

**КАЛУЖСКИЙ ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Э. БАУМАНА (национальный исследовательский университет)»**



Факультет «Информатика и управление»

Кафедра "Программное обеспечение ЭВМ, информационные технологии"

Intent объекты

Permissions

AndroidManifest.xml

```
<!--Declaring the required permissions-->
<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
    android:name="android.permission.CAMERA" />
```

Java

```
int permissionCheck = ContextCompat.checkSelfPermission(this,
    Manifest.permission.READ_PHONE_STATE);
if (permissionCheck != PackageManager.PERMISSION_GRANTED) {
    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
        Manifest.permission.READ_PHONE_STATE)) {
        showExplanation("Permission Needed", "Rationale",
            Manifest.permission.READ_PHONE_STATE,
            REQUEST_PERMISSION_PHONE_STATE);
    } else {
        requestPermission(Manifest.permission.READ_PHONE_STATE,
            REQUEST_PERMISSION_PHONE_STATE);
    }
} else {
    Toast.makeText(MainActivity.this, "Permission (already) Granted!",
        Toast.LENGTH_SHORT).show();
}
```

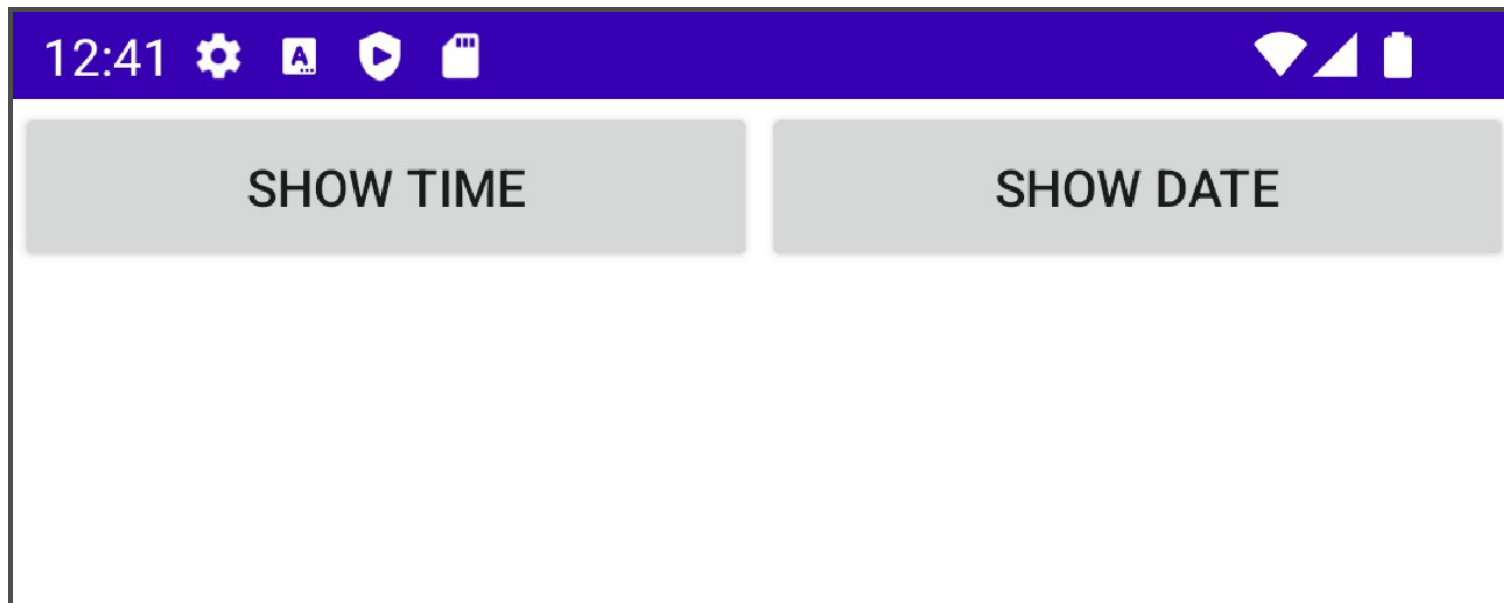
Kotlin

```
if (ContextCompat.checkSelfPermission(this@MainActivity,
    Manifest.permission.ACCESS_FINE_LOCATION) !=
    PackageManager.PERMISSION_GRANTED) {
    if (ActivityCompat.shouldShowRequestPermissionRationale(this@MainActivity,
        Manifest.permission.ACCESS_FINE_LOCATION)) {
        ActivityCompat.requestPermissions(this@MainActivity,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), 1)
    } else {
        ActivityCompat.requestPermissions(this@MainActivity,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), 1)
    }
}
```

Создадим приложение, которое будет отображать текущее время или дату. Реализация этого будет выполнена с помощью трех Activity:

- первое будет содержать две кнопки: Show time и Show date
- второе будет отображать время
- третье будет отображать дату

Нажатие на кнопку Show time будет вызывать второе Activity, а нажатие на кнопку Show date – третье Activity. При это реализация будет осуществлена посредством Intent Filter.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <Button
        android:id="@+id/btnTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Show time" >

    </Button>

    <Button
        android:id="@+id/btnDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Show date" />

</LinearLayout>
```

//Java

```
public class MainActivity extends Activity implements OnClickListener {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnTime = (Button) findViewById(R.id.btnTime);
        Button btnDate = (Button) findViewById(R.id.btnDate);
        btnTime.setOnClickListener(this);
        btnDate.setOnClickListener(this);
    }

    // @Override
    public void onClick(View v) {
        Intent intent;

        switch(v.getId()) {
            case R.id.btnTime:
                intent = new Intent("lec8_1.intent.action.showtime");
                startActivity(intent);
                break;
            case R.id.btnDate:
                intent = new Intent("lec8_1.intent.action.showdate");
                startActivity(intent);
                break;
        }
    }
}
```

//Kotlin

```
class MainActivity : AppCompatActivity() {

    /** Called when the activity is first created. */
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnTime.setOnClickListener(this::onClick)
        btnDate.setOnClickListener(this::onClick)
    }

    fun onClick(view: View){
        var intent : Intent

        when (view.id) {
            R.id.btnTime ->{
                intent = Intent("lec8_1.intent.action.showtime")
                startActivity(intent)
            }
            R.id.btnDate ->{
                intent = Intent("lec8_1.intent.action.showdate")
                startActivity(intent)
            }
        }
    }
}
```


В коде определены кнопки и обработчики нажатий. В методе `onClickListener` определяется какая кнопка была нажата и создается `Intent`.

Для создания `Intent` используем конструктор: **`Intent (String action)`**. Т.е. при создании заполняется атрибут объекта `Intent`, который называется `action`. Это обычная строковая константа. `Action` обычно указывает действие, которое мы хотим произвести. Например, есть следующие системные `action`-константы: `ACTION_VIEW` - просмотр, `ACTION_EDIT` – редактирование, `ACTION_PICK` – выбор из списка, `ACTION_DIAL` – сделать звонок.

Если действие производится с чем-либо, то в пару к `action` идет еще один `Intent`-атрибут – `data`. В нем можно указать какой-либо объект: пользователь в адресной книге, координаты на карте, номер телефона и т.п. Т.е. `action` указывает что делать, а `data` – с чем делать.

Как видно из кода в примере используются следующие action:

