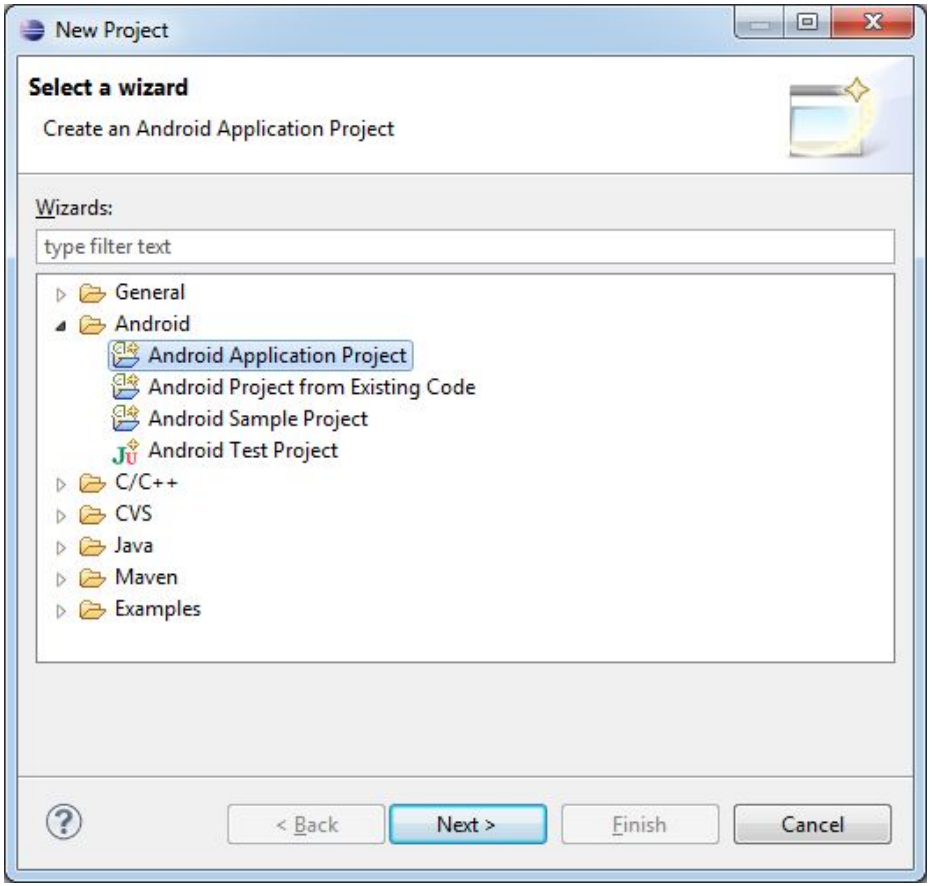
Разработка приложений для ОС Android

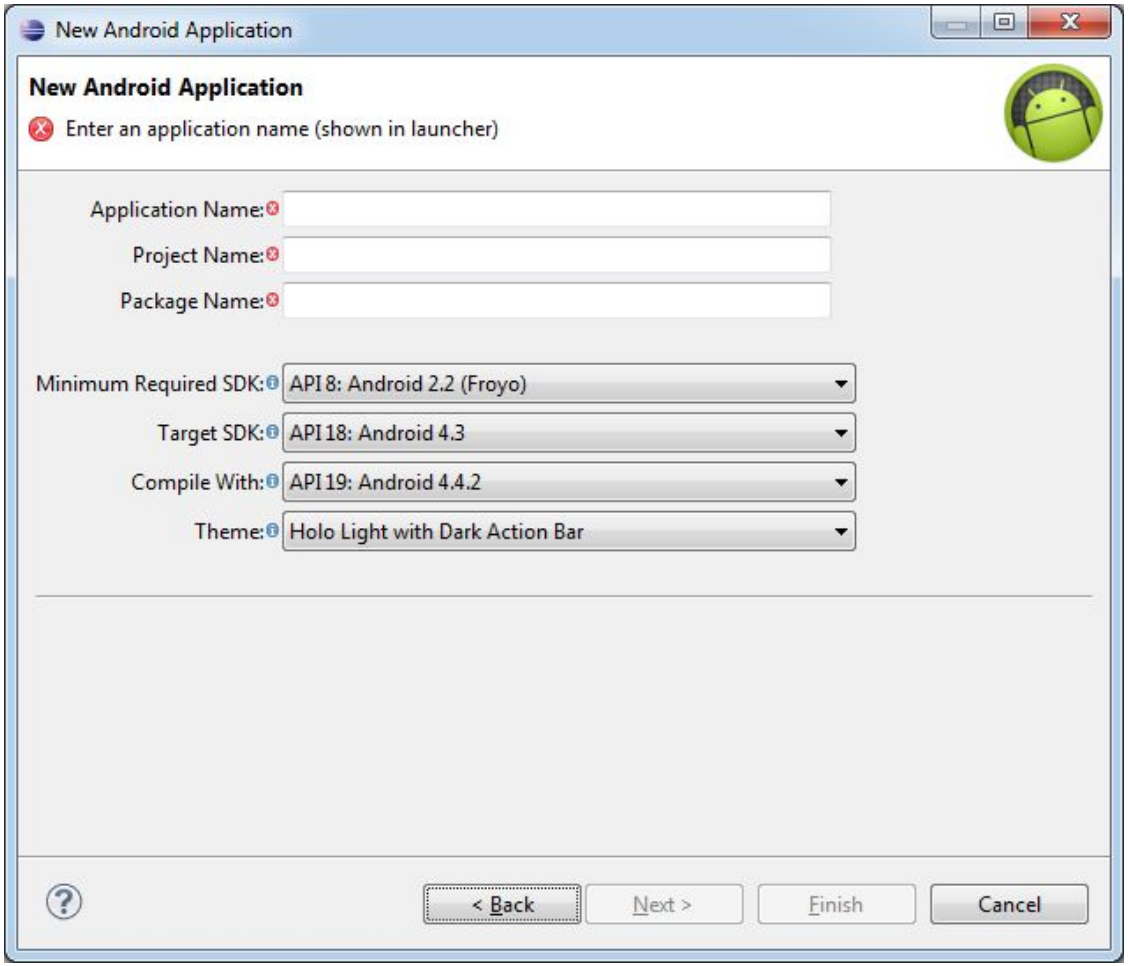
Создание первого приложения

File -> New -> Project



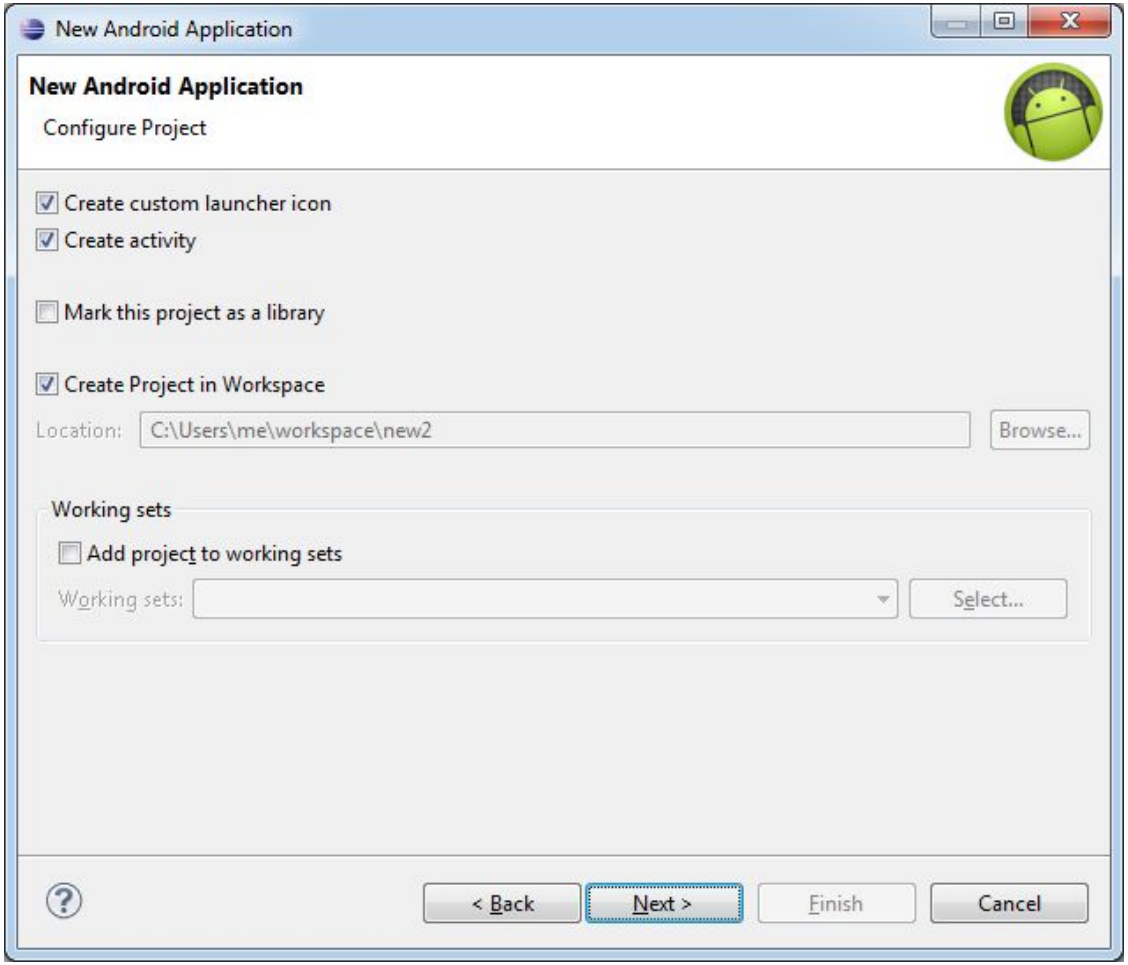
