

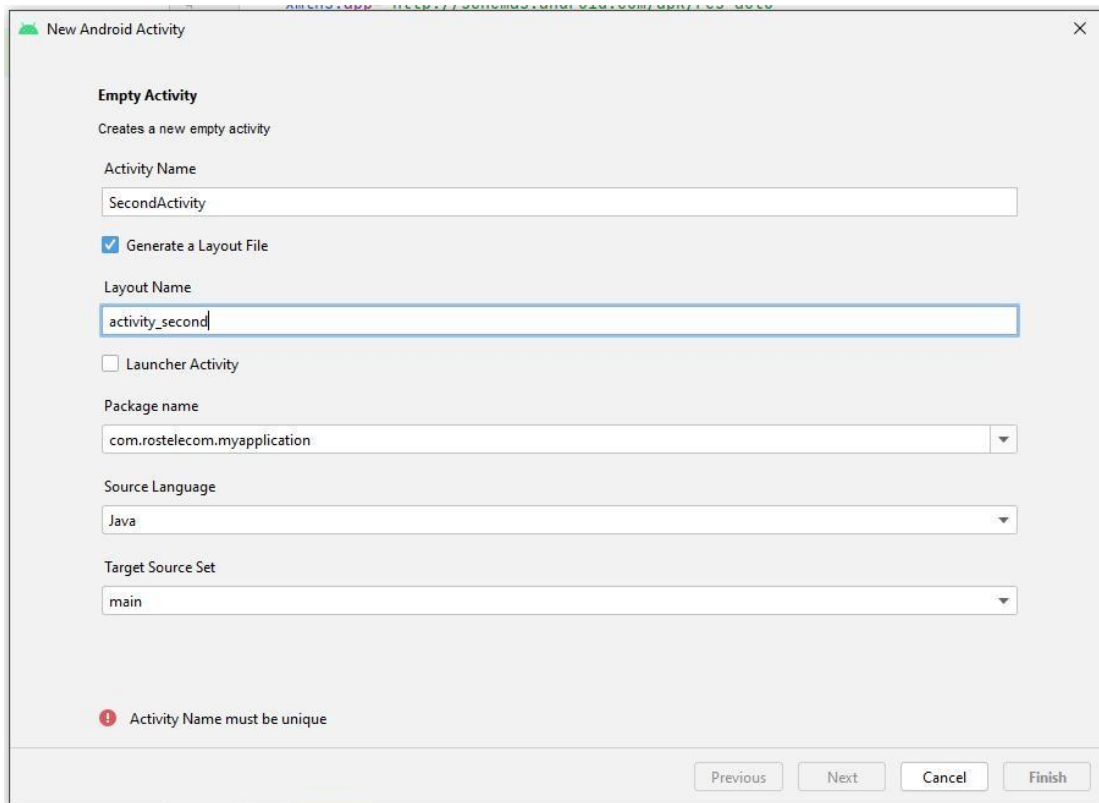
Практическая работа 7

Сериализация

Организуйте передачу данных между активностями тремя разными методами.

Метод 1- putExtra()

Создайте в проекте две activity: **MainActivity** и **SecondActivity**.



В коде **SecondActivity** определите получение данных:

```
package com.rostelecom.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView textView = new TextView( context this);
        textView.setTextSize(26);
        textView.setPadding( left: 16, top: 16, right: 16, bottom: 16);

        Bundle arguments = getIntent().getExtras();

        if(arguments!=null){
            String name = arguments.get("name").toString();
            String company = arguments.getString( key: "company");
            int age = arguments.getInt( key: "age");
            textView.setText("Имя: " + name + "\nОрганизация: " + company +
                "\nВозраст: " + age);
        }

        setContentView(textView);
    }
}
```

Здесь разметка задается с помощью кода языка java, поэтому файл **activity_second.xml** можно оставить без изменений.