```
lec8_1.intent.action.showtime  
lec8_1.intent.action.showdate
```

Первый будет означать, что будет вызвана Activity, которая отобразит текущее время. Второй – Activity с датой.

Здесь надо четко понимать следующее: action – это просто текст, поэтому он может быть произвольным. Текст showtime – отражает то, что требуется сделать, он нагляднее и понятнее. А префикс lec8_1.intent.action. используется, чтобы не было коллизий. В системе может быть приложение, которое уже использует action showtime – наш с ним не должен пересекаться. Поэтому action называется lec8_1.intent.action.showtime.

Итак, был создан Intent с action и запущен в систему для поиска Activity. Чтобы Activity подошла, надо чтобы ее Intent Filter содержал атрибут action с тем же значением, что и action в Intent. Значит необходимо создать две Activity, настроить их IntentFilter и реализовать отображение времени и даты.

Activity создается как обычно - создается класс ActivityShowTime с суперклассом android.app.Activity и прописываем его в манифесте как Activity. После того, как прописали в манифесте, надо будет там же создать Intent Filter.

```
//AndroidManifest.xml
```

```
</activity>
```

```
<activity
```

```
    android:name=".ActivityShowTime"
```

```
    android:exported="true">
```

```
    <intent-filter>
```

```
        <action android:name="lec8_1.intent.action.showtime" />
```

```
        <category android:name="android.intent.category.DEFAULT"
```

```
    />
```

```
    </intent-filter>
```

```
</activity>
```

Создадим layout для нового Activity, назовем его show_time.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tvTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="50dp"
        android:text="Show Time"
        android:textSize="40dp" >
    </TextView>

</LinearLayout>
```

//Java

```
package com.example.lec8_1;
import java.sql.Date;
import java.text.SimpleDateFormat;

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

public class ActivityShowTime extends Activity {

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

        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");
        String time = sdf.format(new Date(System.currentTimeMillis()));

        TextView tvTime = (TextView) findViewById(R.id.tvTime);
        tvTime.setText(time);
    }
}
```

//Kotlin

```
package com.example.lec8_1
```

```
import android.os.Bundle
```

```
import android.widget.TextView
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
import java.text.SimpleDateFormat
```

```
import java.util.*
```

```
class ActivityShowTime : AppCompatActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.show_time)
```

```
        var sdf = SimpleDateFormat("HH:mm:ss")
```

```
        var time = sdf.format(Date(System.currentTimeMillis()))
```

```
        var tvTime = findViewById(R.id.tvTime) as TextView
```

```
        tvTime.setText(time)
```

```
    }
```

```
}
```

Т.е. Intent с action = `lec8_1.intent.action.showtime` нашел и отобразил Activity, у которого action также равен `lec8_1.intent.action.showtime` в Intent Filter.

Вернемся назад (кнопка Back) и нажмем теперь кнопку Show date. Приложение выдаст ошибку, т.к. оно не смогло найти Activity, которое соответствовало бы Intent с action = **`lec8_1.intent.action.showdate`** (создано только для showtime).

Создадим такое Activity, назовем его ActivityShowDate. Действия все те же самые, что и при создании ActivityTime:

- создание класса
- создание Activity в манифесте и создание для него Intent Filter (с action = `lec8_1.intent.action.showdate` и category = `android.intent.category.DEFAULT`)

Layout-файл назовем show_date.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tvDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:text="Show Date"
        android:textSize="30sp" >
    </TextView>

</LinearLayout>
```


Код ActivityShowDate.java:

//Java

```
import java.sql.Date;
import java.text.SimpleDateFormat;

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

public class ActivityShowDate extends Activity {

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

        SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy");
        String date = sdf.format(new Date(System.currentTimeMillis()));

        TextView tvDate = (TextView) findViewById(R.id.tvDate);
        tvDate.setText(date);
    }
}
```

//Kotlin

```
package com.example.lec8_1
```

```
import android.os.Bundle
```

```
import android.widget.TextView
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
import java.text.SimpleDateFormat
```