Разработка приложений для ОС Android

Создание первого приложения



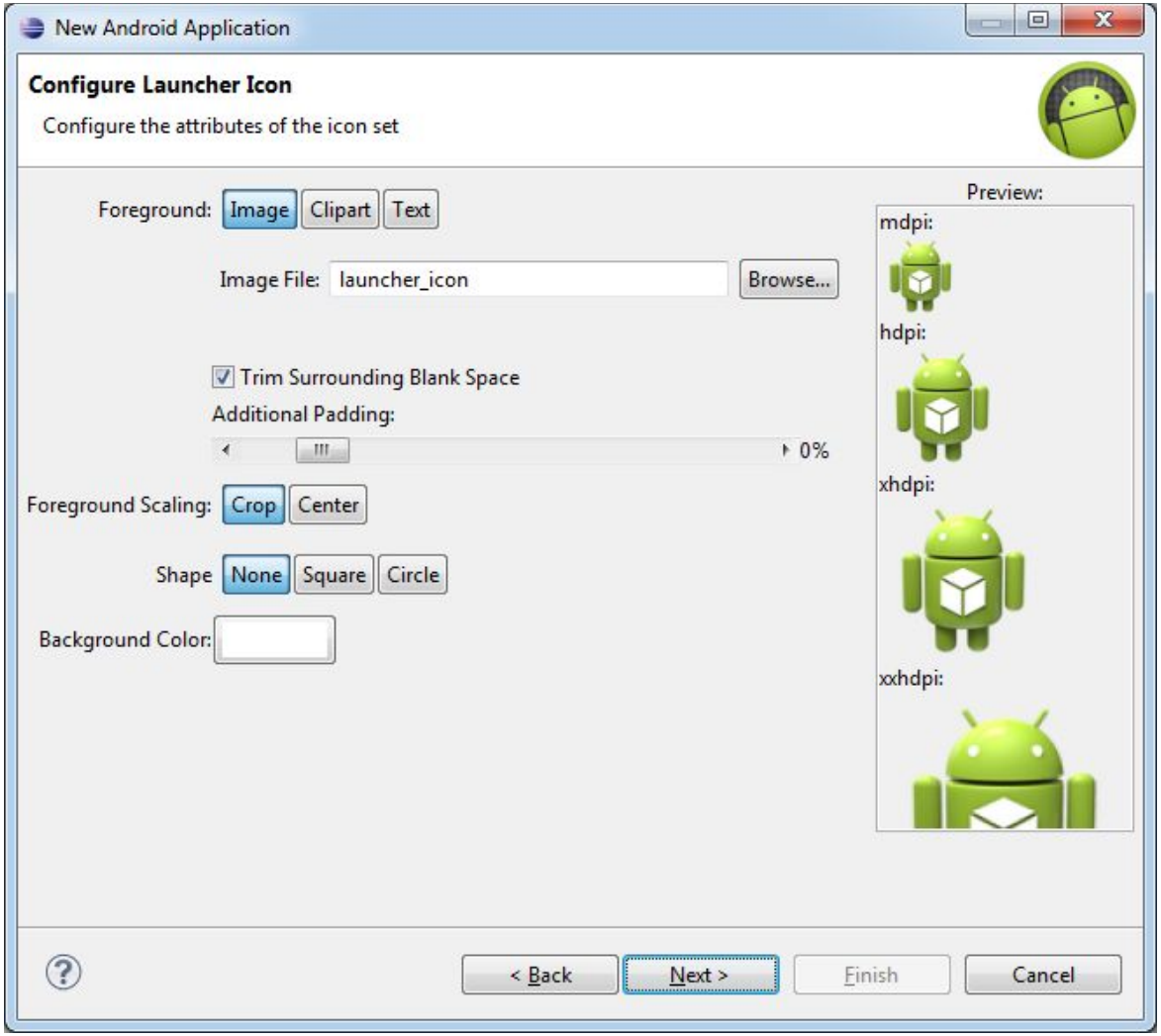
Разработка приложений для ОС Android

Создание первого приложения



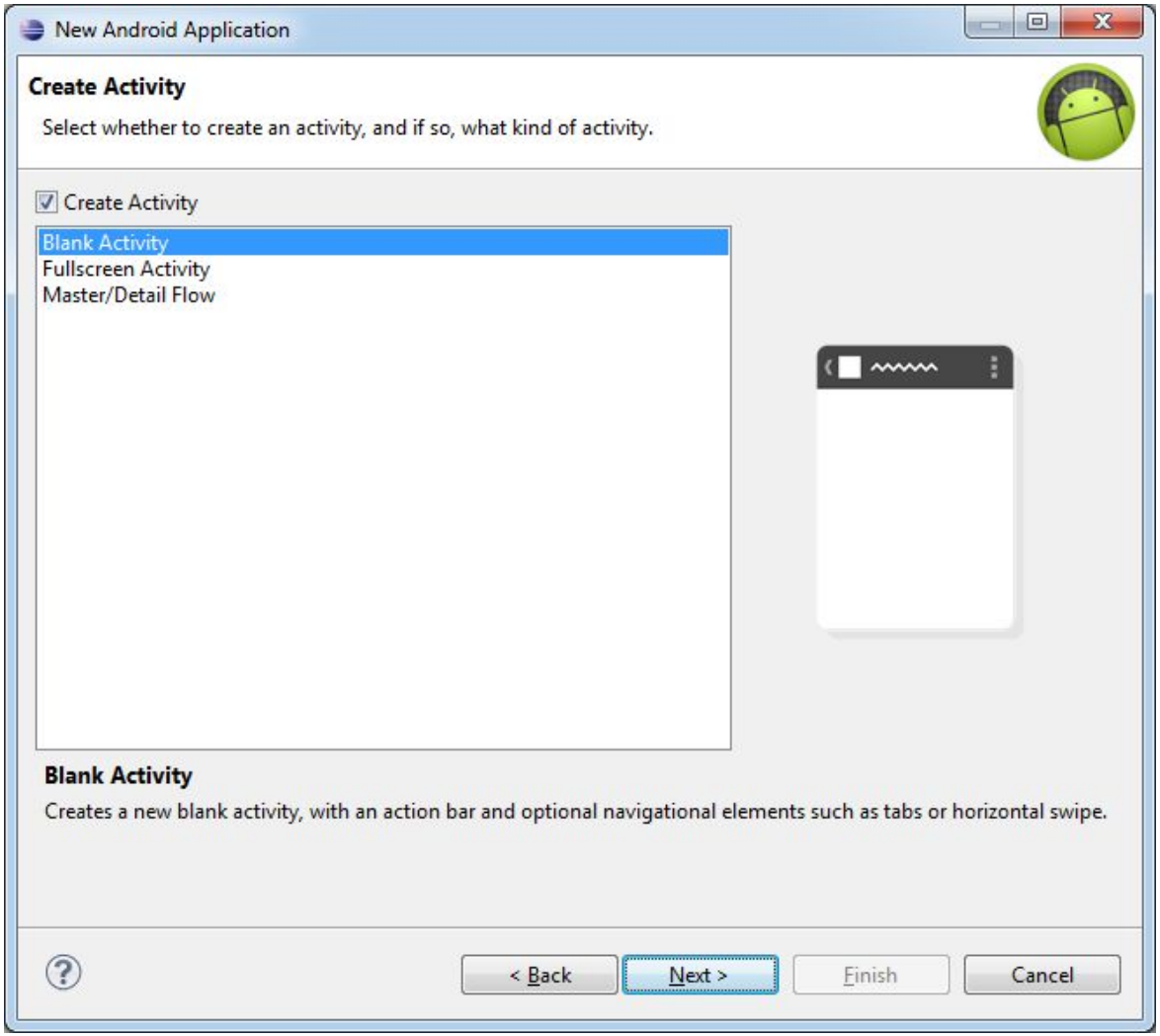
