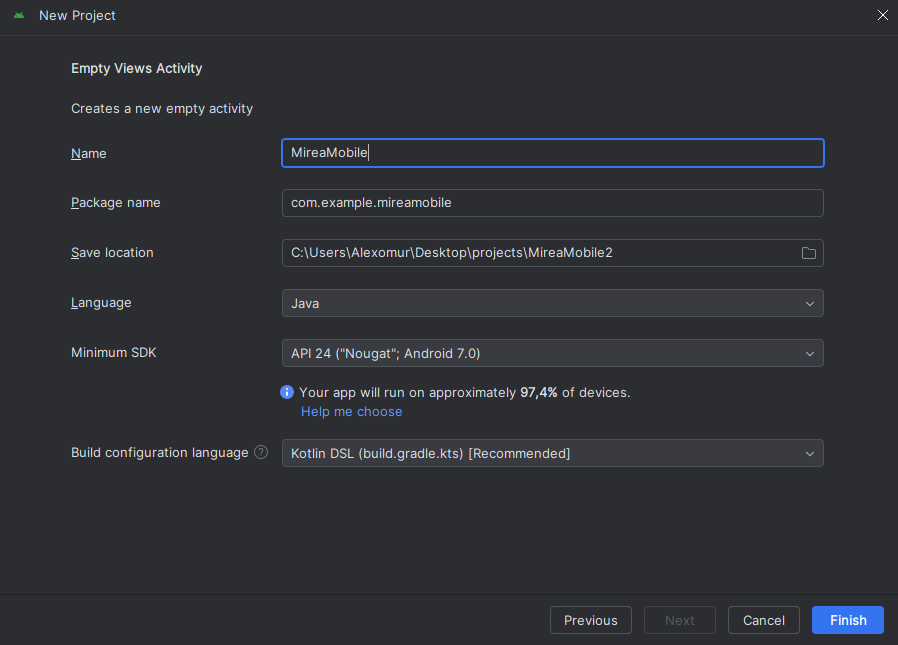
В открывшимся окне выбираем «New Project» (рис. 6).

  
Рисунок 6 – Интерфейс управления проектами в Android Studio

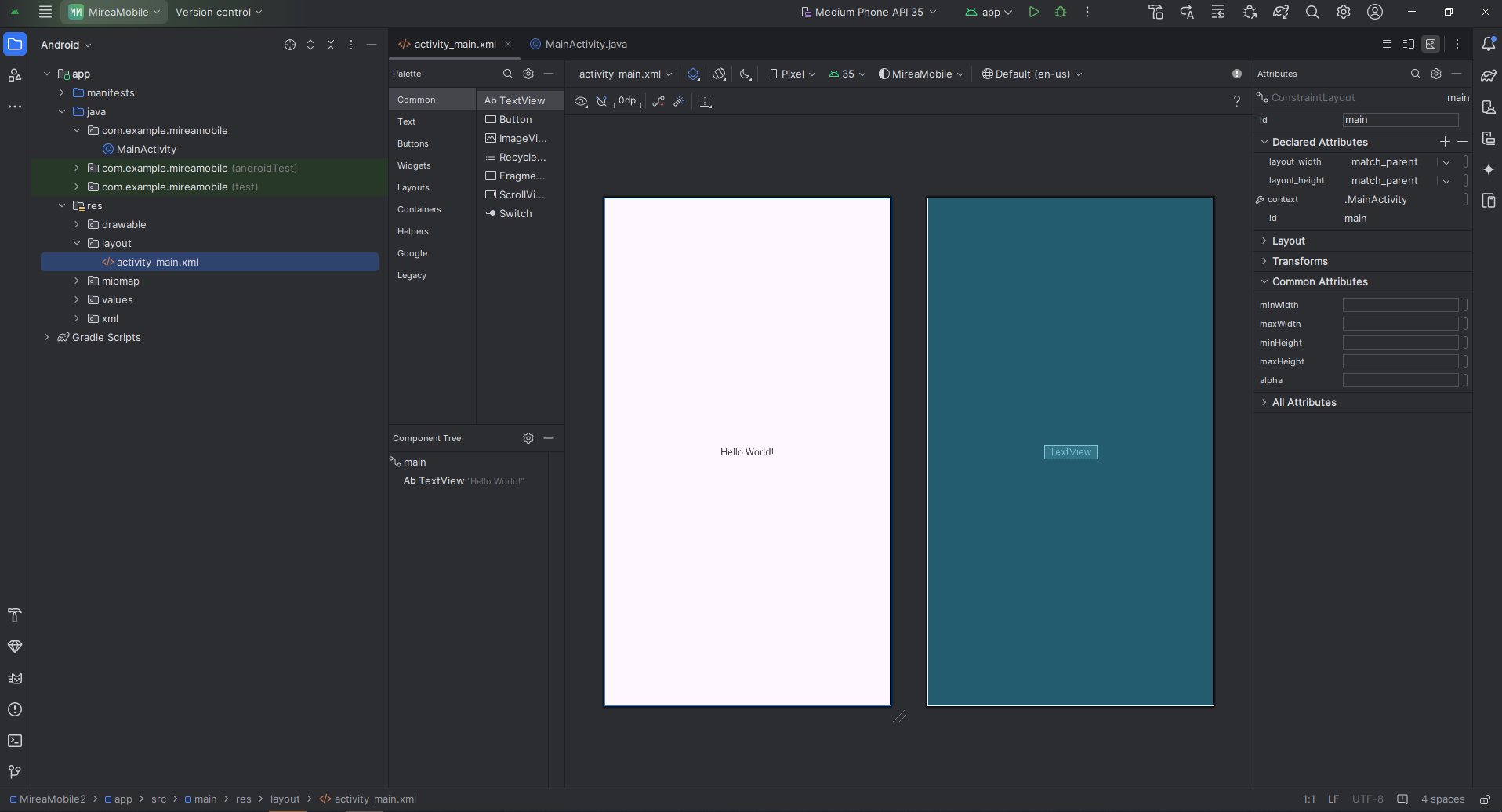
Выбираем шаблон «Empty Views Activity» из предлагаемых шаблонов для телефонов и плашнетов (рис. 7).

