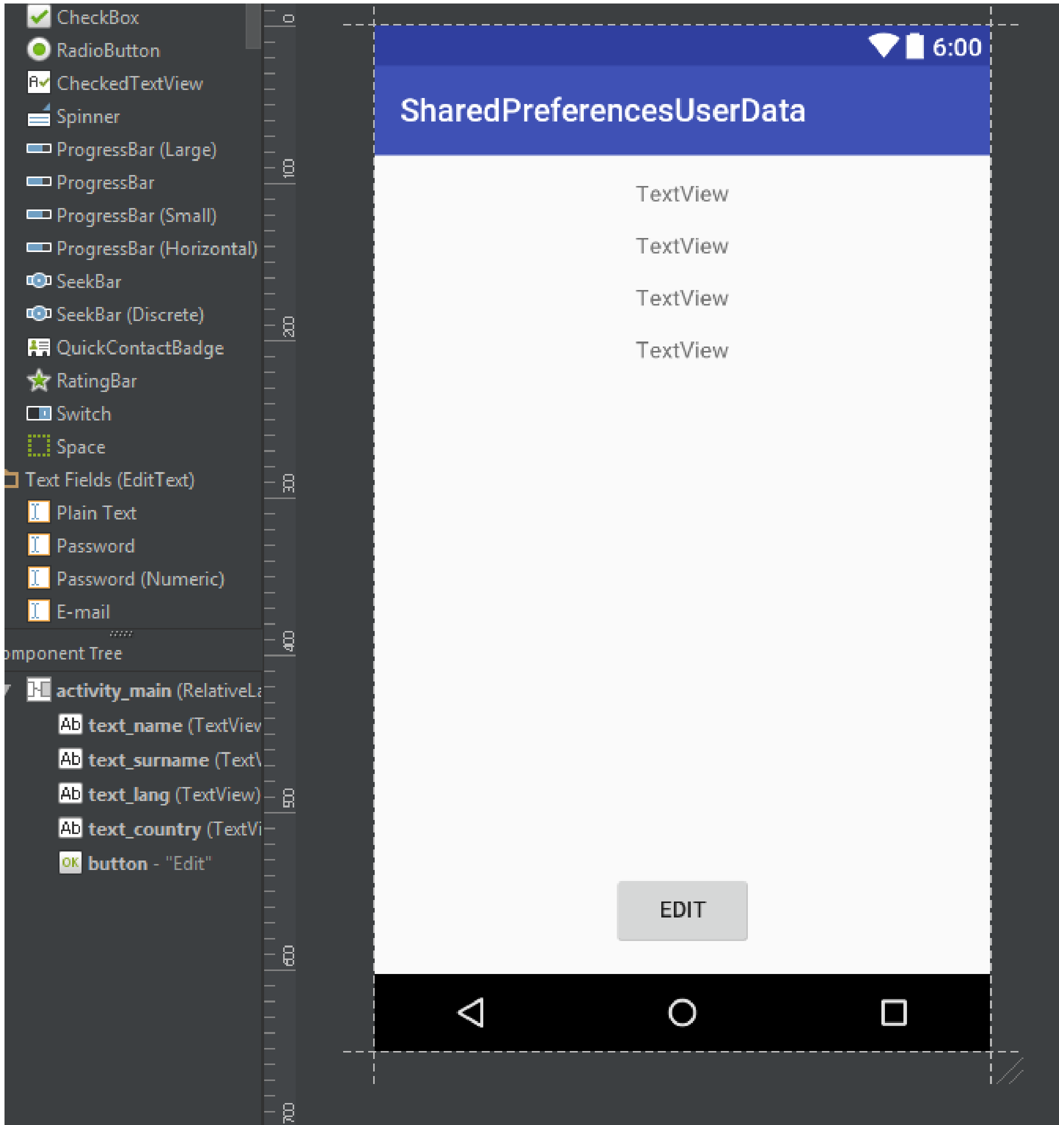
# Java Android

UserDataApplication

# UserDataApplication

Create user interface looking like this:



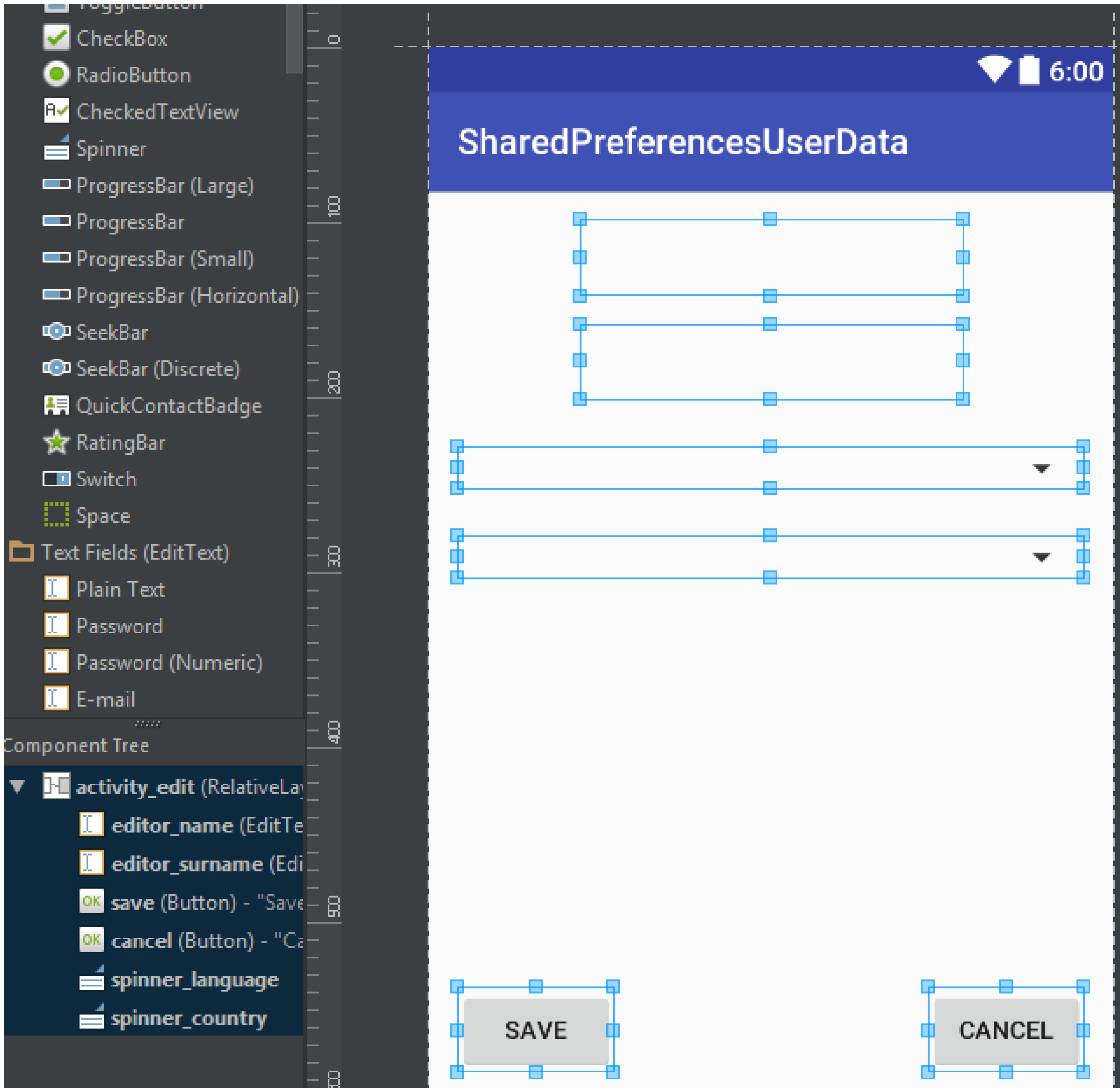
# UserDataApplication

Create user interface looking like this:

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.rent.sharedpreferencesuserdata.MainActivity">
    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/text_name" />
    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_name"
        android:layout_alignStart="@+id/text_name"
        android:layout_marginTop="16dp"
        android:id="@+id/text_surname" />
    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_surname"
        android:layout_alignStart="@+id/text_surname"
        android:layout_marginTop="16dp"
        android:id="@+id/text_lang" />
    <TextView
        android:text="TextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:id="@+id/text_country"
        android:layout_below="@+id/text_lang"
        android:layout_alignStart="@+id/text_lang" />
    <Button
        android:text="Edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/button" />
</RelativeLayout>
```

# UserDataApplication

Create second activity:



Component Tree

- activity_edit (RelativeLay
  - editor_name (EditTe
  - editor_surname (Edi
  - save (Button) - "Save
  - cancel (Button) - "Ca
  - spinner_language
  - spinner_country

# UserDataApplication

Create second activity:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_edit"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.rent.sharedpreferencesuserdata.EditActivity">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:id="@+id/editor_name" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:layout_below="@+id/editor_name"
        android:layout_alignEnd="@+id/editor_name"
        android:layout_marginTop="16dp"
        android:id="@+id/editor_surname" />

    <Button
        android:text="Save"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:id="@+id/save" />
...
```

```
...
    <Button
        android:text="Cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cancel"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true" />

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editor_surname"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="26dp"
        android:id="@+id/spinner_language" />

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/spinner_language"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="26dp"
        android:id="@+id/spinner_country" />

</RelativeLayout>
```

## UserDataApplication

In the main activity, bind controls to their variables:

```java
private TextView text_name;
private TextView text_surname;
private TextView text_lang;
private TextView text_country;

private Button btn;
```

And set OnClickListner on button:

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

    text_name = (TextView) findViewById(R.id.text_name);
    text_surname = (TextView) findViewById(R.id.text_surname);
    text_lang = (TextView) findViewById(R.id.text_lang);
    text_country = (TextView) findViewById(R.id.text_country);

    btn = (Button) findViewById(R.id.button);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

        }
    });
}
```

# UserDataApplication

We want to use button to start new Activity on it’s click Event. So, first we need to create instructions to start this activity. For this, we will use **Intent** and **startActivityForResult** because after it’s closed, we will show short toast with it’s result. I will put my implementation im **startSecondActivity** method.

```
private void startSecondActivity() {
    Intent i = new Intent(getApplicationContext(), EditActivity.class);

    //...

    startActivityForResult(i, 1);
}
```

I will put execution of this method in OnClickListener in the onCreate method:

```
...
    btn = (Button) findViewById(R.id.button);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startSecondActivity();
        }
    });
}
```

# UserDataApplication

Now i will bind EditActivity controls:

```java
private EditText name_input;
private EditText surname_input;

private Button saveBtn, cancelBtn;

private Spinner spin_country, spin_language;
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_edit);

    name_input = (EditText) findViewById(R.id.editor_name);
    surname_input = (EditText) findViewById(R.id.editor_surname);

    spin_language = (Spinner) findViewById(R.id.spinner_language);
    spin_country = (Spinner) findViewById(R.id.spinner_country);

    saveBtn = (Button) findViewById(save);
    cancelBtn = (Button) findViewById(R.id.cancel);

    //...
}
```

# UserDataApplication

First I will define some Method which will return to our MainActivity from EditActivity. I'll call it ***returnToMainActivity.***

```
private void returnToMainActivity() {
    Intent i = new Intent();

    finish();
}
```

But because we used ***startActivityForResult*** we now need to set this activity result. It will be different for save and return button. So I'll pass parameter to my method, which will be result for my activity:

```
private void returnToMainActivity(int result) {
    Intent i = new Intent();

    setResult(result, i);
    finish();
}
```

# UserDataApplication

Now, we need to define two actions for two buttons. First save:
Save button should ave onClickListener which will first – save our preferences, after that it will return to MainActivity with Result **OK**.

```
saveBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // ... Save preferences
        returnToMainActivity(RESULT_OK);
    }
});
```

Second button (Cancel) will work the same way as hardware Back button on our devices.
It will just return to MainActivity, but it will return result **CANCELED**.