Разработка приложений для ОС Android

Создание первого приложения



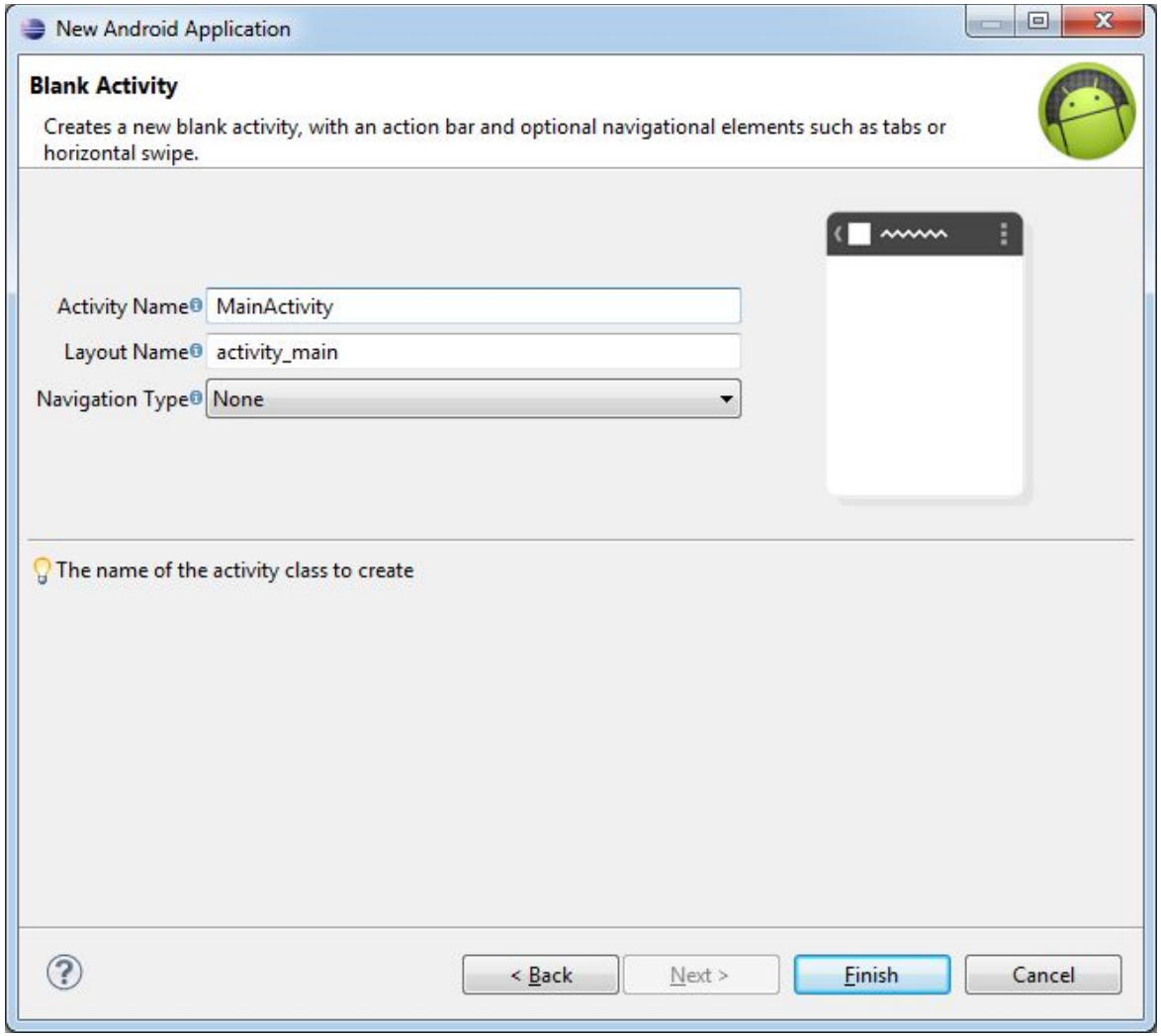
Разработка приложений для ОС Android

Создание первого приложения



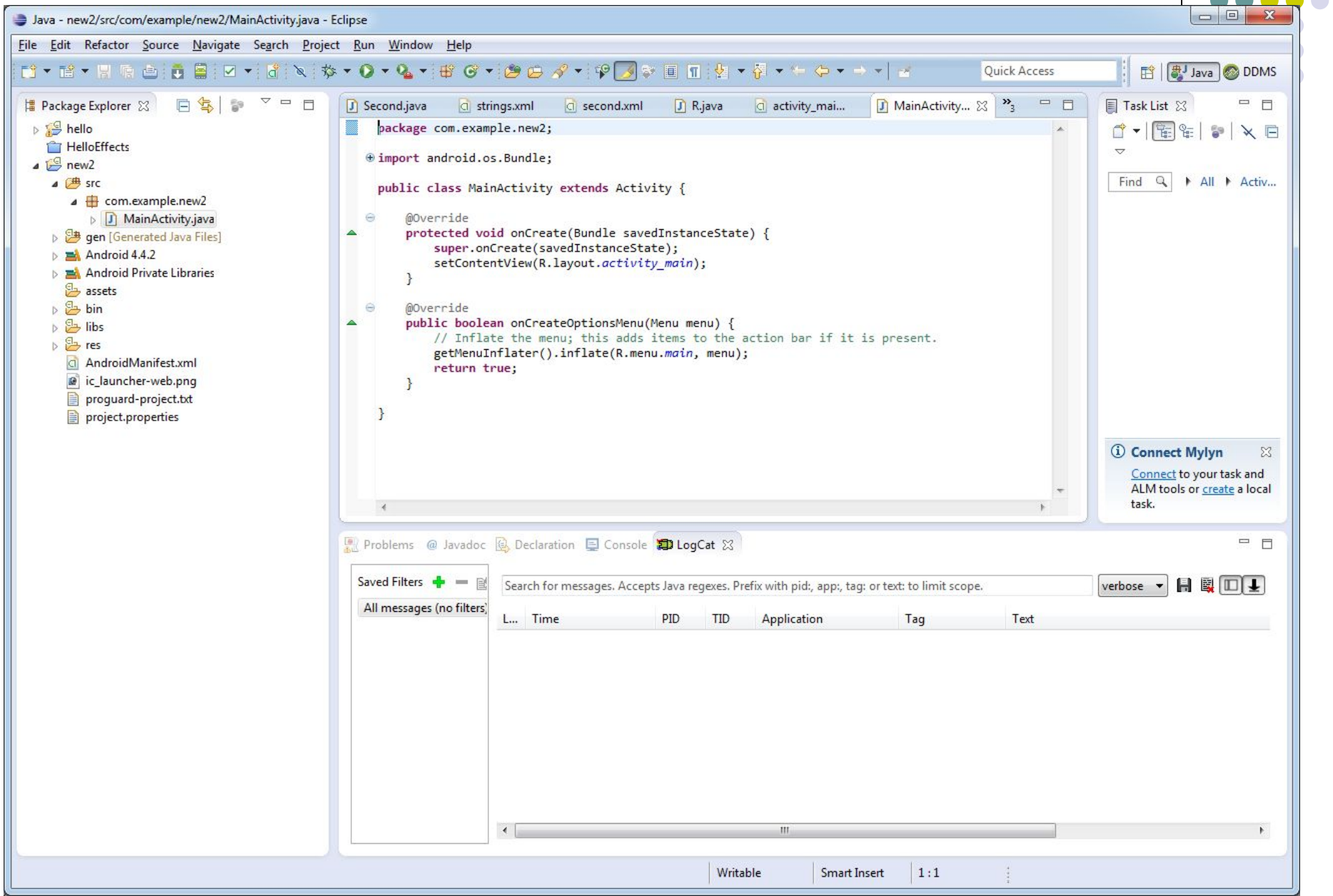
Разработка приложений для ОС Android

Создание первого приложения



Разработка приложений для ОС Android

Создание первого приложения



Разработка приложений для ОС Android

Создание первого приложения

Основные файлы приложения – это:

