

Лекционно-практическое занятие
«Использование нескольких Activity в мобильном приложении.
Передача данных между Activity»

Вспомним что такое Activity. **Activity** – экран, который видит пользователь и с которым он взаимодействует при помощи различных элементов (кнопки, текст и т.п.).

Ранее рассматривалось приложение, в котором было всего одно Activity. НО: Приложение не всегда состоит из одного экрана. Приложение может содержать одну или несколько активностей. Активность можно запустить двумя способами:

- Главная активность, определенная в файле AndroidManifest, запускается при открытии приложения.
- Все остальные активности можно настроить таким образом, чтобы они запускались через явные и неявные **намерения (Интенды (англ. яз Intent))**, определенные в файле AndroidManifest.xml.

Намерения – это объект класса и выдающаяся особенность ОС Android.

- Для явных намерений – объект сообщает что ему нужно выполнить какое-то действие с выбранным компонентом приложения (активность, сервис и т.д.).
- Для неявных намерений – намерение просто сообщает что оно должно сделать, не указывая какой именно компонент должен обработать это намерение, это решает пользователь или сама ОС Android.

С точки зрения пользователя – активности, это экраны, которые могут быть запущены в работающем приложении или из другого стороннего приложения. И как только активность запустится, она также помещается в стеке задач, который могут увидеть пользователи, нажав кнопку Назад.

Задача приложения: Создать приложение, состоящее из трех кнопок: Обо мне, Перевод единиц и Случайное число. Каждая кнопка будет переводить на новое активити и просить ввода некоторых данных, а затем эти данные будем посылать в другую активити для отображения.

Разделим задачу на несколько этапов:

1. Верстка семи активити:
 - a. одна главная, где будут 3 кнопки;
 - b. 3 активити, которые будут открываться только после нажатия кнопки;
 - c. 3 активити для отображения введенных данных.
2. Получение данных с элементов.
3. Передача данных с одной активности в другую.

1. Верстка семи активити

Перед тем как верстать нужно определиться с тем, что будет в каждой активити, а для этого опишем каждую кнопку:

1) Обо мне – эта кнопка открывает новую активити, которая содержит кнопку «Сохранить» и поля ввода: Фамилия, Имя, Возраст, Пол, Электронная почта.

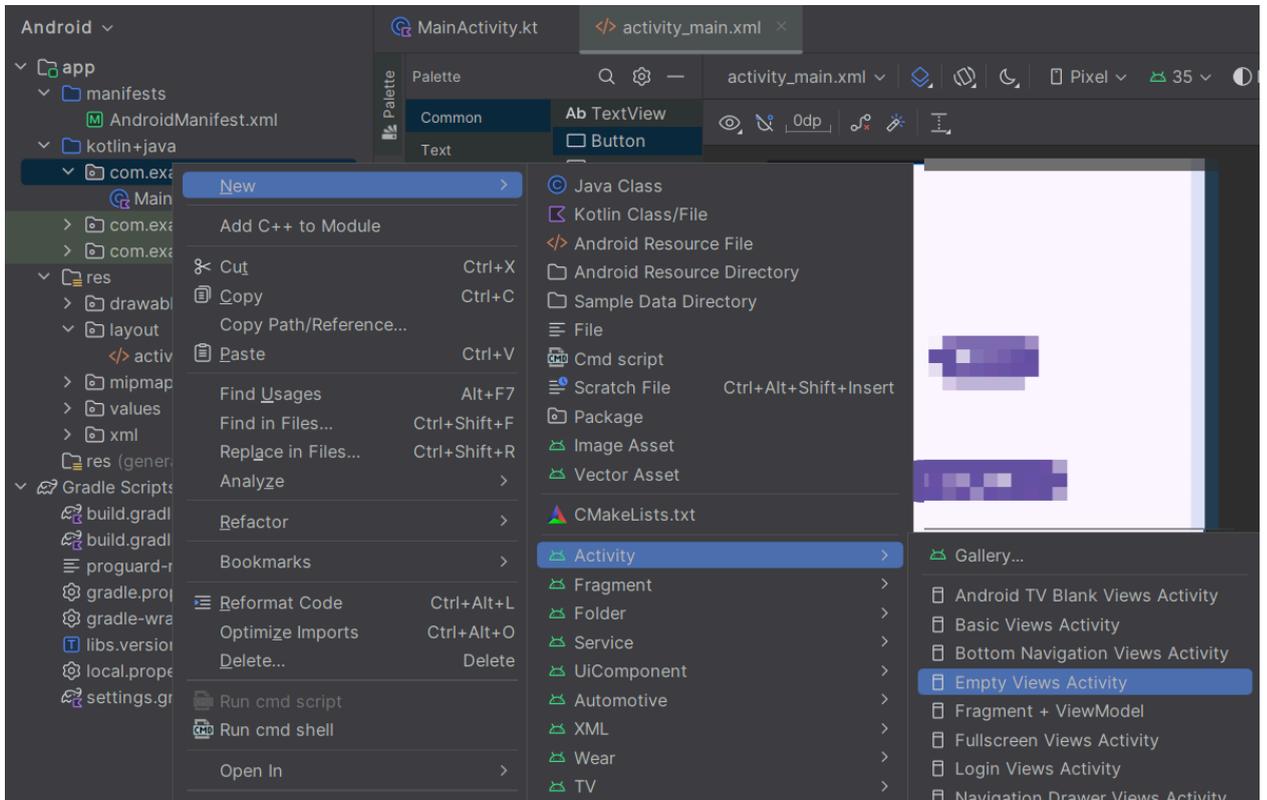
2) Перевод единиц - эта кнопка открывает новую активити, которая содержит кнопку «Сохранить» и числовое поле ввода значения для последующего перевода из полученной единицы измерения количества информации в другие известные.

