

## Лабораторна робота №7. Робота з віджетами **CheckBoxes**, **RadioButtons**, **ToggleButton**s, **Spinners** та обробка подій

**Мета:** вивчити порядок роботи з найбільш поширеними віджетами.

### Теоретичні відомості

#### Прапорець (**CheckBox**).

Елемент **CheckBox** дозволяє користувачу обрати один чи декілька об'єктів з множини (рис. 1).

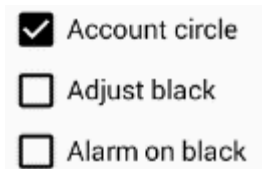


Рис. 1

Для того, щоб користувач міг обрати декілька об'єктів, потрібно оброблювати подію натиснення кожного з елементів **CheckBox**.

**Обробка натискання на CheckBox.** Для обробки події натискання може використовуватися атрибут **android:onClick**, який задається в xml файлі макета. Значення атрибута – ім'я метода обробника події натискання.

Наприклад, два елемента **CheckBox** описуються в xml файлі (рис. 2).

```
<CheckBox android:id="@+id/checkbox_meat"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/meat"
  android:onClick="onCheckboxClicked"/>
<CheckBox android:id="@+id/checkbox_cheese"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/cheese"
  android:onClick="onCheckboxClicked"/>
```

Рис. 2

Визначення метода відбувається **onCheckboxClicked** в java-класі (рис. 3)

```

public void onCheckboxClicked(View view) {
    // Is the view now checked?
    boolean checked = ((CheckBox) view).isChecked();

    // Check which checkbox was clicked
    switch(view.getId()) {
        case R.id.checkbox_meat:
            if (checked)
                // Put some meat on the sandwich
            else
                // Remove the meat
            break;
        case R.id.checkbox_cheese:
            if (checked)
                // Cheese me
            else
                // I'm lactose intolerant
            break;
        // TODO: Veggie sandwich
    }
}

```

Рис. 3

**Приклад 1.** Потрібно розробити додаток, який при натисканні кнопки циклічно змінює картинку, які обрані за допомогою CheckBox. Можлива реалізація виглядає наступним чином.

Файл activity\_mail.xml.

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    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.alex.lab10.MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/change_images" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:layout_marginTop="26dp"
        android:src="@drawable/ic_account_circle_black_48dp"
        android:layout_below="@+id/textView"
        android:layout_toEndOf="@+id/textView" />

    <CheckBox
        android:layout_width="wrap_content"

```

```
android:layout_height=>>wrap_content>>
android:text=>>@string/account_circle>>
android:id=>>@+id/checkBox>>
android:checked=>>true>>
android:layout_alignBaseline=>>@+id/button>>
android:layout_alignBottom=>>@+id/button>>
android:layout_alignEnd=>>@+id/imageView>>
android:onClick=>>checkImage>>/>
```

<CheckBox

```
android:layout_width=>>wrap_content>>
android:layout_height=>>wrap_content>>
android:text=>>@string/adjust_black>>
android:id=>>@+id/checkBox2>>
android:checked=>>false>>
android:layout_below=>>@+id/checkBox>>
android:layout_alignRight=>>@+id/imageView>>
android:onClick=>>checkImage>>/>
```

<CheckBox

```
android:layout_width=>>wrap_content>>
android:layout_height=>>wrap_content>>
android:text=>>@string/alarm_on_black>>
android:id=>>@+id/checkBox3>>
android:checked=>>false>>
android:layout_below=>>@+id/checkBox2>>
android:layout_alignRight=>>@+id/imageView>>
android:onClick=>>checkImage>>/>
```

<CheckBox

```
android:layout_width=>>wrap_content>>
android:layout_height=>>wrap_content>>
android:text=>>@string/assinstant_black>>
android:id=>>@+id/checkBox4>>
android:checked=>>false>>
android:layout_below=>>@+id/checkBox3>>
android:layout_alignRight=>>@+id/imageView>>
android:onClick=>>checkImage>>/>
```

<CheckBox

```
android:layout_width=>>wrap_content>>
android:layout_height=>>wrap_content>>
android:text=>>Brihtness>>
android:id=>>@+id/checkBox5>>
android:checked=>>false>>
android:layout_below=>>@+id/checkBox4>>
android:layout_alignRight=>>@+id/imageView>>
android:onClick=>>checkImage>>/>
```

<TableRow

```
android:layout_width=>>match_parent>>
android:layout_height=>>match_parent>></TableRow>
```

<TableRow