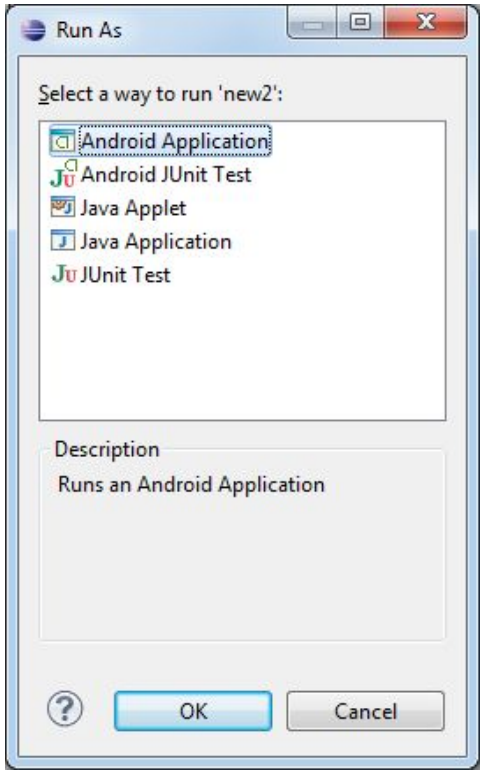
src/com.example.название_приложения/MainActivity.java – файл, содержащий код основного Activity (состояния) нашего приложения;

res/layout/activity_main.xml – файл, содержащий интерфейс приложения;

AndroidManifest.xml – файл манифеста, содержащий всю информацию о приложении и его состояниях, сообщающий её системе eclipse и android.market.

Папки res/drawable-xxxx содержат варианты интерфейса для различных разрешений экрана.

Чтобы запустить приложение, нужно нажать кнопку Run в панели инструментов, при этом выбрать в открывшемся окне Android application:



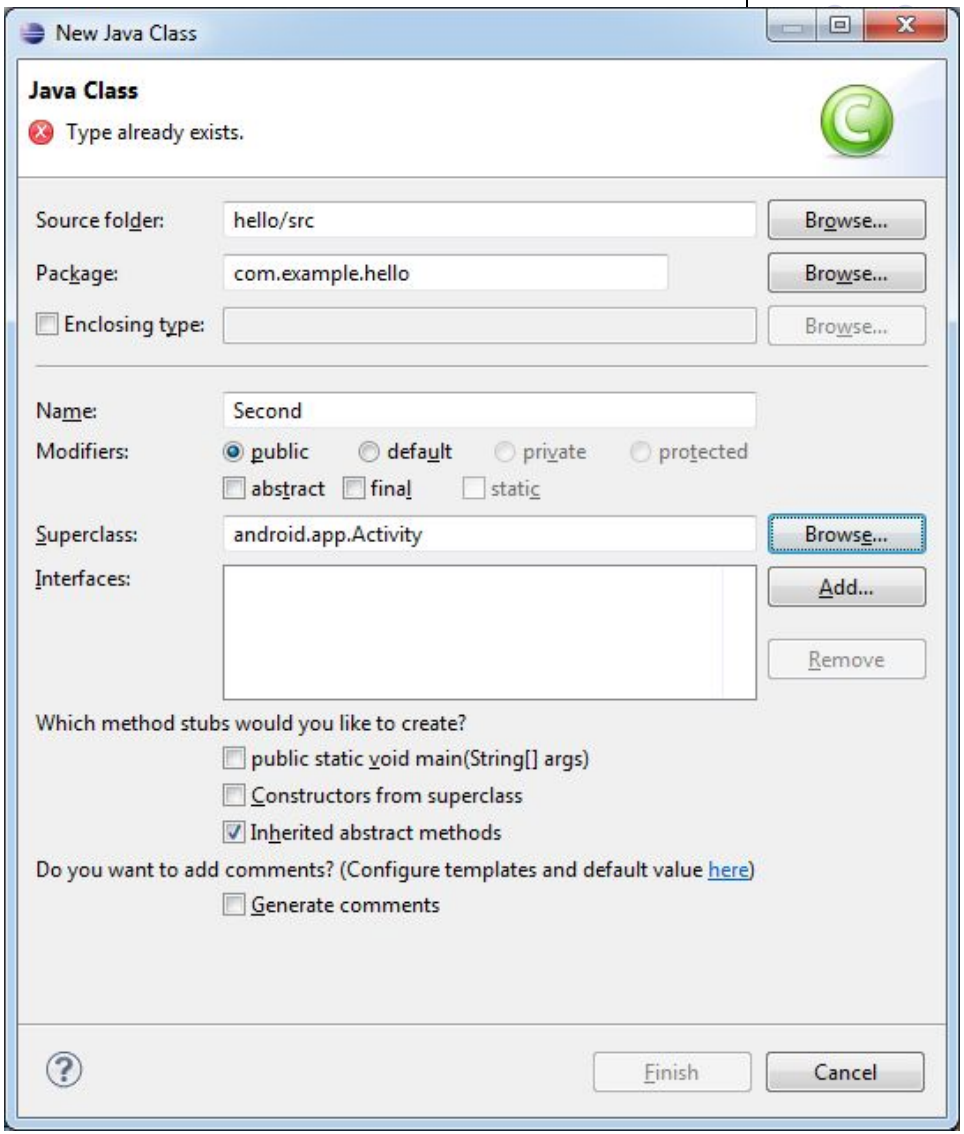
Разработка приложений для ОС Android

Создание первого приложения

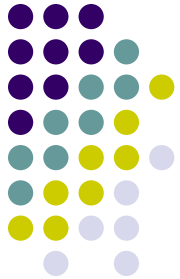
Создадим новое Activity.

Правой кнопкой по com.example. название_приложения, -> New ->Class

После нажатия кнопки Finish появляется второй java файл – Second.java, содержащий логику работы второго окна нашего приложения.



Разработка приложений для ОС Android



Создание первого приложения

Однако переключить мы пока на второе состояние (окно, activity) наше приложение не можем, ибо нечем.

Поэтому мы возвращаемся в xml файл mainactivity.xml и добавляем кнопку. При этом мы видим воскл. Знак, означающий, что название кнопки неплохо бы вынести в файл ресурсов, что мы и делаем.

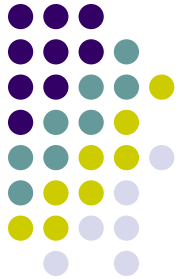
Переходим в файл values/string.xml, добавляем новую надпись (Add-New-String) с именем button1name и значением NEXT.

Затем возвращаемся в файл mainactivity.xml, находим свойство Text нашей кнопки и изменяем его на @string/button1name.

А зачем это всё надо?

Чтобы хранить и изменять при необходимости все текстовые данные приложения в одном файле. Это удобно, и компиляция приложения не требуется.

