

دوشنبه، ۱۱ اردیبهشت ۱۳۹۱، ۰۳:۰۴ ب.ظ

استفاده از مفاهیم آشکار (explicit intents)

تماشای برخط [لینک مستقیم] فیلم «آموزش نحوه استفاده از مفاهیم آشکار»

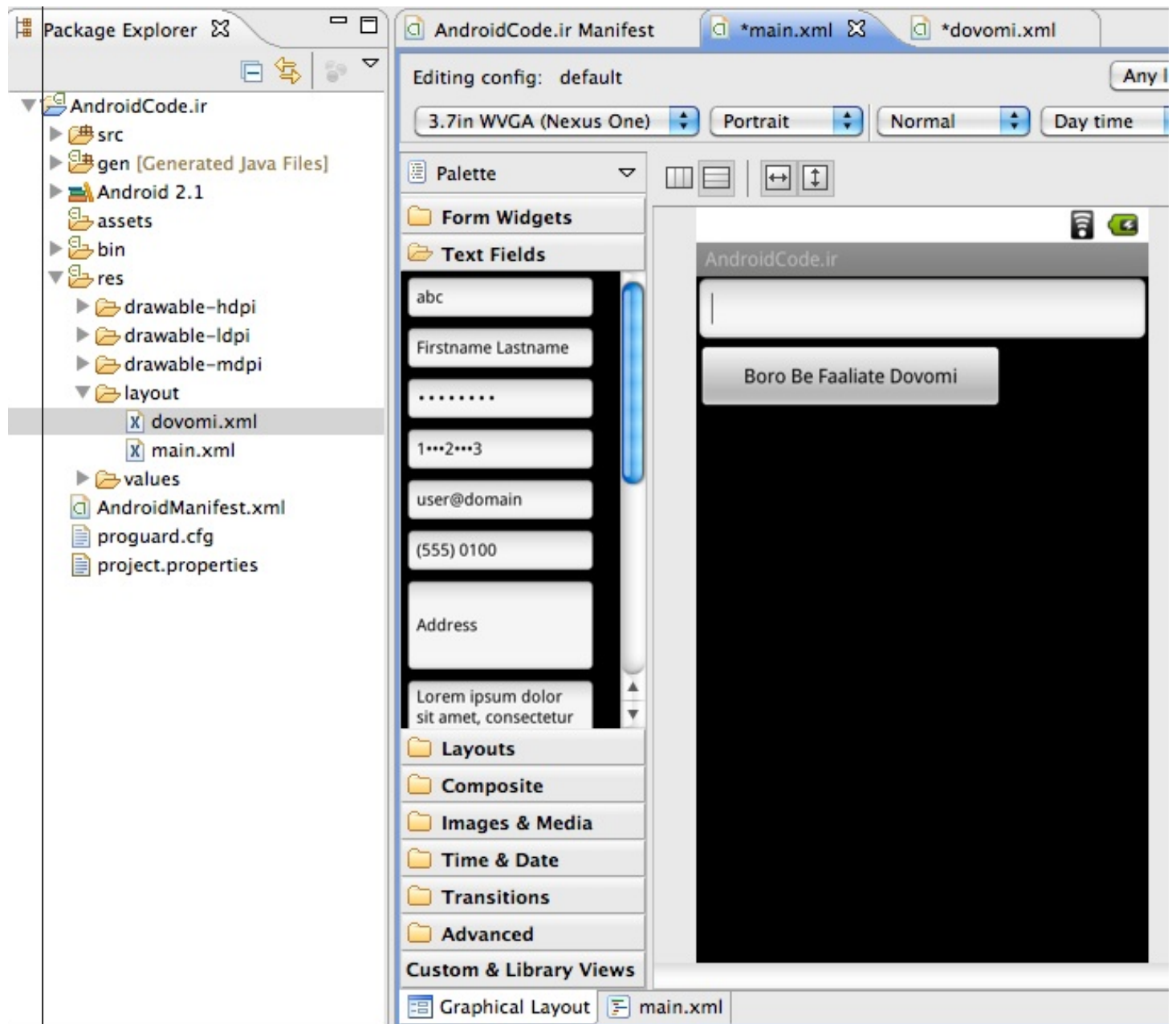
تا اینجا یاد گرفتیم که چگونه با فعالیت ها (activities) کار کنیم، اگر یادتون باشه برای شروع فعالیت Dovomi از intent استفاده کردیم، کار مفهوم (intent) اینه که به عنصری از برنامه تون رو فعال کنه.