```
android:layout_width=>>match_parent>>
android:layout_height=>>match_parent>>
```

<Button

```
android:id=>>@+id/button>>
android:text=>>@string/change_image>>
android:textColor=>>@color/colorAccent>>
android:layout_below=>>@+id/checkBox5>>
android:layout_toEndOf=>>@+id/imageView>>
android:layout_column=>>25>>
android:layout_gravity=>>center_horizontal>> />
```

```
</TableRow>
</TableLayout>
```

## Файл MainActivity.java

```
package com.example.alex.lab10;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.ImageView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ArrayList<ImageView> images = new ArrayList<>();
    Button myButton1;
    ImageView imageView;
    CheckBox image1, image2, image3, image4, image5;
    int count=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView=(ImageView) findViewById(R.id.imageView);
        CheckBox image1=(CheckBox) findViewById(R.id.checkBox);
        CheckBox image2=(CheckBox) findViewById(R.id.checkBox2);
        CheckBox image3=(CheckBox) findViewById(R.id.checkBox3);
        CheckBox image4=(CheckBox) findViewById(R.id.checkBox4);
        CheckBox image5=(CheckBox) findViewById(R.id.checkBox5);

        image1.setChecked(true);
        images.add(R.drawable.ic_account_circle_black_48dp);
        imageView.setImageDrawable(getResources().getDrawable((int)
images.get(0)));

        myButton1 = (Button) findViewById(R.id.button);
        myButton1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                count=(count+1)%5;
                imageView.setImageDrawable(getResources().getDrawable((int)
images.get(count)));
            }
        });

        public void checkImage(View v){
            boolean ch = ((CheckBox) v).isChecked();
            switch (v.getId()){
                case R.id.checkBox:
                    if (ch)
                        images.add(R.drawable.ic_account_circle_black_48dp);
                    else
                        images.remove(R.drawable.ic_account_circle_black_48dp);
                    break;
                case R.id.checkBox2:
                    if (ch)
                        images.add(R.drawable.ic_adjust_black_48dp);
```

```

        else
            images.remove(R.drawable.ic_adjust_black_48dp);
        break;
    case R.id.checkBox3:
        if (ch)
            images.add(R.drawable.ic_alarm_on_black_48dp);
        else
            images.remove(R.drawable.ic_alarm_on_black_48dp);
        break;
    case R.id.checkBox4:
        if (ch)
            images.add(R.drawable.ic_assistant_black_48dp);
        else
            images.remove(R.drawable.ic_assistant_black_48dp);
        break;
    case R.id.checkBox5:
        if (ch)
            images.add(R.drawable.ic_brightness_4_black_48dp);
        else
            images.remove(R.drawable.ic_brightness_4_black_48dp);
        break;
    }
}
}

```

**Перемикач (RadioGroup, RadioButton).** **RadioButton** дозволяє користувачу вибрати один об'єкт з множини. Для групування декількох елементів **RadioButton**, один з яких повинен бути ввімкнено,

використовується **RadioGroup** (рис. 4).



Рис.4

**Обробки натискання RadioButton.** Використовується підхід, аналогічний до **CheckBox** (використання атрибута **android:onClick**) (рис. 5).

```

<?xml version="1.0" encoding="utf-8"?>
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <RadioButton android:id="@+id/radio_pirates"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pirates"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton android:id="@+id/radio_ninjas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ninjas"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>

```

```

public void onRadioButtonClicked(View view) {
    // Is the button now checked?
    boolean checked = ((RadioButton) view).isChecked();

    // Check which radio button was clicked
    switch(view.getId()) {
        case R.id.radio_pirates:
            if (checked)
                // Pirates are the best
            break;
        case R.id.radio_ninjas:
            if (checked)
                // Ninjas rule
            break;
    }
}

```

Рис. 5

Кнопка-перемикач (**ToggleButton**). Використовується для вибору одного з двох можливих налаштувань\станів (рис. 6). Перемикання кнопки виконується за допомогою методів

**CompoundButton.setChecked()** або **CompoundButton.toggle()**.



Рис. 6

Обробка натискання **ToggleButton** за допомогою методів **CompoundButton.OnCheckedChangeListener**, **setOnCheckedChangeListener** (рис. 7).