Разработка приложений для ОС Android



Создание первого приложения

Кнопка пока не работает, поэтому мы сохраняем (ctrl+S) изменения и переходим в файл mainactivity.java.

Там внутри метода onCreate определяем новую кнопку:

```
Button but1 = (Button) findViewById(R.id.button1);
```

И задаём новой переменной listener на событие onclick, периодически нажимая комбинацию ctrl+space:

```
but1.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
}
});
```

Далее определим понятие Intent, чтобы запускалось второе activity:

```
// TODO Auto-generated method stub

Intent intent1 = new Intent(MainActivity.this, Second.class);
startActivity(intent1);
```

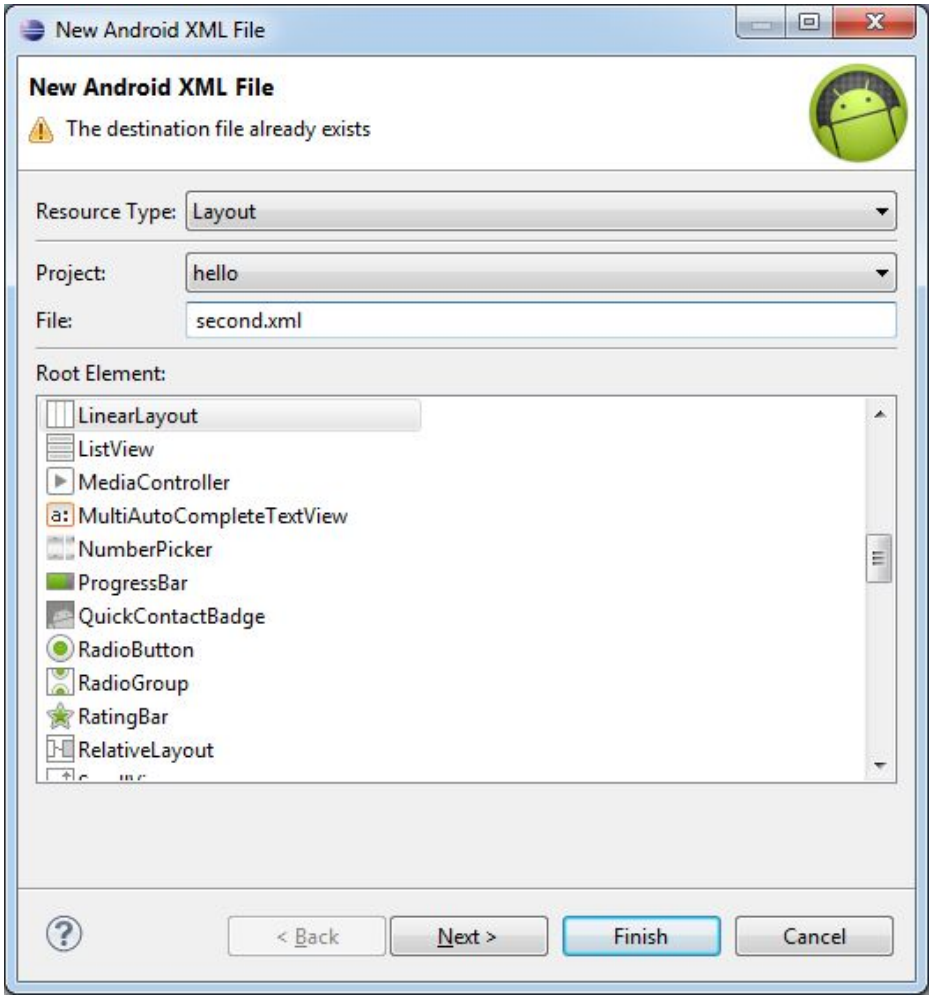
Разработка приложений для ОС Android

Создание первого приложения

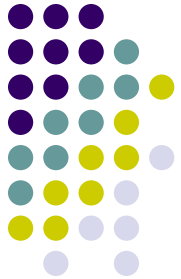
Если сейчас выполнить приложение, то возникнет ошибка.

Ошибка связана с тем, что:

1. Мы добавили в наше приложение новое состояние (activity), но не добавили его в файл манифеста приложения. Добавим его после первого activity: <activity android:name="com.example.hello.Second" />
2. Теперь ошибка исчезнет, но мы ничего не увидим, т.к. у нас нет интерфейса для второго activity – мы его просто не создали. Чтобы создать его, нажмём кнопку на панели инструментов New android xml file. Имя должно состоять из маленьких букв и не содержать спецсимволов.



Разработка приложений для ОС Android



Создание первого приложения

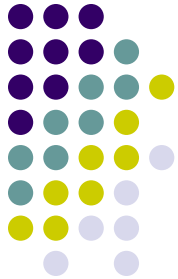
Добавим текст на экран, чтобы отличать состояния приложения. Перетаскиваем объект TextView и делаем для него те же манипуляции, что были и со свойством Text кнопки (name=text2, value=this is the 2nd page, ctrl+S. @string/text2, ctrl+S).

Далее, нам надо зайти в Second.java и создать метод OnCreate для этого activity нашего приложения (ctrl+space).

```
protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.second);
}
```

Запустив приложение после сохранения этого файла мы увидим, что наша кнопка действительно переключает интерфейсы из первого состояния во второе.

Разработка приложений для ОС Android



Создание первого приложения

Теперь попробуем передать что-то с первой страницы (состояния) на вторую. Для этого откроем mainactivity.xml и добавим в интерфейс текстовое поле TextField.

Система попросит нас описать доступные виды ввода для этого текстового поля, поэтому далее мы зайдём в код этого xml файла, найдём текстовое поле EditText и добавим туда строчку

```
android:inputType="text"
```

Ctrl+S, и переходим в mainactivity.java, где создаём переменную для текстового окна:

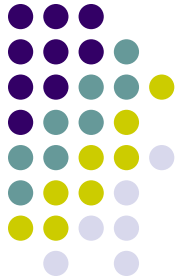
```
final EditText edit1=(EditText) findViewById(R.id.editText1);
```

Теперь нужно создать метод, передающий данные из первого состояния на второе. Для этого добавим экстраданные, куда надо (после объявления переменной intent1) с ключом **text**:

```
intent1.putExtra("text",edit1.getText().toString());
```

Ctrl+S и открываем второй java файл.

Разработка приложений для ОС Android



Создание первого приложения

В файле Second.java добавляем переменную TextView, ссылающуюся на надпись, и присваиваем ей значение, которое получаем по ключу text из первого состояния приложения:

```
TextView text2=(TextView) findViewById(R.id.textView1);
text2.setText(getIntent().getExtras().getString("text"));
```

Сохраняем, запускаем и видим, что всё работает.