3) Случайное число - эта кнопка открывает новую активити, которая содержит текстовое поле со случайным числом и кнопки: «Сохранить», «Случайное число».

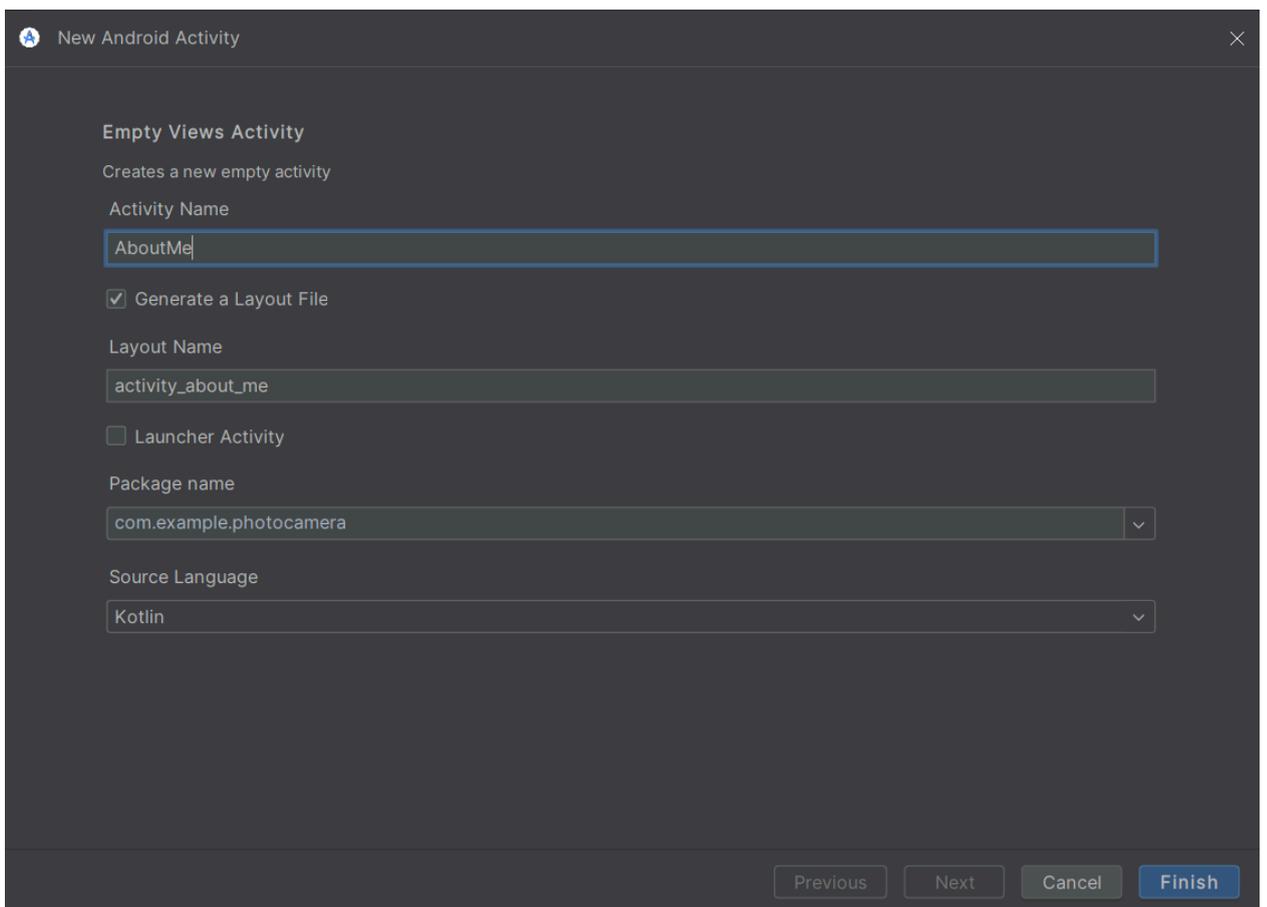
1.1. Создание активностей

Активити содержит не только файл разметки с форматом xml, но и сам класс. По умолчанию при создании проекта создается activity_main.xml и MainActivity.kt, но нам таких активити нужно еще 6. Чтобы не создавать