```
import java.util.*
```

```
class ActivityShowDate : AppCompatActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.show_date)
```

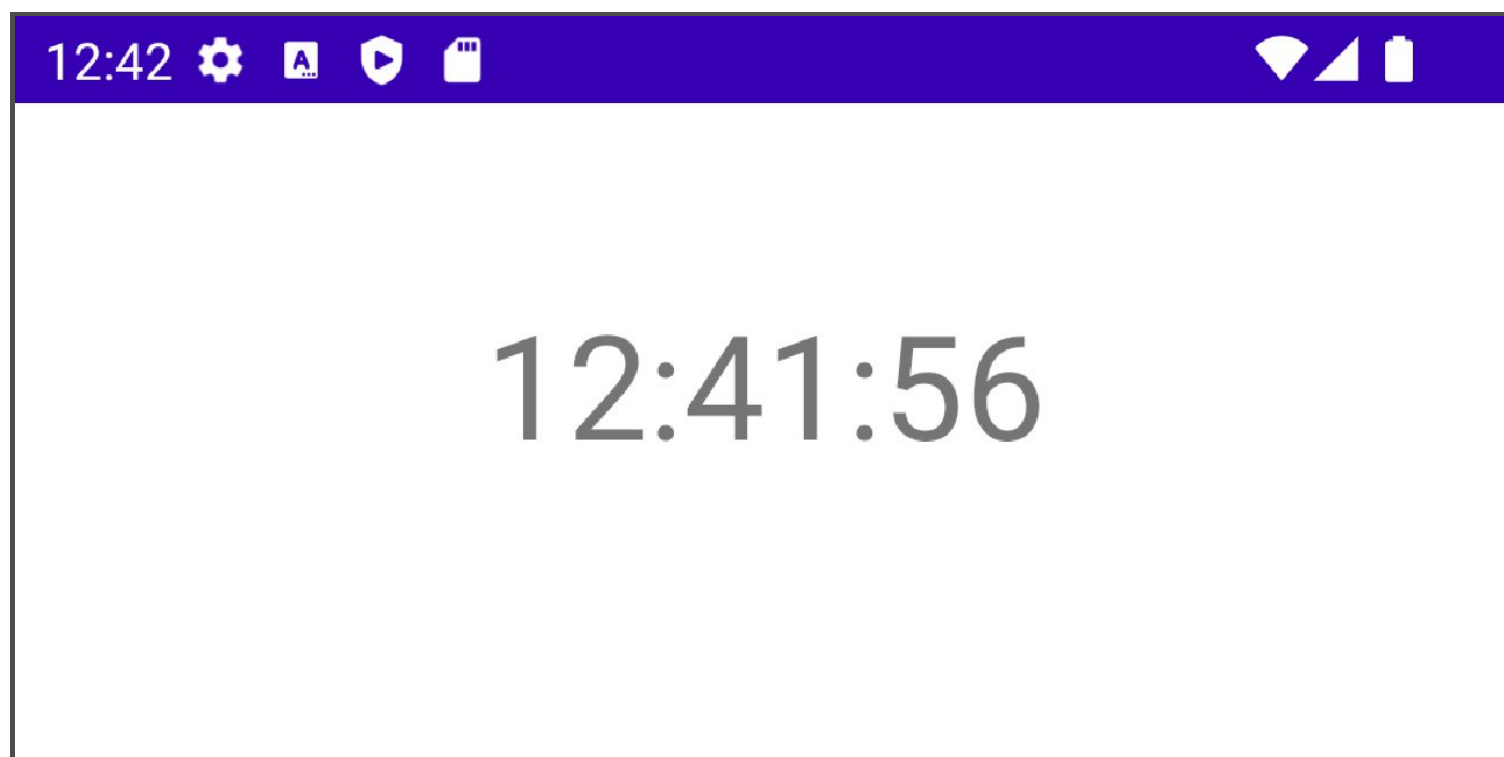
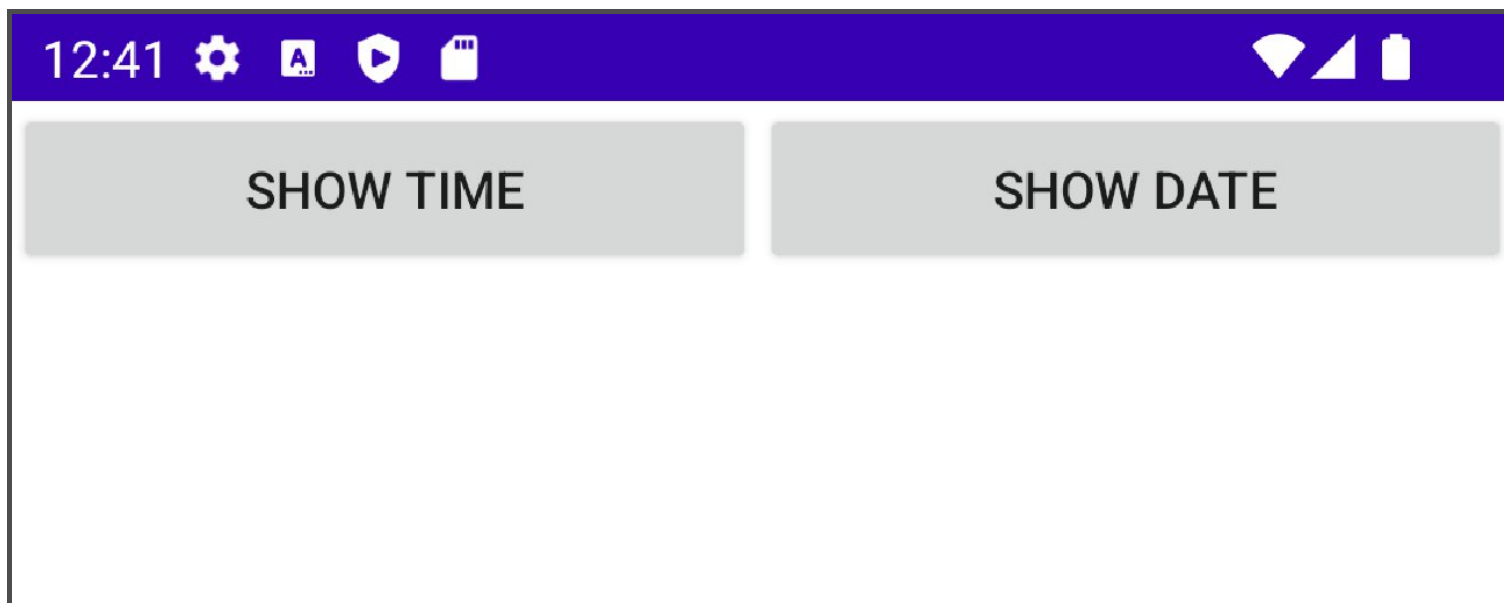
```
        var sdf = SimpleDateFormat("dd.MM.yyyy")
```

```
        var date = sdf.format(Date(System.currentTimeMillis()))
```

```
        var tvDate = findViewById(R.id.tvDate) as TextView  
        tvDate.setText(time)
```

```
    }
```

```
}
```



Intent Filter может содержать в себе несколько компонентов action. Тем самым Activity может выполнять несколько действий. Например, не только просмотр картинки, но и редактирование. Получается, что одно Activity может подойти разным Intent с разными action.

Помимо этого Activity, которое было вызвано с помощью Intent, имеет доступ к этому Intent и может прочесть его атрибуты. Т.е. может узнать какой action использовался.

Рассмотрим приложение, которое демонстрирует данные концепции.

Создадим Activity и настроим Intent Filter на два action = lec8_1.intent.action.showtime и action = lec8_1.intent.action.showdate. Тем самым мы обозначаем, что это Activity способно показать и время и дату. Далее мы будем создавать Intent либо с action lec8_1.intent.action.showtime, либо с lec8_1.intent.action.showdate, но оба они будут вызывать одно Activity. Для того чтобы Activity определяло что отображать дату или время будет производиться чтение action из Intent и по нему определяться соответствующий параметр.

Создаем разметку для Activity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <Button
        android:id="@+id/btnTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Show time" >
    </Button>

    <Button
        android:id="@+id/btnDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Show date" >
    </Button>

</LinearLayout>
```

//Java

Код для MainActivity.java

```
package com.example.lec8_2;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity implements OnClickListener {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnTime = (Button)findViewById(R.id.btnTime);
        Button btnDate = (Button)findViewById(R.id.btnDate);

        btnTime.setOnClickListener(this);
        btnDate.setOnClickListener(this);
    }

    public void onClick(View v) {
        Intent intent;

        switch(v.getId()) {
            case R.id.btnTime:
                intent = new Intent("lec8_1.intent.action.showtime");
                startActivity(intent);
                break;
            case R.id.btnDate:
                intent = new Intent("lec8_1.intent.action.showdate");
                startActivity(intent);
                break;
        }
    }
}
```