```

ToggleButton toggle = (ToggleButton) findViewById(R.id.togglebutton);
toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            // The toggle is enabled
        } else {
            // The toggle is disabled
        }
    }
});

```

Рис. 7

**Контейнер ScrollView**. При великому обсязі інформації, яку потрібно помістити на екран пристрою використовується віджет **ScrollView** – контейнерний елемент, який наслідується від **ViewGroup**.

Методи `scrollBy()` і `scrollTo()` дозволяють програмно прокручувати контент. Наприклад, можна організувати автоматичне прокручування під час читання.

Приклад 2. Потрібно реалізувати додаток відображення великого тексту і прокрутки його за натисканням кнопки «Up» догори і кнопки «Down» донизу. При ввімкненій кнопці-перемикачі прокрутка кнопками працює, інакше – ні.

Можлива реалізація виглядає наступним чином.

Файл `activity_mail.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: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.alex.lab103.MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:id="@+id/scrollView"
        android:layout_alignParentStart="true"
        android:layout_marginStart="23dp"
        android:layout_marginTop="36dp"
        android:scrollbars="vertical"
        android:layout_alignParentEnd="false" >

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Large Text"
            android:id="@+id/textView2" />

    </ScrollView>
    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New ToggleButton"
        android:id="@+id/toggleButton"
        android:layout_alignParentBottom="true"
        android:layout_alignEnd="@+id/scrollView"
        android:checked="false" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/to_begin"
        android:id="@+id/button"
        android:layout_alignTop="@+id/toggleButton"
        android:layout_alignStart="@+id/scrollView" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/down"
        android:id="@+id/button2"
```

```

        android:layout_above="@+id/button"
        android:layout_alignStart="@+id/scrollView" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/up"
    android:id="@+id/button3"
    android:layout_above="@+id/button2"
    android:layout_alignStart="@+id/scrollView" />

</RelativeLayout>

```

## Файл MainActivity.java.

```

package com.example.alex.lab103;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {
    TextView myText;
    Button onTop, up, down;
    ToggleButton mySwitch;
    ScrollView myScroll;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myText=(TextView) findViewById(R.id.textView2);
        onTop=(Button) findViewById(R.id.button);
        up=(Button) findViewById(R.id.button2);
        down=(Button) findViewById(R.id.button3);
        mySwitch=(ToggleButton) findViewById(R.id.toggleButton);
        myScroll=(ScrollView) findViewById(R.id.scrollView);
        myText.setText(R.string.my_text);

        mySwitch.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if (isChecked) {
                    up.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            myScroll.scrollBy(0, +20);
                        }
                    });
                }

                down.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        myScroll.scrollTo(0, -20);
                    }
                });
            }
        });
    }
}

```





Наприклад, якщо множина можливих значень для **Spinner** відома заздалегідь, її можна визначити як строковий масив у файлі string.xml (рис. 10).

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="planets_array">
    <item>Mercury</item>
    <item>Venus</item>
    <item>Earth</item>
    <item>Mars</item>
    <item>Jupiter</item>
    <item>Saturn</item>
    <item>Uranus</item>
    <item>Neptune</item>
  </string-array>
</resources>
```

Рис. 10

З масивом описаним на рис. 10, можна використовувати наступний код у файлі java-класу відповідного activity (рис. 11).

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
// Create an ArrayAdapter using the string array and a default spinner layout
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    R.array.planets_array, android.R.layout.simple_spinner_item);
// Specify the layout to use when the list of choices appears
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinner.setAdapter(adapter);
```

Рис. 11

Метод **createFromResource()** дозволяє створити **ArrayAdapter** зі строкового масива. Третій аргумент метода **createFromResource()** визначає layout у якому будуть відображатися елементи для вибору. Макет **android.R.layout.simple\_spinner\_item** є вбудованим в платформу, так само, як і **android.R.layout.simple\_spinner\_dropdown\_item**, який використовується для представлення спадного меню з варіантами вибору.

**Відповідь на вибір зі списку.** Коли користувач вибере один з варіантів, які пропонуються у спадному меню, об'єкт **Spinner** отримує подію **on-item-selected**. Обробка цієї події може виконуватись наступним чином (рис. 12).

```

public class SpinnerActivity extends Activity implements OnItemSelectedListener {
    ...

    public void onItemSelected(AdapterView<?> parent, View view,
        int pos, long id) {
        // An item was selected. You can retrieve the selected item using
        // parent.getItemAtPosition(pos)
    }

    public void onNothingSelected(AdapterView<?> parent) {
        // Another interface callback
    }
}

Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);

```

Рис. 12

Для визначення обробника події вибору застосовується обробник **AdapterView.OnItemSelectedListener** і методи **onItemSelected()** та **onNothingSelected()**.