  
Рисунок 7 – Настройка будущего проекта, выбор шаблона

Выбираем название, место на диске, язык программирования и другие настройки для создания проекта (рис. 8).

  
Рисунок 8 – Настройка будущего проекта, первичные настройки

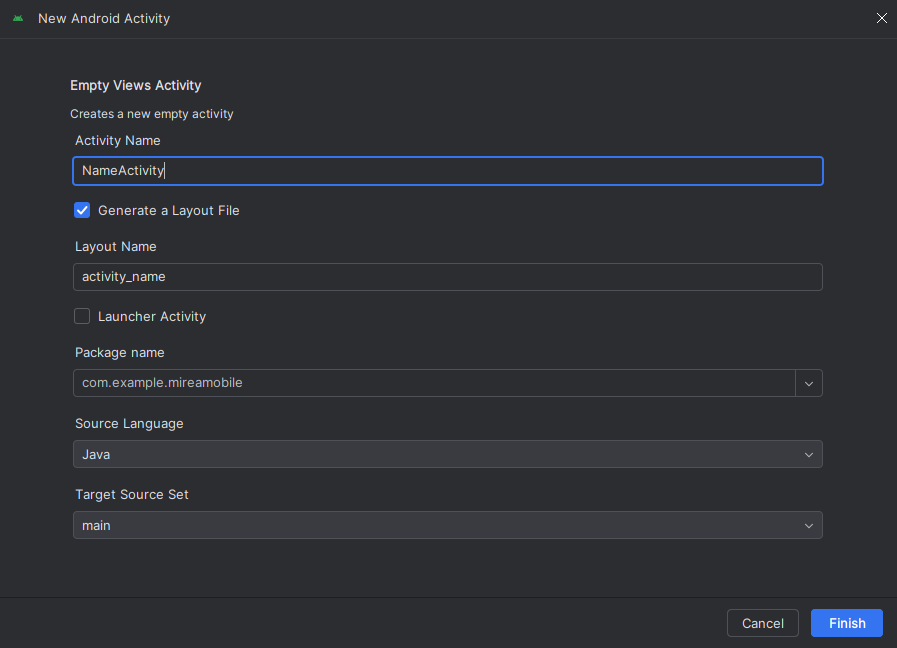
Android Studio создаст пустой проект с одной страницей (рис. 9).

Рисунок 9 – Созданный пустой проект

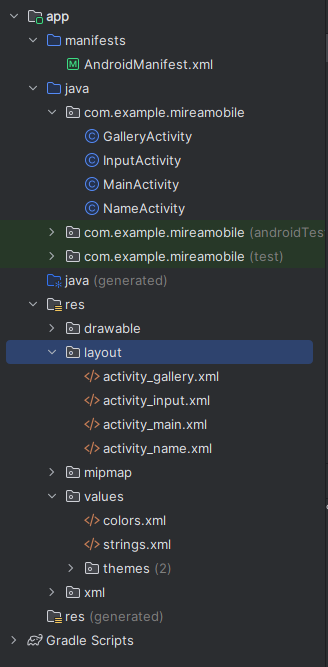
Теперь среда разработки полностью готова к реализации проекта.

# Реализация проекта

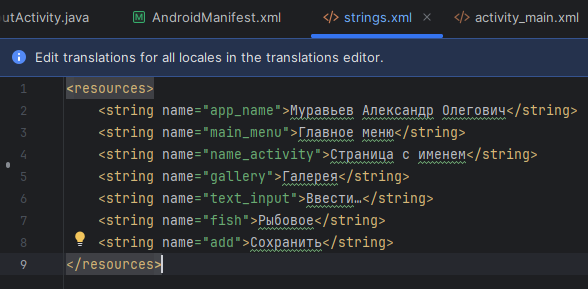
Планируется реализация четырёх страниц: главная страница, страница с именем студента, галерея и страница с вводом. Главная страница создана по умолчанию (MainActivity). Создадим страницу NameActivity с именем студента (рис. 10).

  
Рисунок 10 – Создание страницы NameActivity

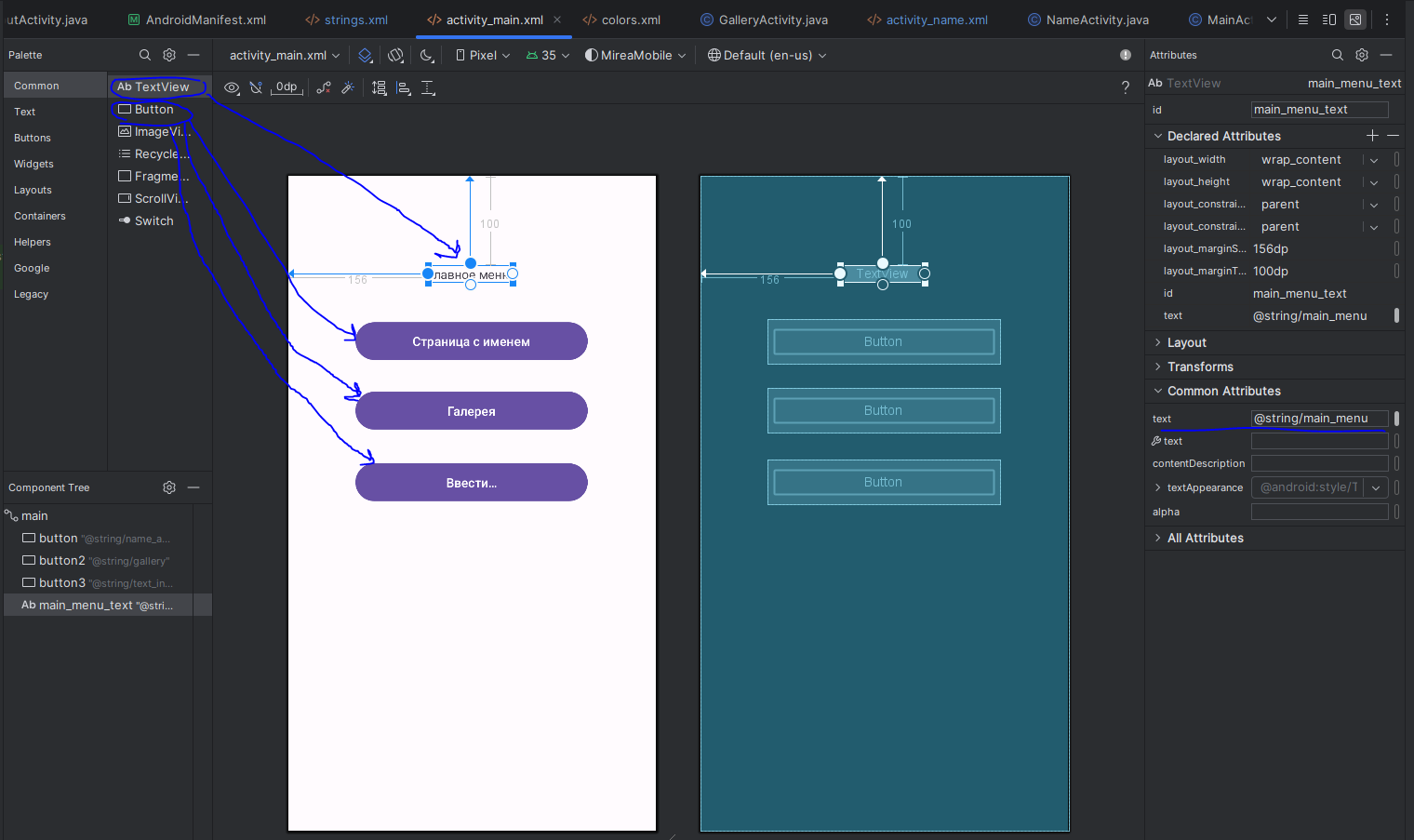
Таким же образом создаём страницы GalleryActivity (галерея) и InputActivity (страница с вводом). Должна выйти архитектура проекта, как на рисунке 11 (за исключением файлов с меткой «generated»).