//Kotlin

```
package com.example.lec8_2
```

```
import android.content.Intent
```

```
import androidx.appcompat.app.AppCompatActivity
```

```
import android.os.Bundle
```

```
import android.view.View
```

```
import kotlinx.android.synthetic.main.activity_main.*
```

```
class MainActivity : AppCompatActivity() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        setContentView(R.layout.activity_main)
```

```
        btnTime.setOnClickListener(this::onClick)
```

```
        btnDate.setOnClickListener(this::onClick)
```

```
    }
```

```
    fun onClick(view: View) {
```

```
        var intent : Intent
```

```
        when(view.id) {
```

```
            R.id.btnTime ->{
```

```
                intent = Intent("lec8_1.intent.action.showtime")
```

```
                startActivity(intent)
```

```
            }
```

```
            R.id.btnDate ->{
```

```
                intent = Intent("lec8_1.intent.action.showdate")
```

```
                startActivity(intent)
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

Код и разметка для MainActivity остается такой же, как и в прошлом примере. Теперь необходимо создать Activity, которая будет обрабатывать оба этих Intent.

Необходимо создать layout-файл timedate.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tvInfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:text="TextView"
        android:textSize="30sp" >
    </TextView>

</LinearLayout>
```


//Java

```
public class ActivityInfo extends Activity {

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

        // получаем Intent, который вызывал это Activity
        Intent intent = getIntent();
        // читаем из него action
        String action = intent.getAction();

        String format = "", textInfo = "";

        // в зависимости от action заполняем переменные
        if (action.equals("lec8_1.intent.action.showtime")) {
            format = "HH:mm:ss";
            textInfo = "Time: ";
        }
        else if (action.equals("lec8_1.intent.action.showdate")) {
            format = "dd.MM.yyyy";
            textInfo = "Date: ";
        }

        // в зависимости от содержимого переменной format
        // получаем дату или время в переменную datetime
        SimpleDateFormat sdf = new SimpleDateFormat(format);
        String datetime = sdf.format(new Date(System.currentTimeMillis()));

        TextView tvDate = (TextView) findViewById(R.id.tvInfo);
        tvDate.setText(textInfo + datetime);
    }
}
```

//Kotlin

```
class ActivityInfo : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.timedate)

        // получаем Intent, который вызывал это Activity
        var intent : Intent = getIntent()
        // читаем из него action
        var action : String? = intent.getAction()

        var format = ""
        var textInfo = ""

        // в зависимости от action заполняем переменные
        if (action.equals("lec8_1.intent.action.showtime")) {
            format = "HH:mm:ss"
            textInfo = "Time: "
        }
        else if (action.equals("lec8_1.intent.action.showdate")) {
            format = "dd:MM:yyyy"
            textInfo = "Date: "
        }

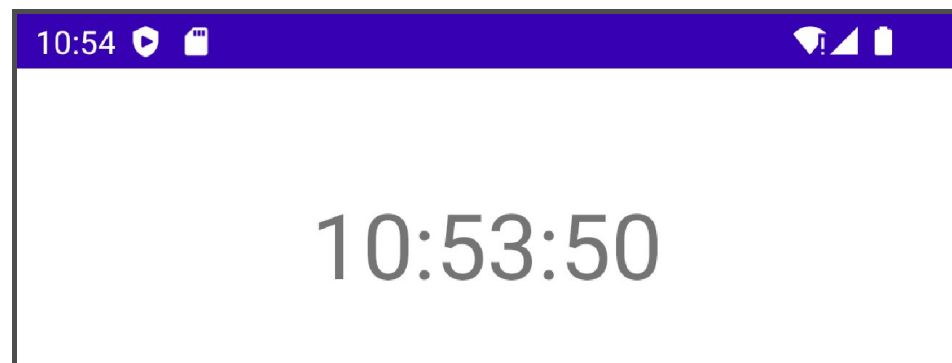
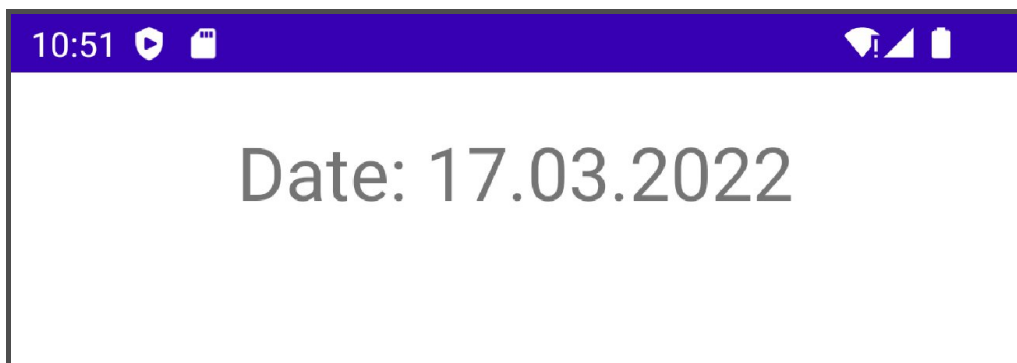
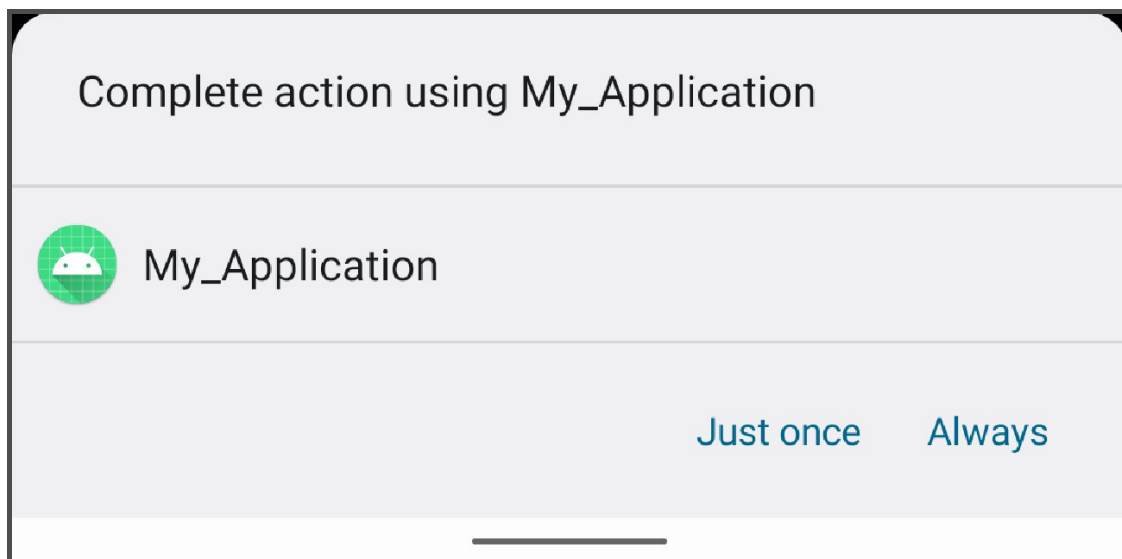
        // в зависимости от содержимого переменной format
        // получаем дату или время в переменную datetime
        var sdf = SimpleDateFormat(format)
        var dateTime = sdf.format(Date(System.currentTimeMillis()))

        var tvInfo = findViewById(R.id.tvInfo) as TextView
        tvInfo.setText(textInfo + dateTime)
    }
}
```

Intent возвращается с помощью метода getIntent(), из него читается action и в зависимости от значения формируется текст и выводится на экран.