### Завдання до лабораторної роботи

1. Протестуйте програму з прикладу 1 та виправіть знайдені помилки. При необхідності використовуйте режим Debug.
2. У попередньому додатку додайте на макет декілька елементів **RadioButton** з функцією вибору кольору фону кнопки.
3. Створіть додаток сортування числових даних різними алгоритмами. Приблизний макет інтерфейса (рис. 13)

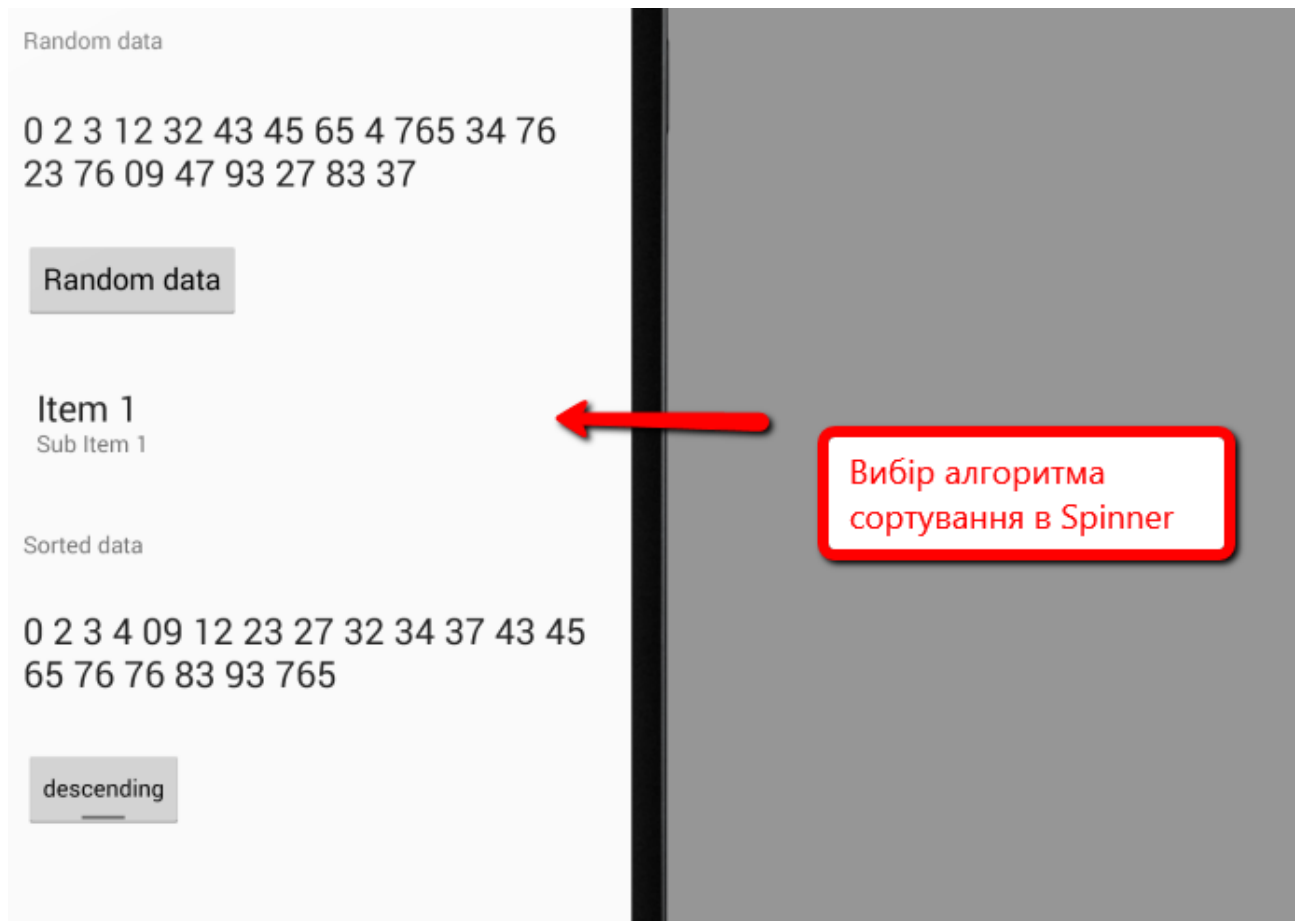


Рис. 13

### Контрольні запитання

1. Які основні віджети Android додатку ви знаєте?
2. Яким чином оброблюється натискання кнопки?
3. Яким чином оброблюється вибір одного варіанту з множини можливих в елементі Spinner?