```
cancelBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        returnToMainActivity(RESULT_CANCELED);
    }
});
```

# UserDataApplication

Now, let’s get to our preferences. First of all i will define some ***public static final String***s:

```java
public static final String PREFERENCE_NAME = "key_name";
public static final String PREFERENCE_SURNAME = "key_surname";
public static final String PREFERENCE_LANGUAGE = "key_lang";
public static final String PREFERENCE_COUNTRY = "key_country";
```

They are keys for my Shared Preferences. All preferences need keys, and for each key there can only be one value assigned in one preference. These above are keys for all my preferences (we will have 4 preferences). Also I defined one path – which is the name of our shared preferences file:

```java
public static final String PREFERENCES_PATH = "our.preferences.file";
```

Now, let’s get to saving and loading preferences.

# UserDataApplication

Loading preferences. First I will create some variables to hold my preferences (in MainActivity):

```
private String name, surname, countr, language;
```

Now, define Method:

```
private void loadSharedPreferences() {
    SharedPreferences preferences = getSharedPreferences(PREFERENCES_PATH, MODE_PRIVATE);

    name = preferences.getString(PREFERENCE_NAME, "");
    surname = preferences.getString(PREFERENCE_SURNAME, "");
    countr = preferences.getString(PREFERENCE_COUNTRY, "");
    language = preferences.getString(PREFERENCE_LANGUAGE, "");
}
```

This method uses shared preferences from **PREFERENCES_PATH** and loads 4 variables with default value to be empty string. All preferences are being assigned to previously defined variables.

## UserDataApplication

Loading preferences. First I will create some variables to hold my preferences (in MainActivity):

```
private String name, surname, countr, language;
```

Now, define Method:

```
private void loadSharedPreferences() {
    SharedPreferences preferences = getSharedPreferences(PREFERENCES_PATH, MODE_PRIVATE);

    name = preferences.getString(PREFERENCE_NAME, "");
    surname = preferences.getString(PREFERENCE_SURNAME, "");
    countr = preferences.getString(PREFERENCE_COUNTRY, "");
    language = preferences.getString(PREFERENCE_LANGUAGE, "");
}
```

This method uses shared preferences from **PREFERENCES_PATH** and loads 4 variables with default value to be empty string. All preferences are being assigned to previously defined variables.

# UserDataApplication

Now, because we want to load our preferences from SharedPreferences every time the UI activity comes to the first plan, I will use my loadSharedPreferences method in onResume method:

```java
@Override
protected void onResume() {
    loadSharedPreferences();

    super.onResume();
}
```

This way it will be called every time activity shows up. After those preferences are loaded to class fields (variables), we can now easily assign them to TextViews. Let's do this:

```java
@Override
protected void onResume() {
    loadSharedPreferences();

    text_name.setText(name);
    text_surname.setText(surname);
    text_lang.setText(language);
    text_country.setText(countr);

    super.onResume();
}
```

# UserDataApplication

There is only one more thing I’d like to add to the „onResume” method. Whenever activity shows up, we want our user to have those preferences defined. So I’ll add additional check in my onResume method to check if PREFERENCE_NAME and PREFERENCE_SURNAME is empty. If it is empty, than I will call startSecondActivity and force user to define them:

```
@Override
protected void onResume() {
    loadSharedPreferences();

    if (name.isEmpty() && surname.isEmpty()) {
        startSecondActivity();
    }

    text_name.setText(name);
    text_surname.setText(surname);
    text_lang.setText(language);
    text_country.setText(countr);

    super.onResume();
}
```

This way second activity will show up every time when those fields are empty.

# UserDataApplication

Let’s get back to EditActivity. First –saving preferences:

```
private void savePreferences() {
    SharedPreferences prefs = getSharedPreferences(PREFERENCES_PATH, MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();

    editor.putString(PREFERENCE_NAME, name_input.getText().toString());
    editor.putString(PREFERENCE_SURNAME, surname_input.getText().toString());
    editor.putString(PREFERENCE_LANGUAGE, spin_language.getSelectedItem().toString());
    editor.putString(PREFERENCE_COUNTRY, spin_country.getSelectedItem().toString());

    editor.commit();
}
```

Ok, now go back to the onCreate method, there we had previously defined **saveBtn.** It’s onClickListener had one commented line waiting for us to implement savePreferences method. Let’s add this line:

```
saveBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        savePreferences();
        returnToMainActivity(RESULT_OK);
    }
});
```

# UserDataApplication

Now, filling up our spinners. I will use resources in my implementation. In the **res/values/strings.xml** define:

```xml
<resources>
    <string name="app_name">SharedPreferencesUserData</string>
    <string-array name="languages">
        <item>Polish</item>
        <item>English</item>
        <item>French</item>
        <item>Vietnamese</item>
    </string-array>