  
Рисунок 11 – Полученная архитектура проекта

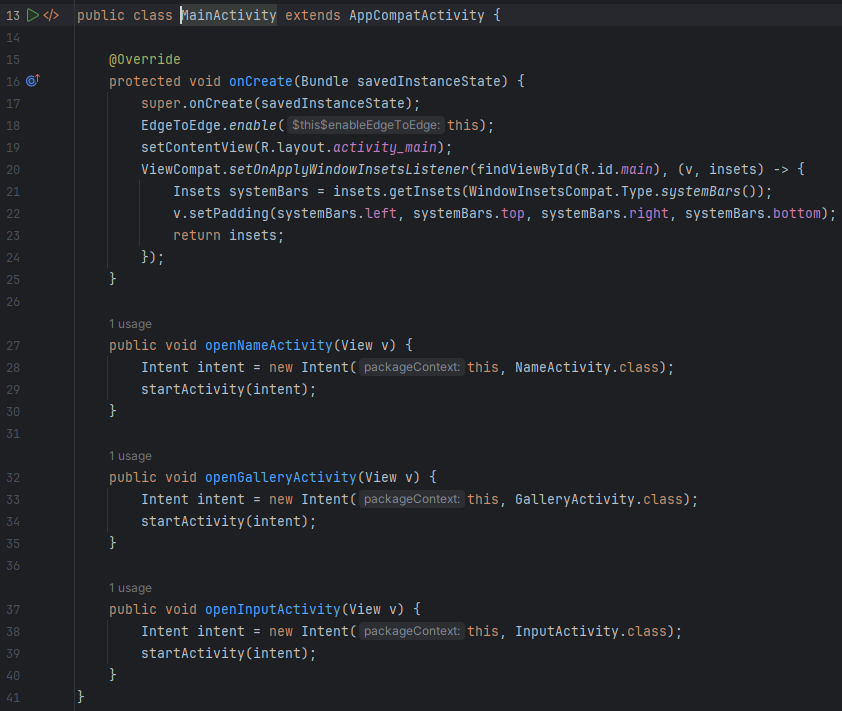
Нужно заготовить строки для проекта через настройку файла strings.xml. Переименовать проект, назвав его именем студента, а также установить несколько тегов, которые понадобятся позже (рис. 12).

  
Рисунок 12 – Содержимое файла strings.xml

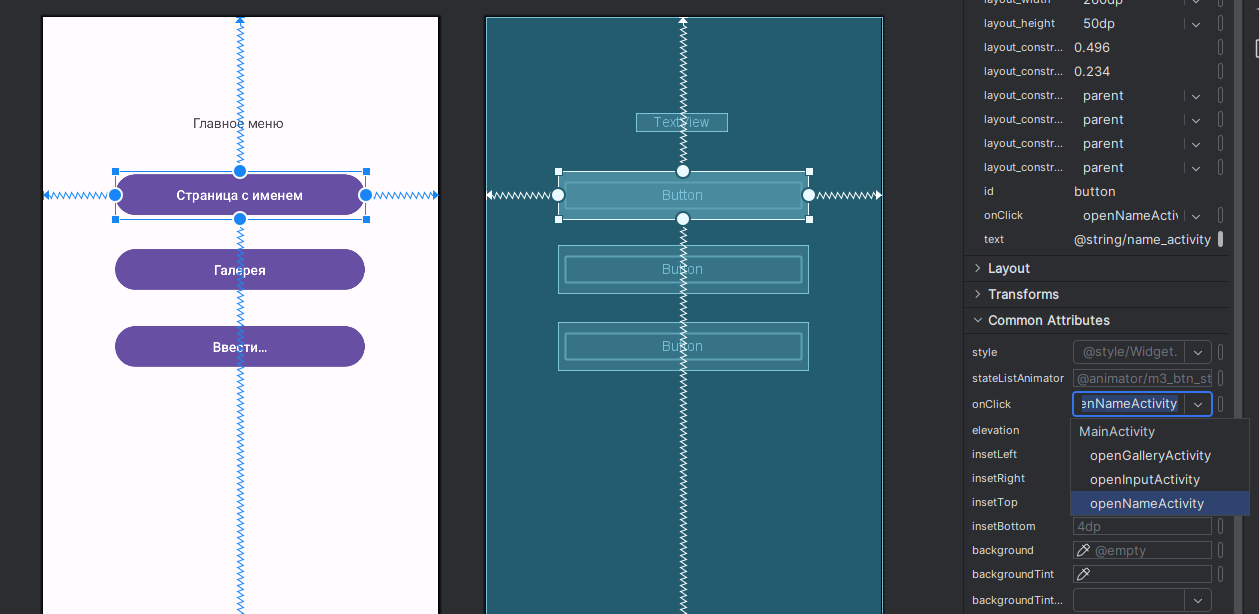
С помощью графического редактора Android Studio добавим в окно activity\_main.xml текст и несколько кнопок навигации. Установим текст новых элементов в поле text с помощью нотации «@string/», указывая названия строк, записанных ранее. На рисунке 13 продемонстрировано новое главное меню.