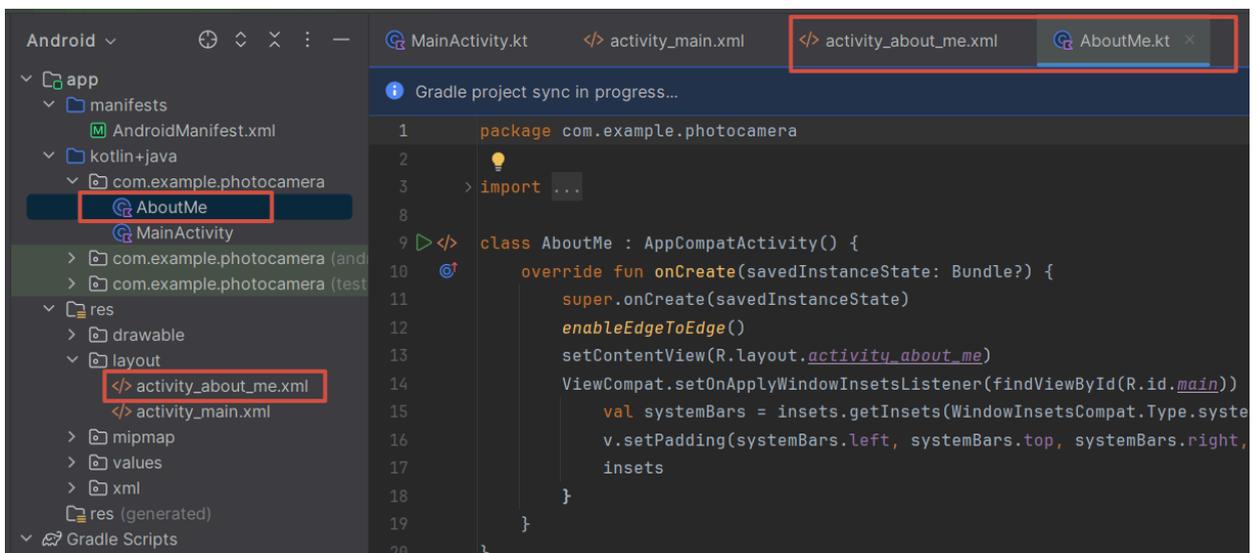
отдельно xml-файлы и классы воспользуемся возможностью Android Studio. Для это нажмите ПКМ по папке, содержащей MainActivity, затем **New – Activity – Empty Views Activity**:



Далее дайте имя, соответствующее названию кнопки и нажмите Finish:



Будут созданы 2 файла в вашем проекте: класс и разметка новой активности (выделено красным на изображении ниже):