В данном случае в **SecondActivity** получает все данных из объекта **Bundle** и выводит их в текстовое поле **TextView**. Данной activity будут передаваться три элемента - две строки с ключами name и company и число с ключом age.

Далее определите передачу в **SecondActivity** данных. Определите для **MainActivity** следующий интерфейс в файле **activity_main.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/nameLabel"
        android:layout_width="0dp"
        android:layout_height="20dp"
        android:text="Имя:"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

    <EditText
        android:id="@+id/name"
        android:layout_width="0dp"
        android:layout_height="48dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nameLabel"
        tools:ignore="SpeakableTextPresentCheck" />

    <TextView
        android:id="@+id/companyLabel"
        android:layout_width="0dp"
        android:layout_height="20dp"
        android:text="Организация:"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/name"/>

    <EditText
        android:id="@+id/company"
        android:layout_width="0dp"
        android:layout_height="48dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/companyLabel"
        tools:ignore="SpeakableTextPresentCheck" />

    <TextView
        android:id="@+id/ageLabel"
        android:layout_width="0dp"
        android:layout_height="20dp"
        android:text="Возраст:"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/company"/>

    <EditText
        android:id="@+id/age"
        android:layout_width="0dp"
        android:layout_height="48dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ageLabel"
        tools:ignore="SpeakableTextPresentCheck" />

    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="320dp"
        android:layout_marginTop="4dp"
        android:onClick="onClick"
        android:text="Save"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/age" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Здесь определены три текстовых поля для ввода данных и кнопка.

В классе **MainActivity** определите следующее содержимое:

```
package com.rostelecom.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick(View v) {

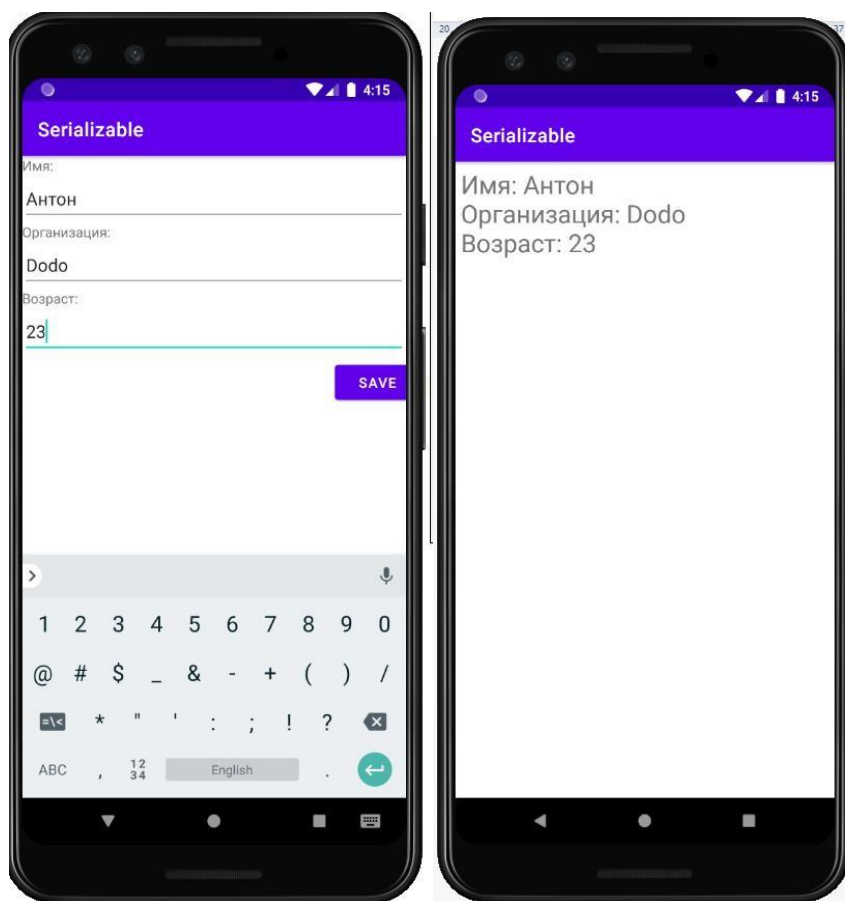
        EditText nameText = findViewById(R.id.name);
        EditText companyText = findViewById(R.id.company);
        EditText ageText = findViewById(R.id.age);

        String name = nameText.getText().toString();
        String company = companyText.getText().toString();
        int age = Integer.parseInt(ageText.getText().toString());

        Intent intent = new Intent( packageContext: this, SecondActivity.class);
        intent.putExtra( name: "name", name);
        intent.putExtra( name: "company", company);
        intent.putExtra( name: "age", age);
        startActivity(intent);
    }
}
```