  
Рисунок 13 – Созданное главное меню, стрелками указаны типы соответствующих элементов, подчеркнуто поле редактирования текста

Для добавления функционала перехода кнопкам нужно модифицировать стандартный класс MainActivity, добавив методы перехода: openNameActivity, openGalleryActivity, openInputActivity (рис. 14).

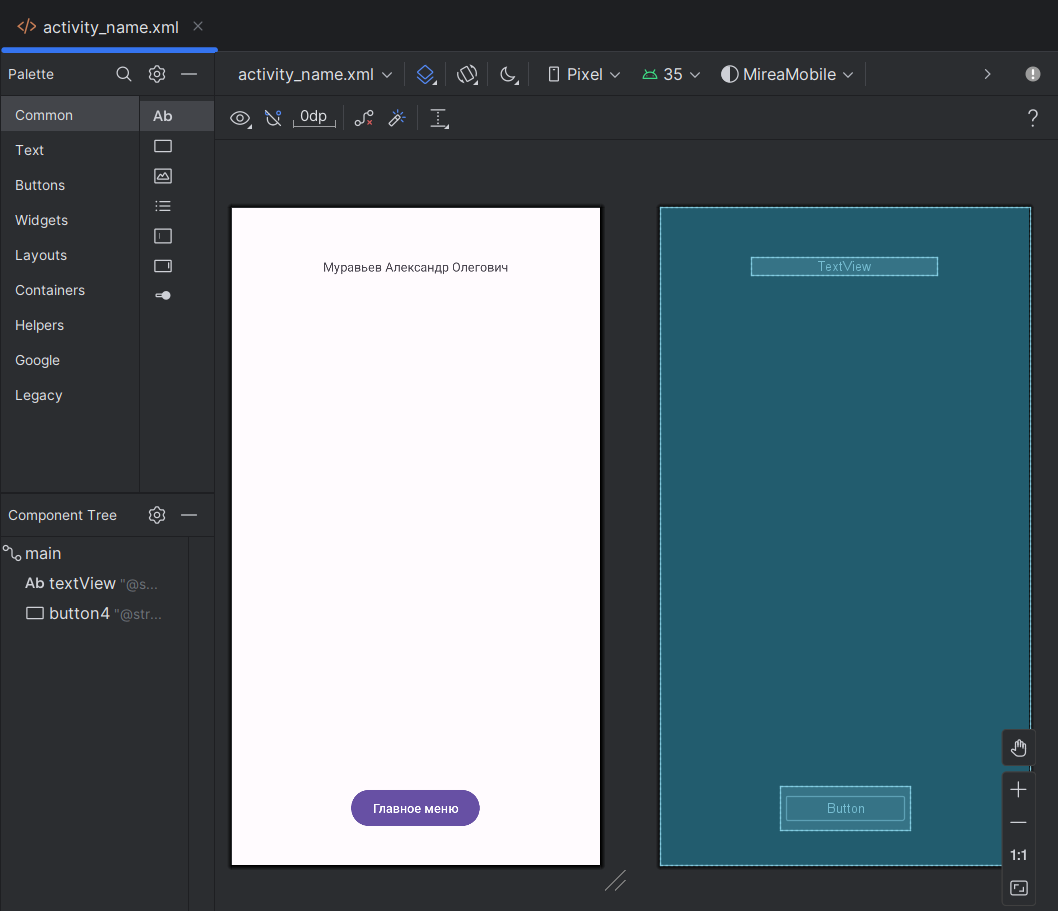
  
Рисунок 14 – Код класса MainActivity

Для присвоения метода к нажатию кнопки нужно указать соответствующий метод в поле onClick (рис. 15).

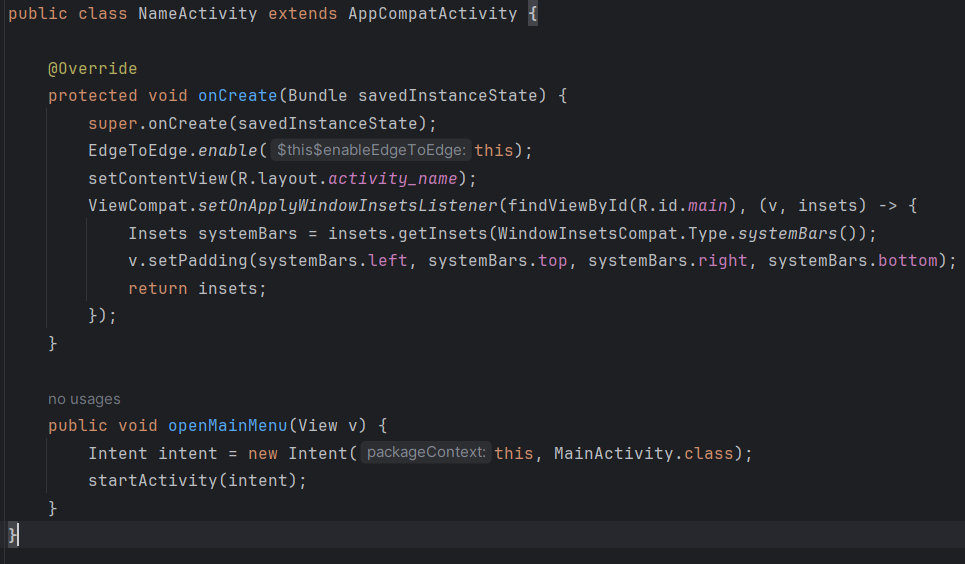
  
Рисунок 15 – Указание метода openNameActivity для вызова при нажатии на кнопку «Страница с именем»

Таким же способом указываем методы для кнопок «Галерея» и «Ввести…».