    <string-array name="countries">
        <item>Poland</item>
        <item>England</item>
        <item>France</item>
        <item>Vietnam</item>
    </string-array>
</resources>
```

Two string arrays named **countries** and **languages**.

# UserDataApplication

In the onCreate method of EditActivity, implement adding those values to the spinners:

```
ArrayAdapter<String> spinnerCountriesAdapter = new ArrayAdapter<>(this,
        R.layout.support_simple_spinner_dropdown_item,
        getResources().getStringArray(R.array.countries));
spin_country.setAdapter(spinnerCountriesAdapter);

ArrayAdapter<String> spinnerLanguageAdapter = new ArrayAdapter<>(this,
        R.layout.support_simple_spinner_dropdown_item,
        getResources().getStringArray(R.array.languages));
spin_language.setAdapter(spinnerLanguageAdapter);
```

I will add additional feature which will check every text change on both name_input and surname_input fields to watch if both fields are or aren’t empty. If they are, i will disable save and cancel buttons (because i don’t want our users to leave those fields empty). We’ll do this using TextWatcher class:

```
TextWatcher watcher = new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {      }
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        checkIfEmptyFields();
    }
    @Override
    public void afterTextChanged(Editable s) {      }
};
```

For checkEmptyFields method, look on the next slide:

# UserDataApplication

checkEmptyFields method will check if PREFERENCE_NAME field and PREFERENCE_SURNAME field is empty (same way as checking variables in MainActivity.onResume method).

```java
private void checkIfEmptyFields() {
    if (name_input.getText().toString().isEmpty() && surname_input.getText().toString().isEmpty()) {
        cancelBtn.setEnabled(false);
        saveBtn.setEnabled(false);
    } else {
        cancelBtn.setEnabled(true);
        saveBtn.setEnabled(true);
    }
}
```

I will disallow users to go back or save whenever those fields are empty.

# UserDataApplication

We can use TextWatcher same way as Listeners, and we can add them using addTextWatcherListener on each input:

```
name_input.addTextChangedListener(watcher);
surname_input.addTextChangedListener(watcher);
```

# UserDataApplication

Current result:

```java
public class EditActivity extends AppCompatActivity {
    private EditText name_input;
    private EditText surname_input;
    private Button saveBtn, cancelBtn;
    private Spinner spin_country, spin_language;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit);

        name_input = (EditText) findViewById(R.id.editor_name);
        surname_input = (EditText) findViewById(R.id.editor_surname);
        spin_language = (Spinner) findViewById(R.id.spinner_language);
        spin_country = (Spinner) findViewById(R.id.spinner_country);

        saveBtn = (Button) findViewById(save);
        cancelBtn = (Button) findViewById(R.id.cancel);

        ArrayAdapter<String> spinnerCountriesAdapter = new ArrayAdapter<>(this,
                R.layout.support_simple_spinner_dropdown_item,
                getResources().getStringArray(R.array.countries));
        spin_country.setAdapter(spinnerCountriesAdapter);

        ArrayAdapter<String> spinnerLanguageAdapter = new ArrayAdapter<>(this,
                R.layout.support_simple_spinner_dropdown_item,
                getResources().getStringArray(R.array.languages));
        spin_language.setAdapter(spinnerLanguageAdapter);

        cancelBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                returnToMainActivity(RESULT_CANCELED);
            }
        });
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                savePreferences();
                returnToMainActivity(RESULT_OK);
            }
        });
```

# UserDataApplication

Current result **EditActivity**:

```java
        checkIfEmptyFields();

        TextWatcher watcher = new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {      }
            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                checkIfEmptyFields();
            }
            @Override
            public void afterTextChanged(Editable s) {       }
        };
        name_input.addTextChangedListener(watcher);
        surname_input.addTextChangedListener(watcher);
    }

    private void checkIfEmptyFields() {
        if (name_input.getText().toString().isEmpty() && surname_input.getText().toString().isEmpty()) {
            cancelBtn.setEnabled(false);
            saveBtn.setEnabled(false);
        } else {
            cancelBtn.setEnabled(true);
            saveBtn.setEnabled(true);
        }
    }
    private void savePreferences() {
        SharedPreferences prefs = getSharedPreferences(PREFERENCES_PATH, MODE_PRIVATE);
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString(PREFERENCE_NAME, name_input.getText().toString());
        editor.putString(PREFERENCE_SURNAME, surname_input.getText().toString());
        editor.putString(PREFERENCE_LANGUAGE, spin_language.getSelectedItem().toString());
        editor.putString(PREFERENCE_COUNTRY, spin_country.getSelectedItem().toString());