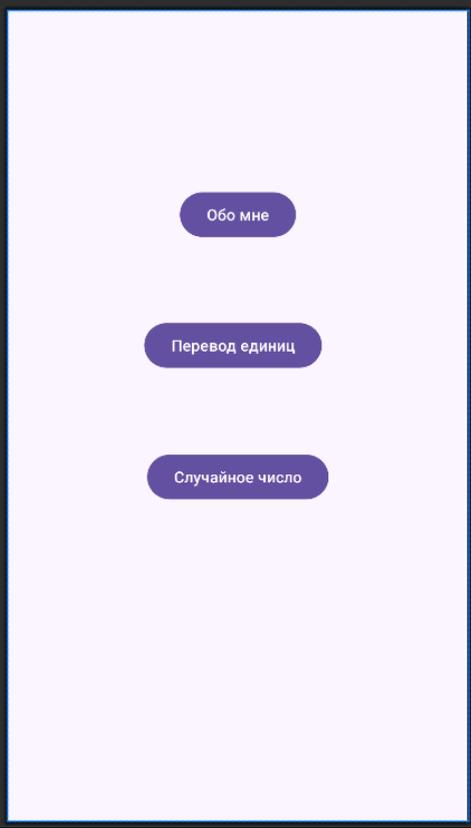
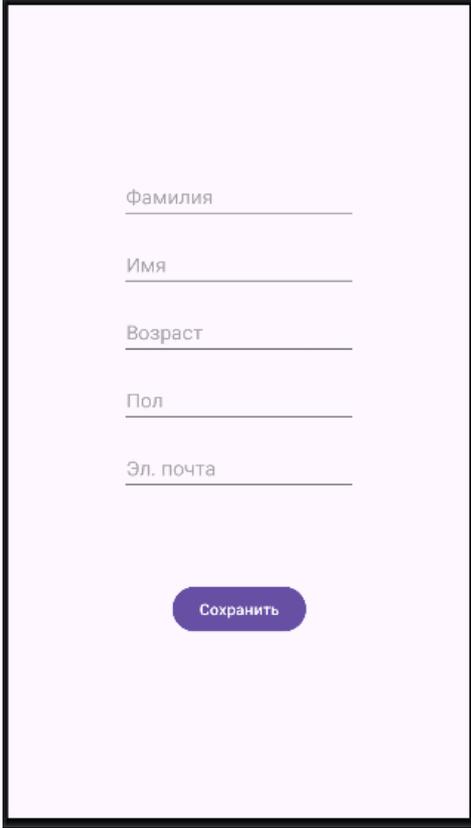
Задание 1: Создайте еще 2 активности с названием, обозначающим наименование кнопок «Перевод единиц» и «Случайное число».