Для реализации страницы с именем нужно в файле activity\_name.xml добавить необходимые элементы с помощью графического редактора. На этой странице должна быть надпись с ФИО студента, а также кнопка «Главное меню» (рис. 16).

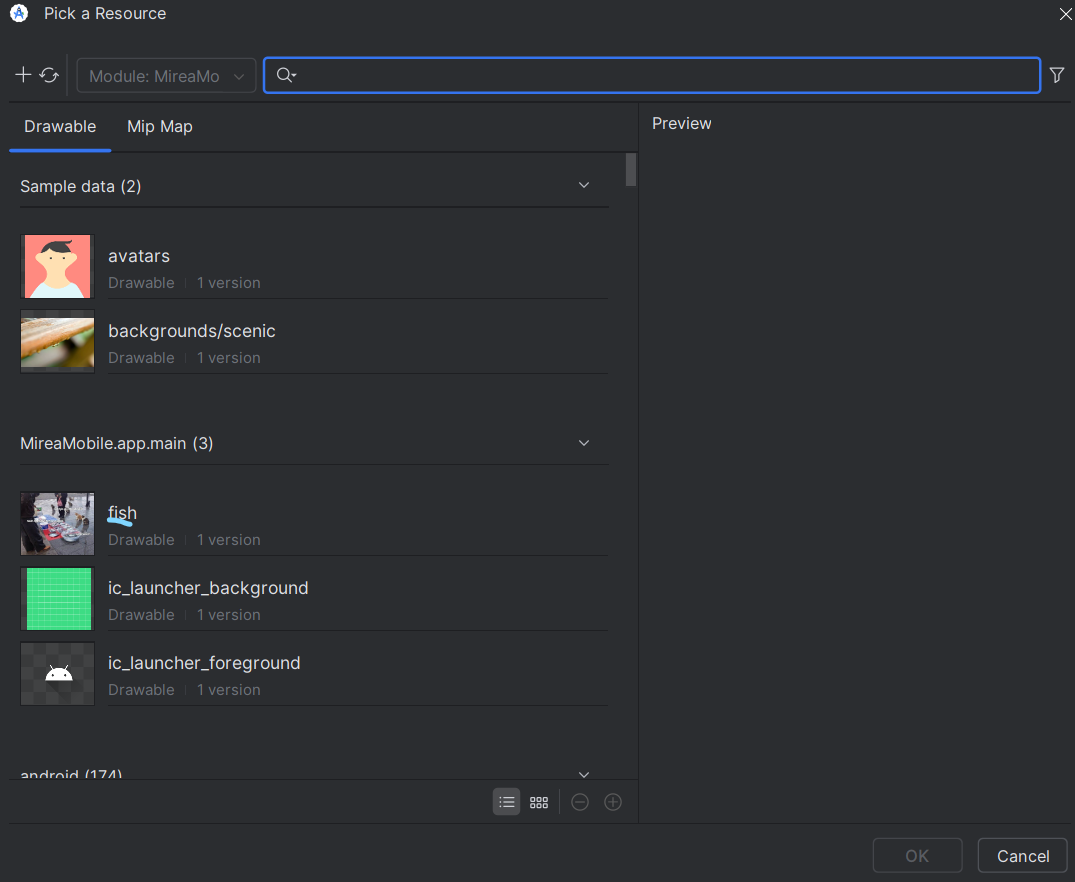
  
Рисунок 16 – Созданная страница activity\_name.xml

Для реализации логики кнопки «Главное меню» нужно добавить в класс NameActivity код, показанный на рисунке 17.