        editor.commit();
    }
    private void returnToMainActivity(int result) {
        Intent i = new Intent();
        setResult(result, i);
        finish();
    }
}
```

# UserDataApplication

Last thing to do is implementing toasts on activityResult:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (resultCode == RESULT_OK) {
        Toast.makeText(getApplicationContext(), "OK", Toast.LENGTH_SHORT).show();

    } else {
        Toast.makeText(getApplicationContext(), "Not", Toast.LENGTH_SHORT).show();

    }
}
```

In MainActivity

# UserDataApplication – using intent

I will now implement additional intent passing from first to second activity, to set preferences values on controls. In **MainActivity:**

```
private void startSecondActivity() {
    Intent i = new Intent(getApplicationContext(), EditActivity.class);

    i.putExtra(PREFERENCE_NAME, name);
    i.putExtra(PREFERENCE_SURNAME, surname);
    i.putExtra(PREFERENCE_COUNTRY, countr);
    i.putExtra(PREFERENCE_LANGUAGE, language);

    startActivityForResult(i, 1);
}
```

And in EditActivity in onCreate method (at the end of it):

```
name_input.setText(getIntent().getStringExtra(PREFERENCE_NAME));
surname_input.setText(getIntent().getStringExtra(PREFERENCE_SURNAME));

String lang = getIntent().getStringExtra(PREFERENCE_LANGUAGE);
String country = getIntent().getStringExtra(PREFERENCE_COUNTRY);

if (lang != null && !lang.isEmpty()) {
    int positionLang = spinnerLanguageAdapter.getPosition(lang);
    spin_language.setSelection(positionLang);
}
if (country != null && !country.isEmpty()) {
    int positionCountry = spinnerLanguageAdapter.getPosition(country);
    spin_country.setSelection(positionCountry);
}
```

# UserDataApplication – using intent

Whole MainActivity:

```java
public class MainActivity extends AppCompatActivity {

    public static final String PREFERENCE_NAME = "key_name";
    public static final String PREFERENCE_SURNAME = "key_surname";
    public static final String PREFERENCE_LANGUAGE = "key_lang";
    public static final String PREFERENCE_COUNTRY = "key_country";

    public static final String PREFERENCES_PATH = "our.preferences.file";

    private TextView text_name;
    private TextView text_surname;
    private TextView text_lang;
    private TextView text_country;

    private Button btn;

    private String name, surname, countr, language;

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

        text_name = (TextView) findViewById(R.id.text_name);
        text_surname = (TextView) findViewById(R.id.text_surname);
        text_lang = (TextView) findViewById(R.id.text_lang);
        text_country = (TextView) findViewById(R.id.text_country);

        btn = (Button) findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startSecondActivity();
            }
        });
    }
```

# UserDataApplication – using intent

Whole MainActivity:

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK) {
            Toast.makeText(getApplicationContext(), "OK", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Not", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    protected void onResume() {
        loadSharedPreferences();
        if (name.isEmpty() && surname.isEmpty()) {
            startSecondActivity();
        }

        text_name.setText(name);
        text_surname.setText(surname);
        text_lang.setText(language);
        text_country.setText(countr);
        super.onResume();
    }

    private void loadSharedPreferences() {
        SharedPreferences preferences = getSharedPreferences(PREFERENCES_PATH, MODE_PRIVATE);
        name = preferences.getString(PREFERENCE_NAME, "");
        surname = preferences.getString(PREFERENCE_SURNAME, "");
        countr = preferences.getString(PREFERENCE_COUNTRY, "");
        language = preferences.getString(PREFERENCE_LANGUAGE, "");
    }

    private void startSecondActivity() {
        Intent i = new Intent(getApplicationContext(), EditActivity.class);
        i.putExtra(PREFERENCE_NAME, name);
        i.putExtra(PREFERENCE_SURNAME, surname);
        i.putExtra(PREFERENCE_COUNTRY, countr);
        i.putExtra(PREFERENCE_LANGUAGE, language);
        startActivityForResult(i, 1);
    }
}
```

The end