Новое Activity необходимо прописать в AndroidManifest.xml и создать ему Intent Filter с двумя action и category в качестве label можно указать Date/time info.

Intent Filter для ActivityInfo содержит два action. Следовательно если придет Intent с любым из них – то Activity будет вызвана.



Передача данных с помощью Intent

Рассмотрим простое приложение. На первом экране вводится имя и фамилия, а второй экран будет эти данные отображать. Данные будут передаваться внутри Intent.

Создаем разметку для первого Activity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res
/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:gravity="center_horizontal"
        android:text="Input your Name" >
    </TextView>
```

```
<TableLayout
```

```
    android:id="@+id/tableLayout1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:stretchColumns="1" >
```

```
<TableRow
```

```
    android:id="@+id/tableRow1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" >
```

```
    <TextView
```

```
        android:id="@+id/textView1"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="First Name" >
```

```
    </TextView>
```

```
    <EditText
```

```
        android:id="@+id/etFName"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginLeft="5dp" >
```

```
        <requestFocus>
```

```
        </requestFocus>
```

```
    </EditText>
```

```
</TableRow>
```

```
<TableRow
```

```
    android:id="@+id/tableRow2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" >
```

```
    <TextView
```

```
        android:id="@+id/textView2"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Last Name" >
```

```
    </TextView>
```

```
    <EditText
```

```
        android:id="@+id/etLName"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginLeft="5dp" >
```

```
    </EditText>
```

```
</TableRow>
```

```
</TableLayout>
```

```
<Button
```

```
    android:id="@+id/btnSubmit"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:text="OK" >
```

```
</Button>
```

```
</LinearLayout>
```

В EditText будем вводиться имя и фамилия, а кнопка ОК будет вызывать другой экран и передавать ему эти данные.

Код для MainActivity.java

//Java

```
public class MainActivity extends Activity implements OnClickListener {

    EditText etFName;
    EditText etLName;

    Button btnSubmit;

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

        etFName = (EditText) findViewById(R.id.etFName);
        etLName = (EditText) findViewById(R.id.etLName);

        btnSubmit = (Button) findViewById(R.id.btnSubmit);
        btnSubmit.setOnClickListener(this);
    }

    public void onClick(View v) {
        Intent intent = new Intent(this, ResultActivity.class);
        intent.putExtra("fname", etFName.getText().toString());
        intent.putExtra("lname", etLName.getText().toString());
        startActivity(intent);
    }
}
```

//Kotlin

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnSubmit.setOnClickListener(this::onClick)
    }

    fun onClick(view : View){
        var intent = Intent(this@MainActivity,
ResultActivity::class.java)
        var etFName = findViewById(R.id.etFName) as TextView
        var etLName = findViewById(R.id.etLName) as TextView
        intent.putExtra("fname", etFName.getText().toString())
        intent.putExtra("lname", etLName.getText().toString())
        startActivity(intent)
    }
}
```


В коде определяются поля ввода и кнопка. Кнопке присваиваем обработчик – Activity (this). Рассмотрим реализацию метода onClick. Intent создается с использованием класса, а не action. Это означает, что система просмотрит манифест файл приложения, и если найдет Activity с таким классом – отобразит его. ResultActivity пока не создан, поэтому код будет подчеркнут красным. Это не мешает нам сохранить файл. Чуть позже мы создадим это Activity и ошибка исчезнет.

Итак, Intent создан. Используется метод putExtra. Он имеет множество вариаций и аналогичен методу put для Map, т.е. добавляет к объекту пару. Первый параметр – это ключ (имя), второй - значение.

В Intent поместили два объекта с именами: fname и lname. fname содержит значение поля etFName, lname – значение поля etLName. Остается только отправить укомплектованный Intent с помощью метода startActivity.

Теперь создадим второе Activity. Назовем его ResultActivity.
Создаем для него layout-файл result.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tvView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:text="TextView"
        android:textSize="20sp" >
    </TextView>

</LinearLayout>
```

TextView отображает пришедшие данные.

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class ResultActivity extends Activity {

    TextView tvView;

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

        tvView = (TextView) findViewById(R.id.tvView);

        Intent intent = getIntent();

        String fName = intent.getStringExtra("fname");
        String lName = intent.getStringExtra("lname");

        tvView.setText("Your name is: " + fName + " " + lName);
    }
}
```

//Kotlin

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

class ResultActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.result)

        var tvView = findViewById(R.id.tvView) as TextView

        var fName = intent.getStringExtra("fname")
        var lName = intent.getStringExtra("lname")

        tvView.setText("Your name is: " + fName + " " + lName)
    }
}
```

Находим TextView, затем получаем Intent и извлекаем из него String-объекты с именами fname и lname. Это те самые объекты, которые мы помещали в коде MainActivity.java. Формируем строку вывода в TextView с использованием полученных данных.

11:22

Input your Name

First Name

Last Name

OK

Alexandr | Alexandria | Alexandra

q w e r t y u i o p

a s d f g h j k l

↑ z x c v b n m ↵

?123 , 😊 . ↵

▽

11:23

Your name is: Pushkin Alexandr

//Java

Изменение цвета Activity

```
public class MainActivity extends Activity implements OnClickListener {
```

```
    EditText Redit, Gedit, Bedit;
```

```
    Button btnSubmit;
```

```
    /** Called when the activity is first created. */
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Redit = (EditText) findViewById(R.id.Rcolor);
```

```
        Gedit = (EditText) findViewById(R.id.Gcolor);
```

```
        Bedit = (EditText) findViewById(R.id.Bcolor);
```

```
        btnSubmit = (Button) findViewById(R.id.btnSubmit);
```

```
        btnSubmit.setOnClickListener(this);
```

```
    }
```

```
    public void onClick(View v) {
```

```
        Intent intent = new Intent(this, ResultActivity.class);
```

```
        intent.putExtra("Rc", Redit.getText().toString());
```

```
        intent.putExtra("Gc", Gedit.getText().toString());
```

```
        intent.putExtra("Bc", Bedit.getText().toString());
```

```
        startActivity(intent);
```

```
    }
```

```
}
```

//Kotlin

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnSubmit.setOnClickListener(this::onClick)
    }

    fun onClick(view : View){
        var intent = Intent(this@MainActivity,
ResultActivity::class.java)
        var Redit = findViewById(R.id.Rcolor) as TextView
        var Gedit = findViewById(R.id.Gcolor) as TextView
        var Bedit = findViewById(R.id.Bcolor) as TextView

        intent.putExtra("Rc", Redit.getText().toString())
        intent.putExtra("Gc", Gedit.getText().toString())
        intent.putExtra("Bc", Bedit.getText().toString())

        startActivity(intent)
    }
}
```