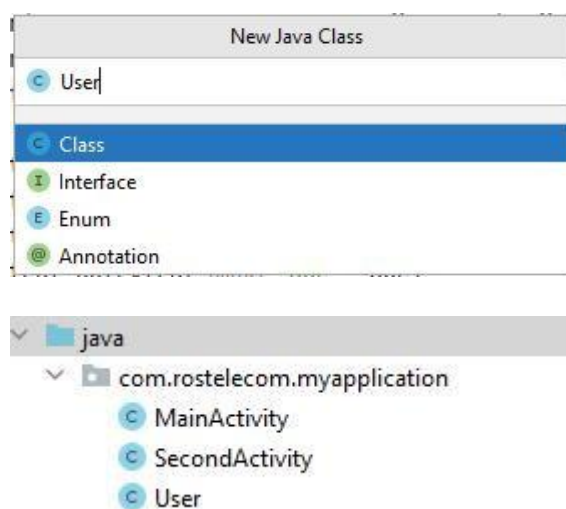
В обработчике нажатия кнопки получаем введенные в текстовые поля **EditText** данные и передаем их в объект **Intent** с помощью метода **putExtra()**. Затем запускаем **SecondActivity**. В итоге при нажатии на кнопку запустится **SecondActivity**, которая получит введенные в текстовые поля данные.

Результат:



Метод 2- Serializable

Также можно передавать более сложные данные. В этом случае используется механизм сериализации. Для этого нажмите на папку пакета, где находятся классы **MainActivity** и **SecondActivity**, правой кнопкой мыши и в контекстном меню выберем **New -> Java Class**



Пусть класс **User** имеет следующий код:

```

package com.rostelecom.myapplication;

import java.io.Serializable;

public class User implements Serializable {

    private String name;
    private String company;
    private int age;

    public User(String name, String company, int age){
        this.name = name;
        this.company = company;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

```

Теперь вместо трех разрозненных данных передается один объект **User**. В качестве ключа используется результат метода **User.class.getSimpleName()**, который по сути возвращает название класса.

Измените класс **SecondActivity**:


```

package com.rostelecom.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_second);
        TextView textView = new TextView( context: this);
        textView.setTextSize(26);
        textView.setPadding( left: 16, top: 16, right: 16, bottom: 16);

        Bundle arguments = getIntent().getExtras();

        User user;
        if(arguments!=null){
            user = (User) arguments.getSerializable(User.class.getSimpleName());

            textView.setText("Имя: " + user.getName() + "\nОрганизация: " + user.getCompany() +
                "\nВозраст: " + String.valueOf(user.getAge()));
        }
        setContentView(textView);
    }
}

```

Код MainActivity

```

package com.rostelecom.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick(View v) {

        EditText nameText = findViewById(R.id.name);
        EditText companyText = findViewById(R.id.company);
        EditText ageText = findViewById(R.id.age);

        String name = nameText.getText().toString();
        String company = companyText.getText().toString();
        int age = Integer.parseInt(ageText.getText().toString());

        User user = new User(name, company, age);

        Intent intent = new Intent( packageContext: this, SecondActivity.class);
        intent.putExtra(User.class.getSimpleName(), user);
        startActivity(intent);
    }
}