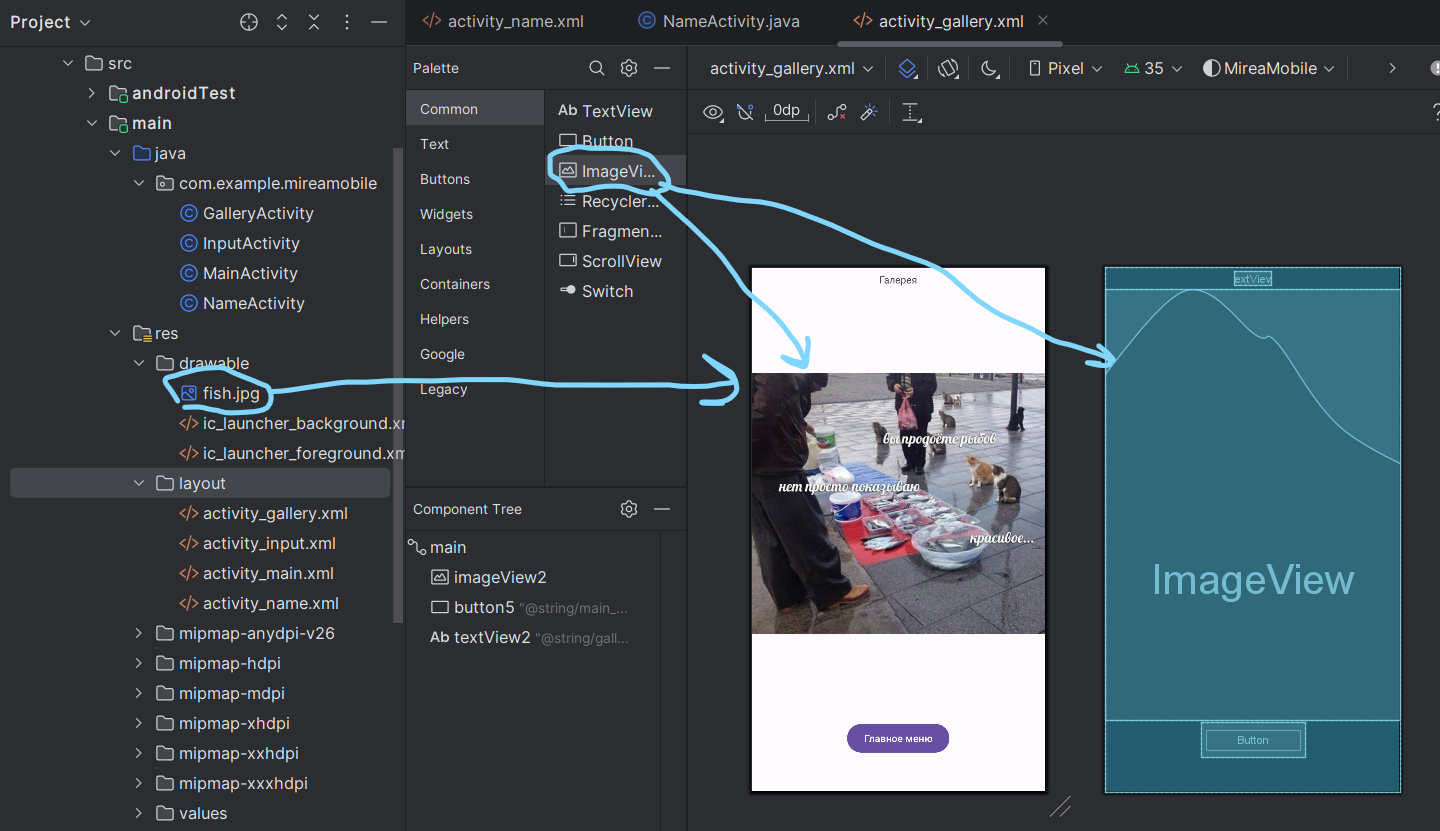
  
Рисунок 17 – Код класса NameActivity

Также нужно привязать кнопку к методу, как это было сделано ранее с кнопками в главном меню.

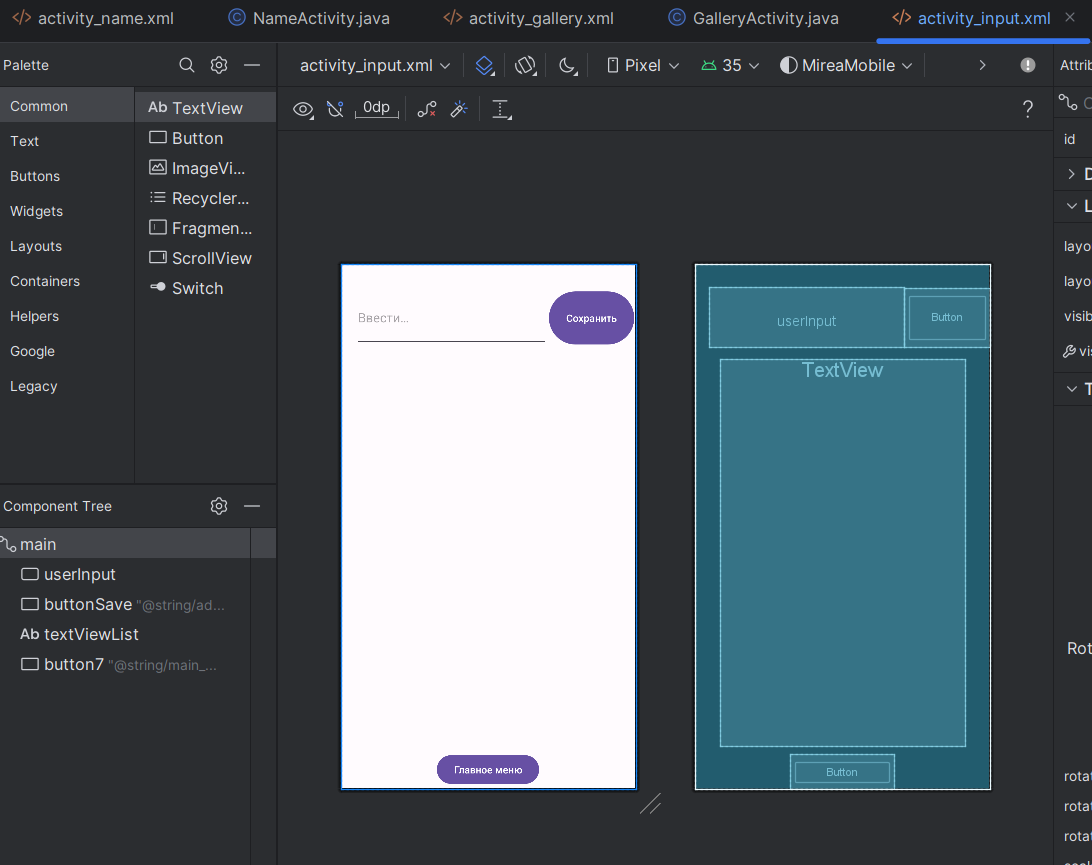
Для реализации галереи картинку для галереи нужно сохранить в папку drawable. На странице нужно создать элемент ImageView и в открывшемся окне выбрать соответствующую картинку (рис. 18).