//Java

```
public class ResultActivity extends Activity {

    TextView tvView;
    LinearLayout lo;
    Color myRGB;
    int col;

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

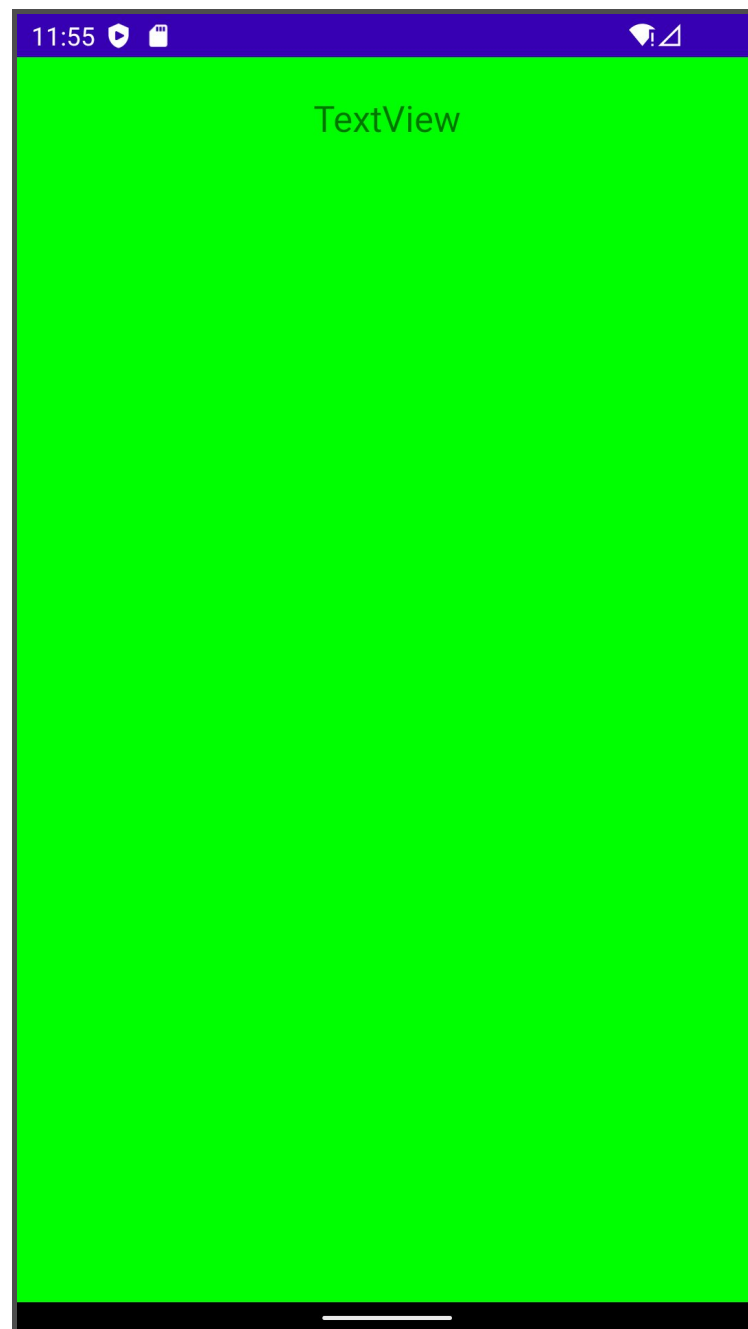
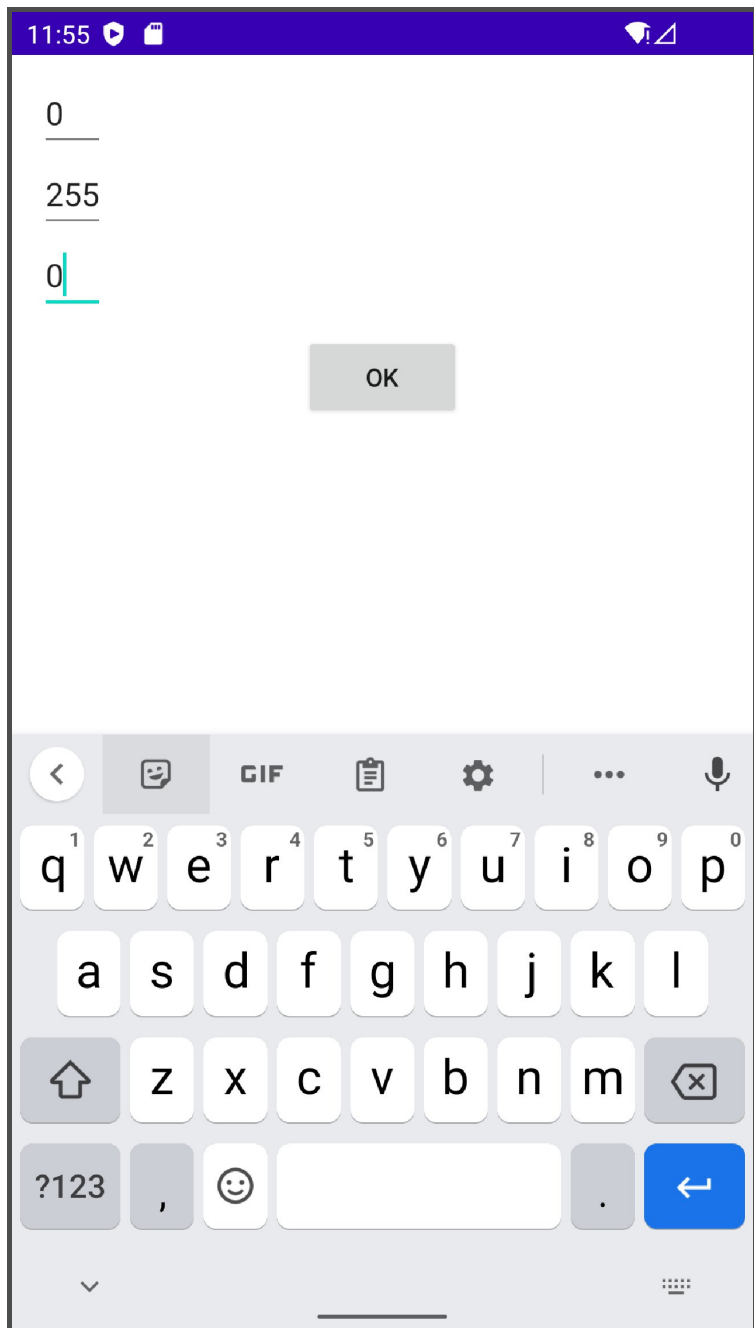
        Intent intent = getIntent();

        String rr = intent.getStringExtra("Rc");
        String gg = intent.getStringExtra("Gc");
        String bb = intent.getStringExtra("Bc");

        lo= (LinearLayout)findViewById(R.id.LResult);
        col = myRGB.rgb(Integer.parseInt(rr), Integer.parseInt(gg),
Integer.parseInt(bb));
        lo.setBackgroundColor(col);
    }
}
```