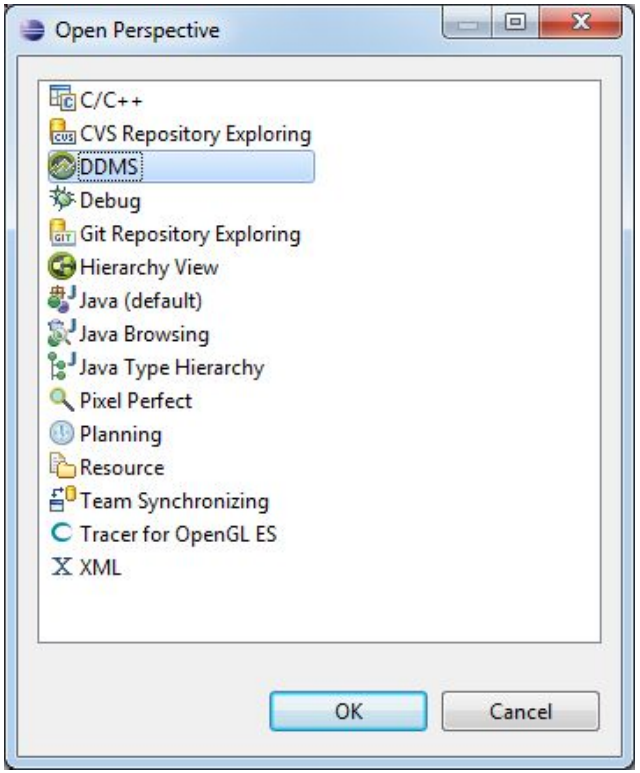
Далее мы изменим приложение так, чтобы оно могло получать данные из другого приложения.

Для этого в правом верхнем углу найдём кнопку open perspective. В открывшемся окне выберем DDMS и нажмём ок.

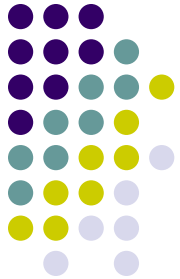
Разработка приложений для ОС Android

Создание первого приложения

Для этого в правом верхнем углу найдём кнопку open perspective. В открывшемся окне выберем DDMS и нажмём ок.



Разработка приложений для ОС Android



Создание первого приложения

Далее запустим телефон, откроем вкладку file explorer справа, и увидим все файлы, которые есть на нашем виртуальном телефоне.
Зайдём в папку mnt/sdcard/LOST.DIR и скопируем туда пару первых попавшихся картинок путем drag n drop.

После этого можно проверить, появились ли скопированные картинки на телефоне. Для этого надо закрыть телефон и запустить его с нуля, т.е. не из Snapshot.

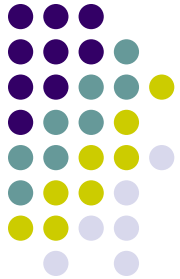
После этого можно зайти в галерею и убедиться, что картинки там есть.

Далее можно расшарить картинку, однако сделать это можно только по sms. Сделаем так, чтобы эту картинку мы могли передать в наше приложение.

Для этого нужно использовать внешний Intent – т.е. действие от внешнего, другого приложения, которое надо обработать в нашем приложении.

Разработка приложений для ОС Android

Создание первого приложения



Начать надо с того, что добавить новый Intent фильтр в наш файл манифеста:

```
<intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="image/*"/>
</intent-filter>
```

Далее мы идём в mainactivity.xml файл, и добавляем в интерфейс компонент ImageView для отображения пересылаемой из галереи картинки.

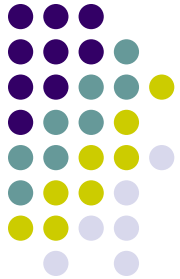
После добавления нужно откорректировать xml код, для этого переключаемся на него, находим элемент ImageView и добавляем строчку

```
android:contentDescription="@+id/imageView1”
```

И сохраняем.

Разработка приложений для ОС Android

Создание первого приложения



Далее переходим в mainactivity.java файл и вносим следующие изменения:

```
ImageView image1=(ImageView) findViewById(R.id.imageView1);
image1.setImageURI((Uri) getIntent().getExtras().get(Intent.EXTRA_STREAM));
```

Если после этого запустить приложение, то будет выдано сообщение об ошибке, т.к. не существует ещё никакой посланной из галереи картинки и объекту ImageView отображать, мягко говоря, нечего.

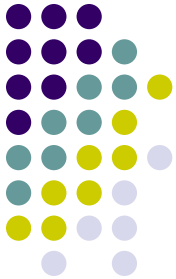
Однако мы можем проверить наше приложение, открыв галерею и расшарив одну из картинок.

Чтобы приложение запускалось без галереи, сделаем некоторую модификацию кода mainactivity.java:

```
        Intent intent=getIntent();
        String action=intent.getAction();
        if (intent.ACTION_SEND.equals(action)){
        image1.setImageURI((Uri)
getIntent().getExtras().get(Intent.EXTRA_STREAM));
        }
```

Разработка приложений для ОС Android

Создание первого приложения



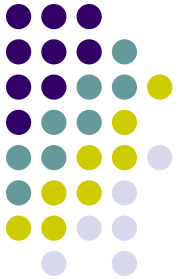
Теперь при запуске приложения нет ошибки, но нет и картинки. Картинка появляется только в том случае, если через внешний Intent галерея или любое другое приложение с картинками пытается расшарить одну из своих картинок, тогда можно выбрать наше приложение и возникает этот внешний интент SEND, который мы слушаем и обрабатываем в нашем приложении.

Лабы:

Цель:
Познакомиться с инструментами разработки Android-приложений.
На примере простейших программ разобрать структуру типичного Android-приложения.
Научиться запускать приложение на эмуляторе.
Научиться тестировать приложение с помощью Dalvik Debug Monitor Server (DDMS).
Задачи:
Создать эмулятор своего телефона.
Разобрать следующие приложения, запустить их на эмуляторе и выполнить для каждого примера дополнительное задание:
Hello, World!
Работа с кнопками
Работа с анимацией
Работа с GPS

Разработка приложений для ОС Android

Лабы:



Данное приложение содержит тексты программ, рассматриваемых в примерах.