  
Рисунок 18 – Окно выбора картинки для ImageView

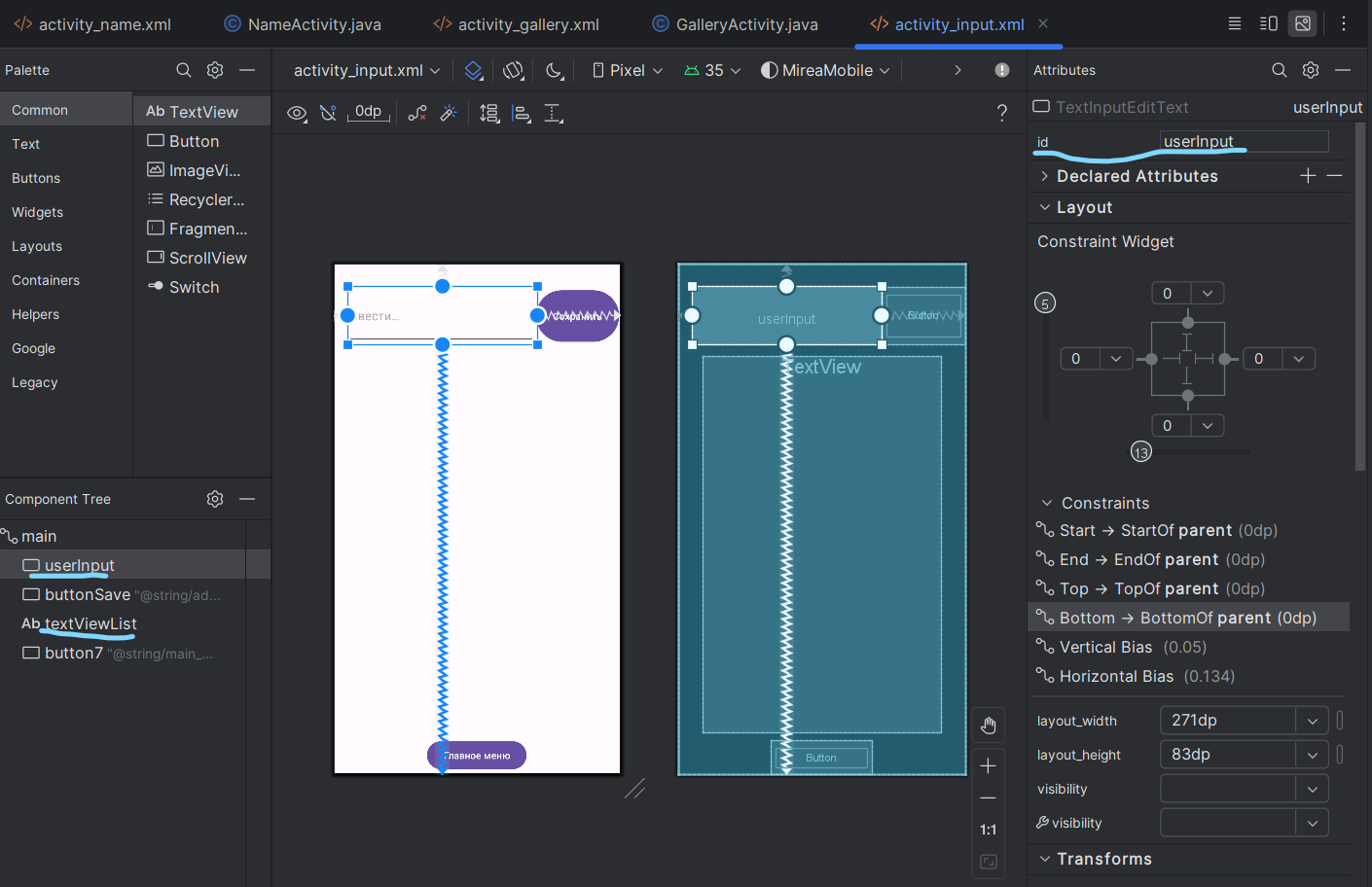
Также нужно добавить остальные элементы страницы – название страницы и кнопка «Главное меню», как в предыдущей странице (рис. 19).

  
Рисунок 19 – Созданная страница activity\_gallery.xml, стрелками указаны элементы, использованные на странице

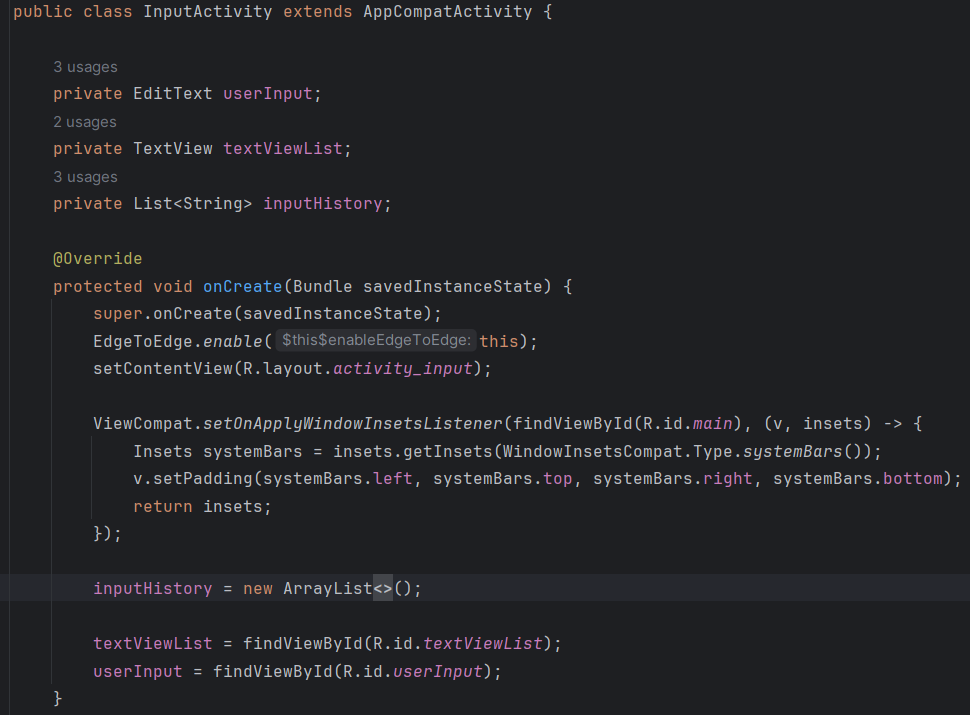
Страница с вводом текста должна включать в себя поле ввода, кнопку сохранения, поле показа текста и кнопку «Главное меню». При вводе текста в поле ввода и нажатии на кнопку сохранения текст должен добавляться в поле показа текста. Для реализации поля ввода текста должен быть использован элемент TextInputEditText. Созданная страница показана на рисунке 20.

  
Рисунок 20 – Созданная страница activity\_input.xml

Для управления полем ввода и вывода нужно дать этим полям id. Для этого нужно установить соответствующие значения в поле id для соответствующих полей (рис. 21).

  
Рисунок 21 – Созданная страница activity\_input.xml, подчёркнуты id полей ввода и вывода

Чтобы введённый текст отслеживался приложением, в классе страницы нужно сохранить историю ввода и информацию о полях ввода и вывода. Информация должна сохраняться при создании страницы (рис. 22).