```

Метод 3 - Parcelable

Измените класс **User**, теперь он применяет интерфейс **Parcelable**:

```

package com.rostelecom.myapplication;

import android.os.Parcel;
import android.os.Parcelable;

public class User implements Parcelable {

    private String name;
    private String company;
    private int age;

    public static final Creator<User> CREATOR = new Creator<User>() {
        @Override
        public User createFromParcel(Parcel source) {
            String name = source.readString();
            String company = source.readString();
            int age = source.readInt();
            return new User(name, company, age);
        }

        @Override
        public User[] newArray(int size) {
            return new User[size];
        }
    };

    public User(String name, String company, int age){
        this.name = name;
        this.company = company;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCompany() {
        return company;
    }

    public void setCompany(String company) {
        this.company = company;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(name);
        dest.writeString(company);
        dest.writeInt(age);
    }
}

```

Кроме того, объект **Parcelable** должен содержать статическое поле **CREATOR**, которое представляет объект **Creator<User>**. Причем этот объект реализует два метода. Они нужны для создания из ранее сериализованных данных исходных объектов типа **User**.

Так, метод **newArray()** создает массив объектов типа **User**.

Метод **createFromParcel** создает из **Parcel** новый объект типа **User**. То есть этот метод противоположен по действию методу **writeToParcel**. Для получения данных из **Parcel** применяются методы типа **readString()**, **readInt()**, **readParcelable()** и так далее - для чтения определенных типов данных.

Причем важно, что данные в **createFromParcel** считываются из объекта **Parcel** именно в том порядке, в котором они добавляются в этот объект в методе **writeToParcel**.

В activity, которая называется **SecondActivity** будем получать объект **User**, измените код:

```
package com.rostelecom.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView textView = new TextView( context: this);
        textView.setTextSize(26);
        textView.setPadding( left: 16, top: 16, right: 16, bottom: 16);

        Bundle arguments = getIntent().getExtras();

        User user;
        if(arguments!=null){
            user = arguments.getParcelable(User.class.getSimpleName());

            textView.setText("Имя: " + user.getName() + "\nОрганизация: " + user.getCompany() +
                "\nВозраст: " + String.valueOf(user.getAge()));
        }
        setContentView(textView);
    }
}
```

Для получения объекта **Parcelable**, переданного в activity, применяется метод **getParcelable()**. Причем никакого приведения типов не требуется.

Для тестирования передачи **Parcelable** определите в файле **activity_main.xml** простейший интерфейс для **MainActivity**:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/nameLabel"
        android:layout_width="0dp"
        android:layout_height="20dp"
        android:text="Имя:"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

    <EditText
        android:id="@+id/name"
        android:layout_width="0dp"
        android:layout_height="48dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/nameLabel"
        tools:ignore="SpeakableTextPresentCheck" />

    <TextView
        android:id="@+id/companyLabel"
        android:layout_width="0dp"
        android:layout_height="20dp"
        android:text="Организация:"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/name"/>
```



```

1  <EditText
    android:id="@+id/company"
    android:layout_width="0dp"
    android:layout_height="48dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/companyLabel"
    tools:ignore="SpeakableTextPresentCheck" />
3  <TextView
    android:id="@+id/ageLabel"
    android:layout_width="0dp"
    android:layout_height="28dp"
    android:text="Возраст:"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/company"/>
3  <EditText
    android:id="@+id/age"
    android:layout_width="0dp"
    android:layout_height="48dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ageLabel"
    tools:ignore="SpeakableTextPresentCheck" />
3  <Button
    android:id="@+id/btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="328dp"
    android:onClick="onClick"
    android:text="Save"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/age" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