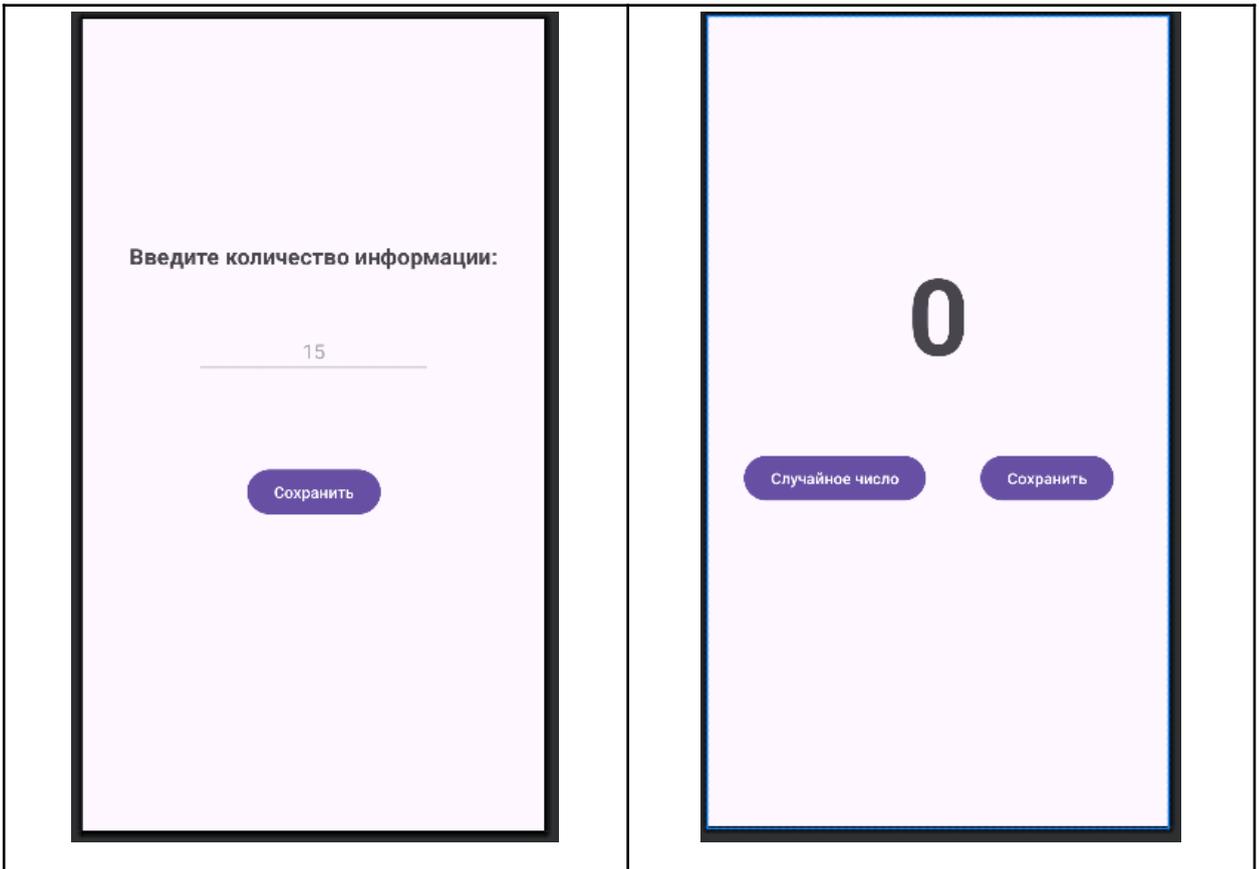
1.2. Верстка созданных активности

Задание 2: Сверстайте как на рисунках ниже.

Примечания:

- Поля ввода используйте соответствующие вводимым данным, т.е. если ожидается число, то Pallet – Text – Number;
- Текст в полях ввода убирайте. Чтобы оставить подсказку для пользователя используйте атрибут **hint**;
- Вы можете использовать, в качестве родительского, любой контейнер;
- Каждому элементу давайте id, чтобы в дальнейшем можно было что-то получить и отловить нужный компонент.
- Для быстрой верстки пользуйтесь окном Attributes.

Главная активити	Обо мне
	
Перевод единиц	Случайное число



1.3. Верстка оставшихся трех активити.

Задание 3: Создайте еще 3 активити, которые будут выводить данные, пришедшие из других, в виде обычного текста.

Информация обо мне	Информация о единицах
---------------------------	------------------------------

<p>Фамилия</p> <p>Имя</p> <p>Возраст</p> <p>Пол</p> <p>Эл. почта</p> <p>Вернуться на главный экран</p>	<p>Кбайт:</p> <p>Мбайт:</p> <p>Гбайт:</p> <p>Тбайт:</p> <p>Вернуться на главный экран</p>
--	---

Информация о случайном числе

<p>0</p> <p>Вернуться на главный экран</p>
--

2. Получение данных с элементов

Обрабатывать каждый элемент не нужно, нам достаточно получить введенные пользователем данные и обработать действие кнопок.

2.1. Переход из одной активности в другую

Наше главное окно имеет 3 кнопки и нужно чтобы каждая открывала новое окно. Для запуска нового экрана необходимо создать экземпляр класса **Intent** и указать в первом параметре текущий класс, а во втором - класс для перехода. После этого вызовем метод **startActivity()**, который и запускает новый экран. Пример запуска экрана «Обо мне» по нажатию на кнопку с id = aboutMe:

```
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.aboutMe.setOnClickListener {
            val intent = Intent(packageContext, this, AboutMe::class.java)
            startActivity(intent)
        }
    }
}
```

Задание 4: Сделайте тоже для двух других кнопок.

2.2. Получение данных

Чтобы получить данные из компонентов экрана, вызовите биндинг соответствующего xml-файла, затем вытащите данные при помощи метода **text** в слушателе кнопки! Сделайте слушатель для кнопки, который будет открывать следующую активность «Информация обо мне»:

```

class AboutMe : AppCompatActivity() {
    lateinit var binding: ActivityAboutMeBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityAboutMeBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.buttonSaveAboutMe.setOnClickListener {
            val familiya = binding.familiya.text.toString()
            val name = binding.name.text.toString()
            val age = binding.age.text.toString()
            val pol = binding.pol.text.toString()
            val adress = binding.adress.text.toString()

            val intent = Intent(packageContext: this, InformationAboutMe::class.java)
            startActivity(intent)
        }
    }
}

```

Задание 5: Сделайте тоже для окна «Перевод единиц». Число вводимое пользователем будет число в Тб. Не забудьте выполнить перевод из полученного числа в другие единицы измерения информации: Кб, Мб, Гб.