//Kotlin

```
class ResultActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.result)  
  
        var tvView = findViewById(R.id.tvView) as TextView  
  
        var rr = intent.getStringExtra("Rc")  
        var gg = intent.getStringExtra("Gc")  
        var bb = intent.getStringExtra("Bc")  
  
        var lo = findViewById(R.id.LResult) as LinearLayout  
        var col = Color.rgb(Integer.parseInt(rr),  
Integer.parseInt(gg), Integer.parseInt(bb))  
        lo.setBackgroundColor(col)  
    }  
}
```



Получение результата из Activity. Метод `startActivityForResult`

Бывает необходимость вызвать второстепенное Activity, выполнить на нем какое-либо действие и вернуть результат главному Activity. Например – при создании SMS. Вы жмете кнопку «добавить адресата», система показывает экран со списком из адресной книги, вы выбираете нужного вам абонента и возвращаетесь в экран создания SMS. Т.е. вы вызвали экран выбора абонента, а он вернул вашему экрану результат.

Создадим приложение с двумя экранами. С первого экрана будем вызывать второй экран, вводить данные, нажимать кнопку и возвращаться на первый экран с введенными данными. Например, будем таким образом запрашивать имя.

На экране находится `TextView`, который будет отображать имя, и кнопка, которая будет вызывать экран для ввода.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/btnName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:text="Input name" >
    </Button>

    <TextView
        android:id="@+id/tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Your name is: " >
    </TextView>

</LinearLayout>
```

//Java

```
public class MainActivity extends Activity implements OnClickListener {

    TextView tvName;
    Button btnName;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvName = (TextView) findViewById(R.id.tvName);
        btnName = (Button) findViewById(R.id.btnName);
        btnName.setOnClickListener(this);

    }

    @Override
    public void onClick(View v) {
        Intent intent = new Intent(this, InputName.class);
        startActivityForResult(intent, 1);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (data == null) {return;}
        String name = data.getStringExtra("name");
        tvName.setText("Your name is " + name);
    }
}
```

//Kotlin

```
class MainActivity : AppCompatActivity() {

    /** Called when the activity is first created. */
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btnName.setOnClickListener(this::onClick)
    }

    fun onClick(view : View){
        var intent = Intent(this@MainActivity, InputName::class.java)
        startActivityForResult(intent, 1)
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int,
data: Intent?) {
        if (data == null) {return}
        var name = data.getStringExtra("name")
        var tvName = findViewById(R.id.tvName) as TextView
        tvName.setText("Your name is " + name)
    }
}
```

Определяем `TextView` и кнопку, настраиваем обработчик. В методе обработчика `onClick` создаем `Intent`, указываем класс второго `Activity`. Для отправки используем `startActivityForResult`. Отличие от обычного `startActivity` в том, что `MainActivity` становится «родителем» для `InputName`. И когда `InputName` закрывается, вызывается метод `onActivityResult` в `MainActivity`, тем самым определяя то, что закрылось `Activity`, которое мы вызывали методом `startActivityForResult` в данном случае `InputName`.

В `startActivityForResult` в качестве параметров передается `Intent` и `requestCode`. `requestCode` – необходим для идентификации. Далее он будет рассмотрен подробнее.

В `onActivityResult` имеются следующие параметры:

`requestCode` – тот же идентификатор, что и в `startActivityForResult`. По нему определяется, с какого `Activity` пришел результат.

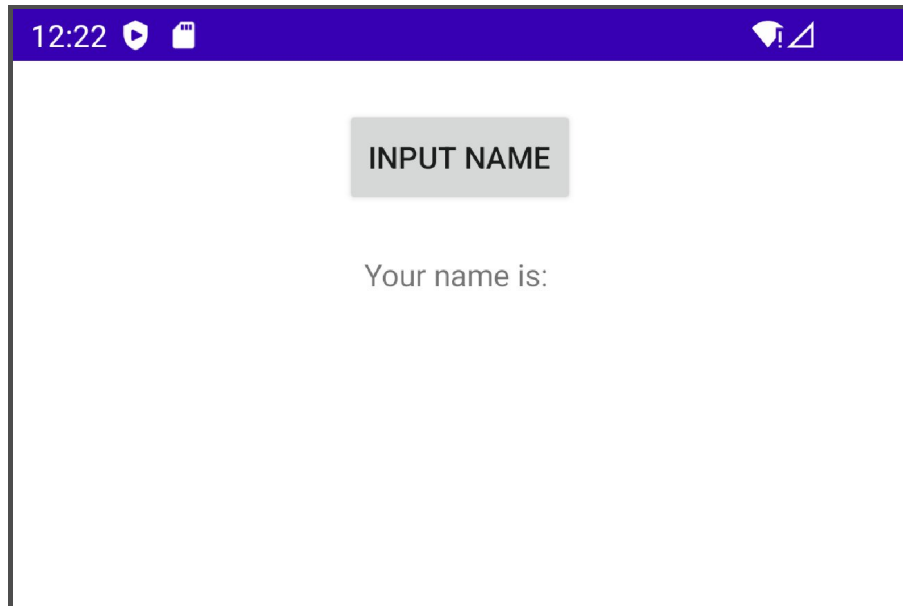
`resultCode` – код возврата. Определяет успешно прошел вызов или нет.

`data` – `Intent`, в котором возвращаются данные.

//Kotlin