  
Рисунок 22 – Код класса InputActivity, сохранение данных о кнопке и полях, в методе onCreate происходит иницализация истории inputHistory и поиск элементов с id «textViewList» и «userInput»

А при нажатии на кнопку «Сохранить» приложение должно считать информацию в поле userInput, сохранить ввод в inputHistory и обновить текст в поле вывода textViewList (рис. 23).

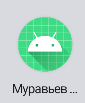
  
Рисунок 23 – Код класса InputActivity, обновление истории и очистка поля ввода, метод кнопки возвращения в главное меню

Для кнопок «Сохранить» и «Главное меню» нужно указать методы save и openMainMenu соответственно, как это было сделано ранее.

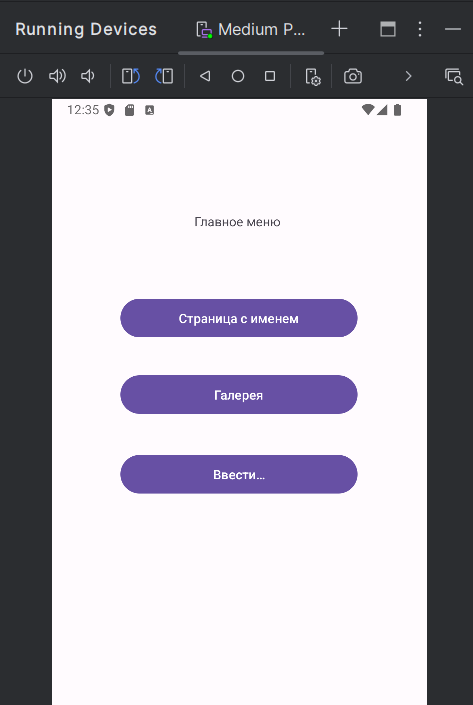
# Результаты работы приложения

С помощью встроенного эмулятора AndroidStudio (использован Medium Phone с версией Android 35) проверим работоспособность приложения.

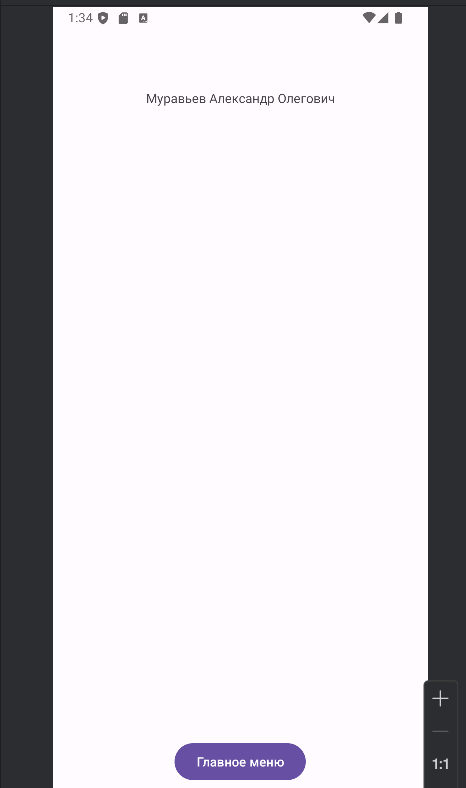
Как видно из рисунка 24, приложение имеет в названии ФИО студента.

  
Рисунок 24 – Иконка и часть названия созданного приложения

На рисунке 25 продемонстрировано главное меню.

  
Рисунок 25 – Главное меню приложения

На рисунке 26 показана страница с ФИО студента.

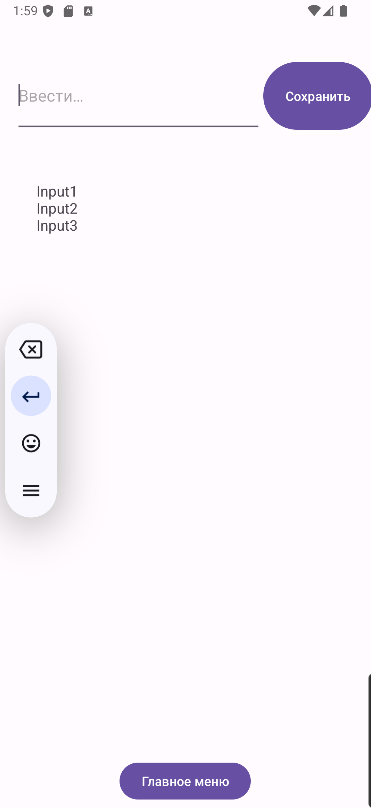
  
Рисунок 26 – Страница с ФИО студента

Кнопка «Главное меню» возвращает на предыдущую страницу.

На рисунке 27 продемонстрирована галерея.

  
Рисунок 27 – Страница с галереей

На странице с пользовательским вводом введём строки «Input1», «Input2», «Input3» для проверки работы приложение и сохраним каждую из них. Тестирование показано на рисунке 28.

  
Рисунок 28 – Страница с вводом, протестирован функционал пользовательского ввода

# Выводы по работе

В ходе выполнения практической работы были получены навыки создания мобильного проекта в среде разработки Android Studio, а также работы с файлами разметки XML. Были освоены основы проектирования пользовательского интерфейса, размещения и настраивания базовых компонентов, таких как текстовые поля (TextView), кнопки (Button), поля ввода (TextInputEditText) и изображения (ImageView).

Кроме того, была изучена работа с различными атрибутами UI-элементов и принципами их компоновки с использованием ConstraintLayout, что позволяет гибко управлять расположением элементов на экране.

Таким образом, в рамках данной работы были закреплены базовые знания по разработке интерфейсов в Android-приложениях.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по Android Studio [Электронный ресурс]. – URL: <https://developer.android.com/studio/intro?hl=ru> (дата обращения: 21.02.2025)
2. Программирование под Android на Java [Электронный ресурс]. – URL: <https://metanit.com/java/android/> (дата обращения: 21.02.2025)
3. Уроки Android Studio с нуля [Электронный ресурс]. – URL: <https://itproger.com/course/java-android> (дата обращения: 21.02.2025)