دو نوع مفهوم (intent) وجود داره، مفاهیم آشکار و مفاهیم ضمنی یا مجازی.

با مفاهیم آشکار، یک عنصر مشخصی از برنامه رو تعیین می کنیم برای فعال شدن. مثلاً می خوایم به فعالیت (activity) دیگه رو تو برنامهمون اجرا کنیم، از به مفهوم آشکار استفاده می کنیم که آشکارا به سیستم بگه می خوام اون فعالیت رو فعال (اجرا) کنم.

مفاهیم ضمنی (که تو پست بعدی بهش می پردازیم) در واقع یک پیام کلی برای سیستم می فرستن : که آیا برنامه ای در سیستم عامل این گوشی وجود داره که بتونه این مفهوم رو اجرا کنه؟ اگر وجود داشته باشه لیست این برنامه ها به کاربر پیشنهاد می شه و کاربر یکی رو انتخاب می کنه و از این طریق اون چیزی که با مفهوم فرستاده شده (ورودی های intent) اجرا می شه.

خوب الان می خوایم رو مفاهیم آشکار تمرکز کنیم. برای شروع به پروژه بسازید با دوتا فعالیت (activity) که تو فعالیت اصلی (main) به متنی که قابل نوشتن باشه یعنی کاربر می تونه توش متن وارد کنه (از توی Text Fields اولی رو انتخاب کنید) و به دکمه (برای اینکه به فعالیت دوم بره)، فعالیت دوم هم به فیلد متنی ساده داشته باشه. (حواستون باشه که فعالیت دوم رو هم تو فایل manifest تعریف کنید).



۲. کاری که با این برنامه می‌خوایم بکنیم اینه که : کاربر یه متنی رو تو فیلد متنی قابل نوشتن می‌نویسه و دکمه رو می‌زنه، فعالیت دوم باز می‌شه و متنی که نوشته بوده رو نشون می‌ده. یعنی می‌خوایم یاد بگیریم اطلاعاتی رو از یه فعالیت به فعالیت دیگه منتقل کنیم.

اول به فایل main.java می‌ریم و یه مرجع (اشاره‌گر) از دکمه و فیلد متنی می‌گیریم. با نوشتن این کد

```
EditText matn = findViewById(R.id.editText1);
```

یه مرجع از فیلد متنی قابل نوشتن مون می‌گیریم. خوب باز هم با ctrl+1 خطاهای برنامه رو درست می‌کنیم. (اینجا کافیه فقط cast انجام بدید.) همین کارو برای دکمه انجام می‌دیم.