2.3. Запустите и протестируйте, убедитесь, что слушатели кнопок правильно работают.

2.4. Окно «Случайное число»

Создайте слушатель кнопки «Случайное число». В нем будем генерировать рандомное число (от 100 до 1000). Затем это число запишем в текстовое поле. Далее при помощи слушателя кнопки «Сохранить» передадим число из текстового поля в окно «Информация о случайном числе».

Для использования класса Random добавьте импорт:

```
import kotlin.random.Random
```

Полный код окна «Случайное число»:

```

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.photocamera.databinding.ActivityRandomBinding
import kotlin.random.Random

class Random : AppCompatActivity() {
    lateinit var binding: ActivityRandomBinding
    var random = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityRandomBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.buttonRandom.setOnClickListener {
            random = Random.nextInt(until: 901) + 100 // 901 - это диапазон (1000 - 100 + 1)
            binding.textRandom.text = random.toString()
        }
        binding.buttonSaveRandom.setOnClickListener {
            val intent = Intent(packageContext: this, InformationRandom::class.java)
            intent.putExtra(name: "random", binding.textRandom.text)
            startActivity(intent)
        }
    }
}

```

3. Передача данных с одной активности в другую

Чтобы передать введенные данные из одной активности в другую нужно задействовать специальную область **extraData**, который имеется у класса **Intent**.

Область **extraData** - это список пар **ключ/значение**, который передается вместе с намерением. В качестве ключей используются строки, а для значений можно использовать любые примитивные типы данных, массивы примитивов, объекты класса **Bundle** и др.

Для передачи данных в другую активность используется метод **putExtra()**. Для получения данных принимающая активность должна вызвать:

- какой-нибудь подходящий типу данных метод: `getIntExtra()`, `getStringExtra()` и т.д., если передано только одно значение;

- если передано несколько значений, то нужно использовать метод **extras**.

3.1. Передача данных

Вернитесь в окно «Обо мне». Мы уже создали переменные, получающие значения из текстовых полей, осталось их передать в окно «Информация обо мне». Как говорилось выше для передачи нужна пара ключ/значение, где ключ должен быть уникальным для каждого значения. У нас уже есть переменная `intent`, которая запускает нужное окно, но перед стартом вызовем все **putExtra()** для передачи данных:

```
binding.buttonSaveAboutMe.setOnClickListener {
    val intent = Intent( packageContext: this, InformationAboutMe::class.java)
    intent.putExtra( name: "famiIiya", familiya)
    intent.putExtra( name: "name", name)
    intent.putExtra( name: "age", age)
    intent.putExtra( name: "poI", poI)
    intent.putExtra( name: "adress", adress)
    startActivity(intent)
}
```

Задание 6: Подобным образом передайте остальные данные в окнах «Перевод единиц» и «Случайное число».

3.2. Получение данных

Вернитесь в окно «Информация обо мне». Здесь будем получать данные из кода «Обо мне» при помощи ключей и соответствующих методов. Сразу будем записывать их в текстовые поля. Так как было передано несколько значений сразу, то воспользуемся методом **extras**, исключая пустые значения (двойной восклицательный знак `!!`). Если они действительно равны `null`, это вызовет исключение `KotlinNullPointerException`.

```

class InformationAboutMe : AppCompatActivity() {
    lateinit var binding: ActivityInformationAboutMeBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityInformationAboutMeBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.textFamilia.text = intent.extras!!.getString(key: "familiya")
        binding.textName.text = intent.extras!!.getString(key: "name")
        binding.textAge.text = intent.extras!!.getString(key: "age")
        binding.textPol.text = intent.extras!!.getString(key: "pol")
        binding.textAdress.text = intent.extras!!.getString(key: "adress")
    }
}

```

Задание 7: Подобным образом получите остальные данные в окнах «Перевод единиц» и «Случайное число».

Задание 8: Откройте файл манифеста «AndroidManifest.xml» и посмотрите какие произошли изменения. Опишите эти изменения.

САМОСТОЯТЕЛЬНАЯ ЧАСТЬ

Задание: Создайте мобильное приложение с тремя и более активити, которые передают любые данные друг другу.

Например: Это может быть активити, в которой имеется минимум 3 поля ввода. Данные из этих полей передаются в другую активити и отображаются в ней в виде текста или того же поля ввода, но уже с полученными значениями. В третьей активити можно сделать вывод какого-нибудь изображения при условии, что данные получены во второй активити и не потерялись в третьей.