```
class InputName : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_name)  
        btnOK.setOnClickListener(this::onClick)  
    }  
  
    fun onClick(view: View) {  
        var etName = findViewById(R.id.etName) as EditText  
        intent.putExtra("name", etName.text.toString())  
        setResult(RESULT_OK, intent)  
        this.finish()  
    }  
}
```


В методе `onClick` создается `Intent` и в него помещаются данные из поля ввода под именем `name`. Здесь важно, что данный `Intent` никак не адресуется, т.е. ни класс, ни `action` не указывается. Поэтому непонятно кому предназначен этот `Intent`. В данном случае метод `setResult` знает, куда его адресовать - в «родительское» `Activity`, в котором был вызван метод `startActivityForResult`. Также в `setResult` передается константа `RESULT_OK`, означающую успешное завершение вызова. И именно она передается в параметр `resultCode` метода `onActivityResult` в `MainActivity.java`. Далее методом `finish` завершается работа `InputName`, чтобы результат был передан в `MainActivity`.



Атрибут data Intent объекта. Uri. Вызов системных приложений

Известно, что Intent имеет атрибут action. С помощью этого атрибута обычно дается указание на выполнение какого-либо действия. Например, просмотр списка контактов или редактирование изображения. Но действие обычно совершается не просто так, а с чем-либо. Значит кроме указания действия, необходимо указывать на объект, с которым эти действия нужно произвести. Для этого Intent имеет атрибут data.

Один из способов присвоения значения этому атрибуту – метод setData (Uri data) у объекта Intent. На вход этому методу подается объект Uri.

Uri – это объект, который берет строку, разбирает ее на составляющие и хранит в себе эту информацию. Строка, должна быть не любая, а составлена в соответствии с документом RFC 2396. Uri имеет несколько методов, которые позволяют извлекать из обработанной строки отдельные элементы.

Рассмотрим такую строку - **http адрес:**

```
Uri uri = Uri.parse("http://developer.android.com/reference/android/net/Uri.html");
```

Смотрим, чего нам возвращают методы:

```
uri.getScheme(): http
```

```
uri.getSchemeSpecificPart(): //developer.android.com/reference/android/net/Uri.html
```

```
uri.getAuthority(): developer.android.com
```

```
uri.getHost(): developer.android.com
```

```
uri.getPath(): /reference/android/net/Uri.html
```

```
uri.getLastPathSegment(): Uri.html
```

Понятия Scheme, Authority, Host, Path и пр. имеются в RFC документа. Там можно найти их полное описание, понять что они означают и свериться с тем, что вернули методы Uri.

FTP

```
Uri uri = Uri.parse("ftp:// bob@google.com:80/data/files");
```

```
uri.getScheme(): ftp
```

```
uri.getSchemeSpecificPart(): // bob@google.com:80/data/files
```

```
uri.getAuthority(): bob@google.com:80
```

```
uri.getHost(): google.com
```

```
uri.getPort(): 80
```

```
uri.getPath(): /data/files
```

```
uri.getLastPathSegment(): files
```

```
uri.getUserInfo(): bob
```

Координаты

```
Uri uri = Uri.parse("geo:55.754283,37.62002");
```

```
uri.getScheme(): geo
```

```
uri.getSchemeSpecificPart(): 55.754283,37.62002
```

Здесь уже получилось выделить только Scheme и SchemeSpecificPart.

Номер телефона

```
Uri uri = Uri.parse("tel:12345");
```

```
uri.getScheme(): tel
```

```
uri.getSchemeSpecificPart():12345
```

Аналогично, получилось выделить только две части из строки.

Контакт из адресной книги

```
Uri uri = Uri.parse("content://contacts/people/1");
```

```
uri.getScheme(): content
```

```
uri.getSchemeSpecificPart(): //contacts/people/1
```

```
uri.getAuthority(): contacts
```

```
uri.getPath(): /people/1
```

```
uri.getLastPathSegment(): 1
```

В этом примере Scheme равен content. Это особый тип данных – Content Provider. Он позволяет любой программе давать доступ к своим данным, а другим программам – читать и менять эти данные.

Примеры показывают, что Uri можно создать из абсолютно разных строк: http-адрес, ftp-адрес, координаты, номер телефона, контакт из адресной книги.

Тип содержимого можно определить по Scheme. И этот же Scheme можно настроить в Intent Filter и отсеивать Intent, только с нужным нам типом данных в Uri, например только http. Рассмотрим простой пример.

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <Button
        android:id="@+id/btnWeb"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="Web" >
    </Button>

    <Button
        android:id="@+id/btnMap"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="Map" >
    </Button>

    <Button
        android:id="@+id/btnCall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="Call" >
    </Button>

</LinearLayout>
```

//Java

```
public class MainActivity extends Activity implements OnClickListener {
    Button btnWeb;
    Button btnMap;
    Button btnCall;

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

        btnWeb = (Button) findViewById(R.id.btnWeb);
        btnMap = (Button) findViewById(R.id.btnMap);
        btnCall = (Button) findViewById(R.id.btnCall);

        btnWeb.setOnClickListener(this);
        btnMap.setOnClickListener(this);
        btnCall.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent intent;
        switch (v.getId()) {
            case R.id.btnWeb:
                intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://developer.android.com"));
                startActivity(intent);
                break;
            case R.id.btnMap:
                intent = new Intent();
                intent.setAction(Intent.ACTION_VIEW);
                intent.setData(Uri.parse("geo:55.76583334,37.68472223"));
                startActivity(intent);
                break;
            case R.id.btnCall:
                intent = new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:12345"));
                startActivity(intent);
                break;
        }
    }
}
```

//Kotlin

```
class MainActivity : AppCompatActivity() {

    /** Called when the activity is first created. */
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnWeb.setOnClickListener(this::onClick)
        btnMap.setOnClickListener(this::onClick)
        btnCall.setOnClickListener(this::onClick)
    }

    fun onClick(view : View){
        var intent : Intent
        when (view.id){
            R.id.btnWeb -> {
                intent = Intent(Intent.ACTION_VIEW,
Uri.parse("http://developer.android.com"))
                startActivity(intent)
            }
            R.id.btnMap -> {
                intent = Intent()
                intent.setAction(Intent.ACTION_VIEW)
                intent.setData(Uri.parse("geo:55.76583334,37.68472223"))
                startActivity(intent)
            }
            R.id.btnCall -> {
                intent = Intent(Intent.ACTION_DIAL)
                intent.setData(Uri.parse("tel:12345"))
                startActivity(intent)
            }
        }
    }
}
```


В случае btnWeb использовался конструктор Intent (String action, Uri uri). Он создает Intent и на вход сразу принимает action и data. Мы используем стандартный системный action – ACTION_VIEW. Это константа в классе Intent – означает просмотр чего-либо. В качестве data подается объект Uri, созданный из веб-ссылки: <http://developer.android.com>.

В случае btnMap использовался конструктор Intent(). Он просто создает Intent. А в следующих строках происходит присвоение ему атрибутов action и data. action – снова ACTION_VIEW, а в качестве data создается Uri из пары координат - 55.754283,37.62002. Этот Intent означает, что мы хотим посмотреть на карте указанные координаты.

В случае btnCall используем конструктор Intent (String action). На вход ему сразу подается action, а data указывается позже. action в данном случае – ACTION_DIAL – открывает приложение телефон и набирает номер, указанный в data, но не начинает звонок. В data – помещаем Uri, созданный из номера телефона 12345.

Также в файле манифеста приложения, на вкладке Permission добавьте элемент Uses Permission и справа в поле Name выберите android.permission.INTERNET. Это даст приложению доступ в интернет.