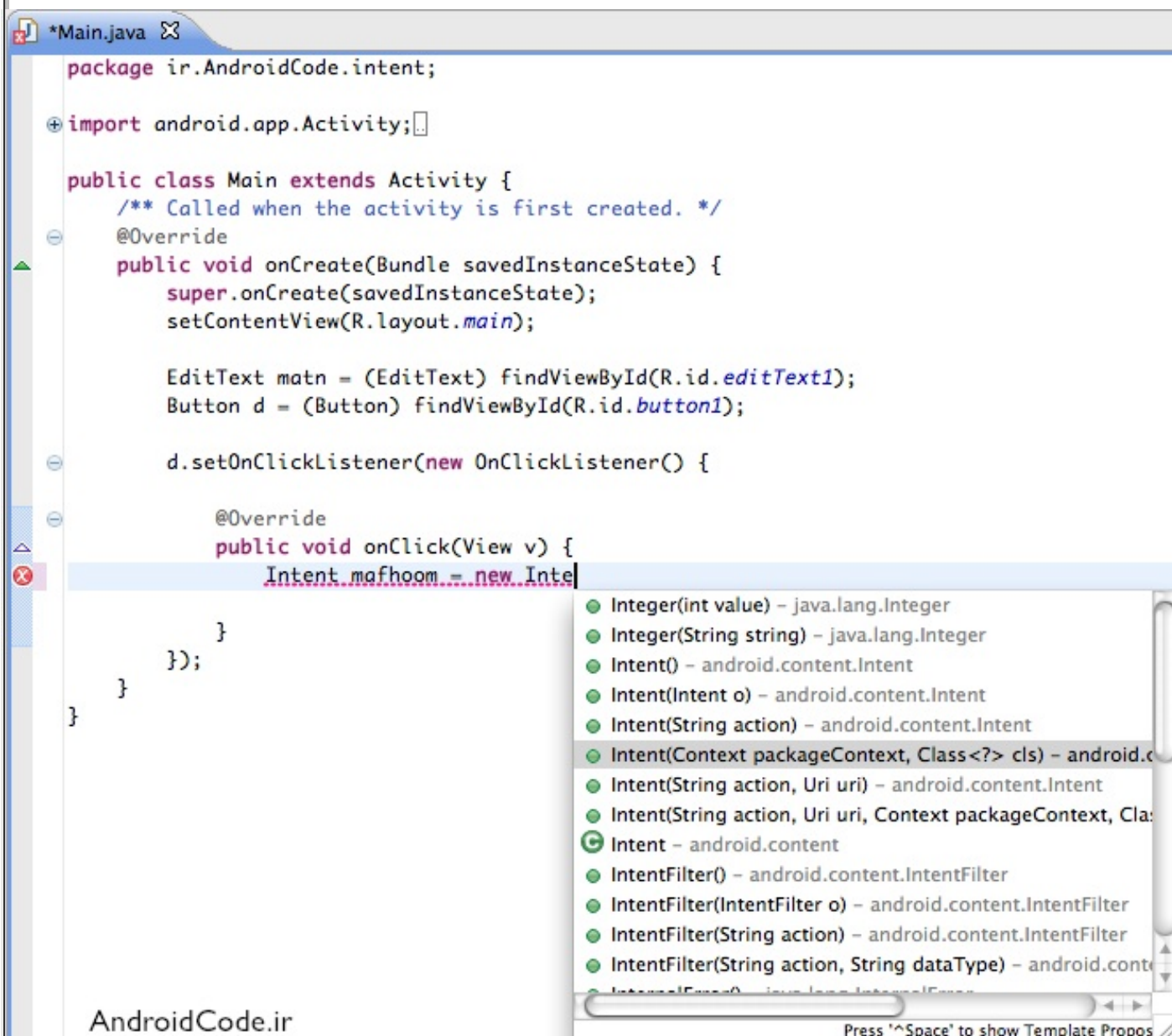
```
Button d = findViewById(R.id.button1);
```

و برای دکمه تابع `setOnClickListener` رو صدا می‌زنیم تا کارایی که قراره با فشردن دکمه انجام بشه رو تعیین کنیم. با نوشتن این کد :

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

همون طور که قبلا گفته شد بعد از نوشتن `new` (خط اول) اگه بنویسید `onclick` و بعد `ctrl+space` رو بزنید خودش بهتون ادامه‌اش رو پیشنهاد می‌ده و روی `OnClickListener` که کلیک کنید، بقیه‌اش رو خودش می‌نویسه.

۳. الان قصد داریم که به Intent بسازیم، همون طور که قبلا اشاره شد Intent یک راه برای فعال کردن جزئی از برنامه است. پس برای اینکه به نمونه از کلاس Intent بسازیم، می‌نویسیم Intent و به اسم براش می‌ذاریم و new می‌کنیم. (با ctrl+space، اون گزینه رو انتخاب می‌کنیم که مناسبه پاس کردن به فعالیته -تو تصویر مشخصه-)



```
package ir.AndroidCode.intent;

import android.app.Activity;

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

        EditText matn = (EditText) findViewById(R.id.editText1);
        Button d = (Button) findViewById(R.id.button1);

        d.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent mafhoom = new Intent
            }
        });
    }
}
```