Button Example

res/layout/activity_main.xml

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.application" android:versionCode="1" android:versionName="1.0" > <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="15" /> <application android:icon="@drawable/ic_launcher" android:label="@string/app_name"> <activity android:name=".MainActivity" android:label="@string/title_activity_main" > <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </application> </manifest>

src/MainActivity.java

package com.example.application; import android.app.Activity; import android.graphics.Color; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.widget.Button; import android.widget.LinearLayout; import android.widget.Toast; public class MainActivity extends Activity implements OnClickListener { private Button switchToGreen; private Button switchToRed; private Button switchToBlue; private LinearLayout screenLayout; private Toast informationToast; @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); // init buttons switchToBlue = (Button) findViewById(R.id.switchBlue); switchToGreen = (Button) findViewById(R.id.switchGreen); switchToRed = (Button) findViewById(R.id.switchRed); screenLayout = (LinearLayout) findViewById(R.id.screenLayout); // setup listeners switchToBlue.setOnClickListener(this); switchToRed.setOnClickListener(this); switchToGreen.setOnClickListener(this); informationToast = Toast.makeText(this, "", Toast.LENGTH_SHORT); } public void onClick(View view) { if (switchToBlue.equals(view)) { screenLayout.setBackgroundColor(Color.BLUE); showToast("Hello blue"); } else if (switchToRed.equals(view)) { screenLayout.setBackgroundColor(Color.RED); showToast("Hello red"); } else if (switchToGreen.equals(view)) { screenLayout.setBackgroundColor(Color.GREEN); showToast("Hello green"); } } private void showToast(String text) { informationToast.cancel(); informationToast.setText(text); informationToast.show(); } }

Разработка приложений для ОС Android

Лабы:

Данное приложение содержит тексты программ, рассматриваемых в примерах.

Animation Example

res/anim/frame_anim.xml

<?xml version="1.0" encoding="utf-8"?> <animation-list xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="false" > <item android:drawable="@drawable/ic_launcher" android:duration="200"/> <item android:drawable="@drawable/ic_launcher1" android:duration="200"/> <item android:drawable="@drawable/ic_launcher2" android:duration="200"/> <item android:drawable="@drawable/ic_launcher3" android:duration="200"/> </animation-list>

res/anim/transform_anim.xml

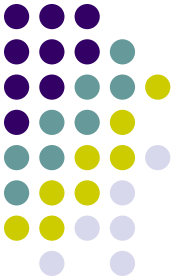
<?xml version="1.0" encoding="utf-8"?> <set xmlns:android="http://schemas.android.com/apk/res/android" android:shareInterpolator="false" > <scale android:duration="700" android:fillAfter="false" android:fromXScale="1.0" android:fromYScale="1.0" android:interpolator="@android:anim/accelerate_decelerate_interpolator" android:pivotX="50%" android:pivotY="50%" android:toXScale="1.4" android:toYScale="0.6" /> <set android:interpolator="@android:anim/decelerate_interpolator" > <scale android:duration="400" android:fillBefore="false" android:fromXScale="1.4" android:fromYScale="0.6" android:pivotX="50%" android:pivotY="50%" android:startOffset="700" android:toXScale="0.0" android:toYScale="0.0" /> <rotate android:duration="400" android:fromDegrees="0" android:pivotX="50%" android:pivotY="50%" android:startOffset="700" android:toDegrees="-45" android:toYScale="0.0" /> </set> </set>

src/MainActivity.java

package com.example.application; import android.app.Activity; import android.graphics.Color; import android.graphics.drawable.AnimationDrawable; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.view.animation.Animation; import android.view.animation.AnimationUtils; import android.widget.Button; import android.widget.ImageView; public class MainActivity extends Activity implements OnClickListener { private Button startFrameAnim; private Button startTransformAnim; private Button cancelAnim; private ImageView animationView; @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); startFrameAnim = (Button) findViewById(R.id.frameAnimationStart); startTransformAnim= (Button) findViewById(R.id.transformAnimationStart); cancelAnim = (Button) findViewById(R.id.cancelAnimation); animationView = (ImageView) findViewById(R.id.animationView); startFrameAnim.setOnClickListener(this); startTransformAnim.setOnClickListener(this); cancelAnim.setOnClickListener(this); } public void onClick(View v) { if (startFrameAnim.equals(v)) { animationView.setBackgroundResource(R.anim.frame_anim); AnimationDrawable animation = (AnimationDrawable) animationView.getBackground(); animation.start(); } else if (startTransformAnim.equals(v)) { animationView.setBackgroundResource(R.drawable.ic_launcher); Animation transformAnimation = AnimationUtils.loadAnimation(this, R.anim.transform_anim); animationView.startAnimation(transformAnimation); } else if (cancelAnim.equals(v)) { animationView.setBackgroundColor(Color.BLACK); } } }

Разработка приложений для ОС Android

Лабы:



Данное приложение содержит тексты программ, рассматриваемых в примерах.

Location Example

src/MainActivity.java

```
package com.example.application; import java.util.Date; import android.app.Activity; import
android.location.Criteria; import android.location.Location; import android.location.LocationListener;
import android.location.LocationManager; import android.os.Bundle; import android.widget.TextView;
public class MainActivity extends Activity implements LocationListener { private TextView latitudeLabel;
private TextView longitudeLabel; private TextView statusLabel; private LocationManager
locationManager; @Override public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); latitudeLabel = (TextView)
findViewById(R.id.latitudeLabel); longitudeLabel = (TextView) findViewById(R.id.longitudeLabel);
statusLabel = (TextView) findViewById(R.id.statusLabel); locationManager = (LocationManager)
getSystemService(Activity.LOCATION_SERVICE); } @Override protected void onResume() {
super.onResume(); // construct a criteria with best accuracy Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE); // get best ENABLED provider that meets the criteria
String provider = locationManager.getBestProvider(criteria, true); // request the updates
locationManager.requestLocationUpdates(provider, 0, 0, this); } @Override protected void onPause() {
super.onPause(); locationManager.removeUpdates(this); } public void onLocationChanged(Location
location) { statusLabel.setText("Location recieved at " + new Date()); latitudeLabel.setText("Latitude: " +
location.getLatitude()); longitudeLabel.setText("Longitude: " + location.getLongitude()); } public void
onProviderDisabled(String provider) { } public void onProviderEnabled(String provider) { } public void
onStatusChanged(String provider, int status, Bundle extras) { } }
```

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.application"
android:versionCode="1" android:versionName="1.0" > <uses-sdk android:minSdkVersion="8"
android:targetSdkVersion="15" /> <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/> <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/> <application
android:icon="@drawable/ic_launcher" android:label="@string/app_name"> <activity
android:name=".MainActivity" android:label="@string/title_activity_main" > <intent-filter> <action
android:name="android.intent.action.MAIN" /> <category
android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </application>
</manifest>
```