В коде **MainActivity** определите передачу данных в **SecondActivity**:

```

package com.rostelecom.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void onClick(View v) {

        EditText nameText = findViewById(R.id.name);
        EditText companyText = findViewById(R.id.company);
        EditText ageText = findViewById(R.id.age);

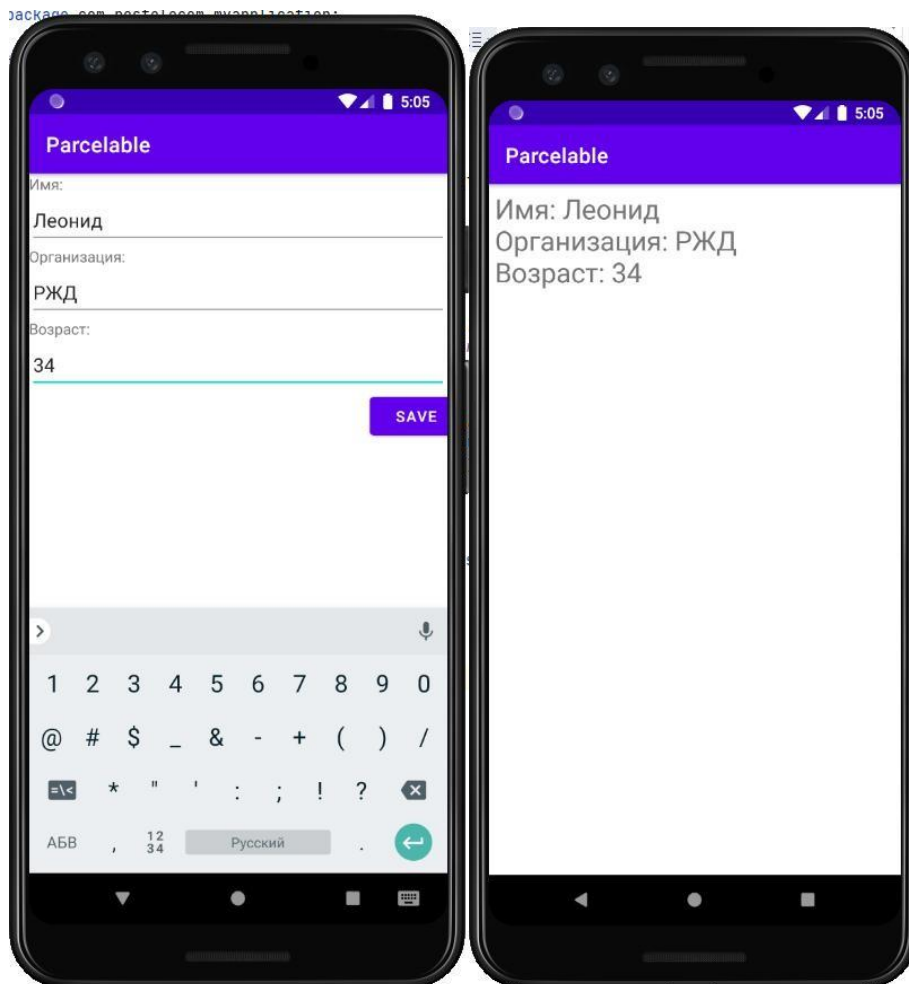
        String name = nameText.getText().toString();
        String company = companyText.getText().toString();
        int age = Integer.parseInt(ageText.getText().toString());

        User user = new User(name, company, age);

        Intent intent = new Intent( packageContext: this, SecondActivity.class);
        intent.putExtra(User.class.getSimpleName(), user);
        startActivity(intent);
    }
}

```

Результат:



Задание: Добавьте в проект еще одно значение, например, телефон или адрес. Организуйте передачу данных при нажатии на кнопку сохранить, любым из указанных методов. Выберите обязательные для заполнения поля, укажите пользователю на них. Перед сохранением, выполните проверку на пустоту обязательных полей.