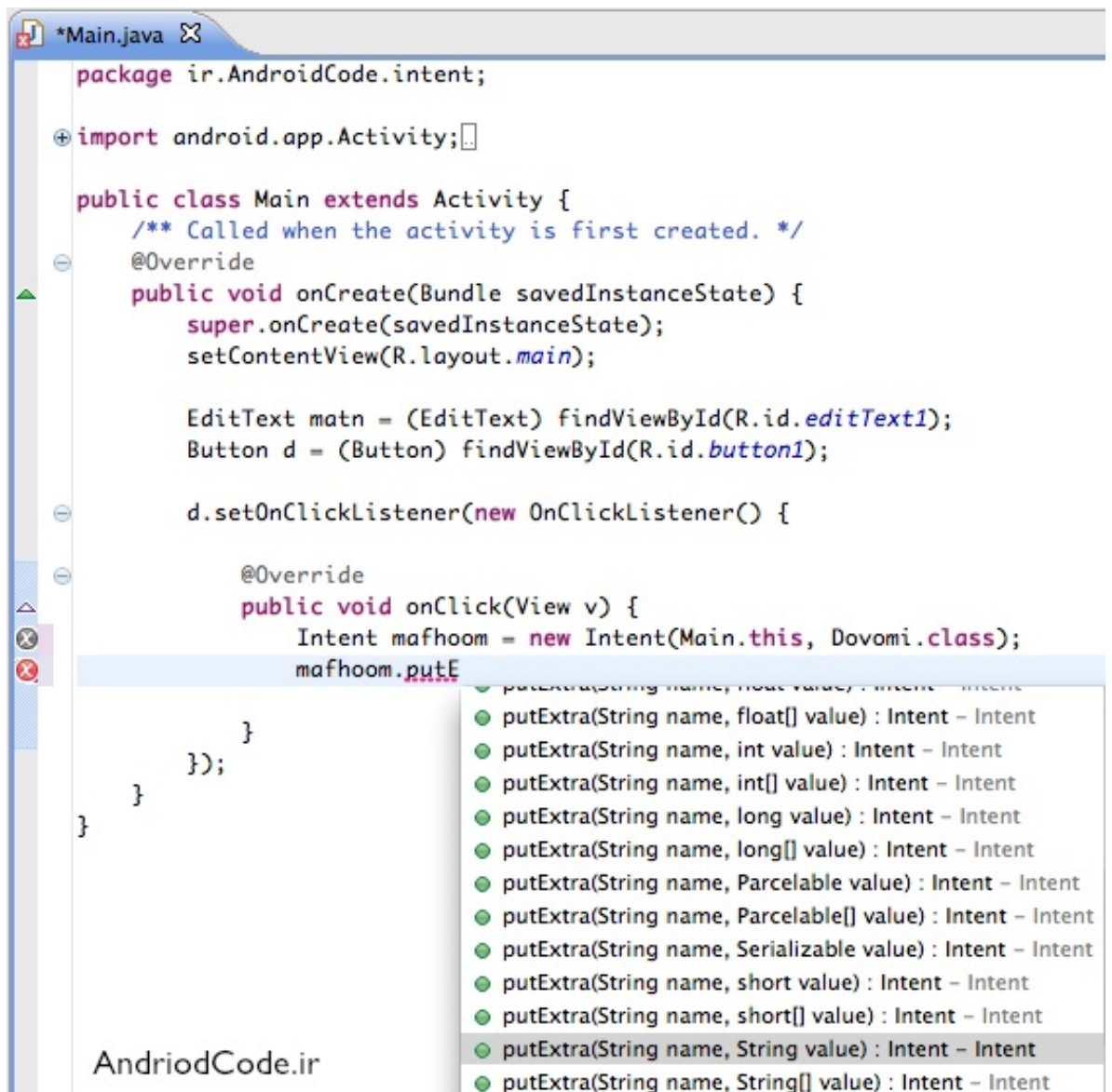
AndroidCode.ir

Press '^Space' to show Template Proposals

برای ورودی‌های سازنده‌ی Intent، اول باید مولفه‌ای که این درخواست رو کرده بنویسیم یعنی Main.this (که به مرجع به این فعالیت اصلی هست). برای ورودی دوم، باید کلاس فعالیتی رو بدیم که می‌خوایم اجرا بشه یعنی Dovomi.class. پس کدمون اینجوری می‌شه:

```
Intent mafhoom = new Intent(Main.this, Dovomi.class);
```

در واقع ما می‌تونستیم همین قسمت new Intent(Main.this, Dovomi.class) رو به عنوان ورودی startActivity به کار ببریم (مثل آموزش قبلی) ولی این دفعه می‌خوایم به چیزایی رو ضمیمه‌ی mafhoom کنیم. چیزی که می‌خوایم اضافه کنیم همون متنی که کاربر تو matn وارد می‌کنه. برای این کار از تابع putExtra از شی mafhoom استفاده می‌کنیم.



۴. ورودی اول این تابع به رشته (String) است که به عنوان اسم اون کلیدی که می‌خوایم اضافه کنیم به کار برده می‌شود و ورودی دوم از هر نوعی می‌تونه باشه و ما چون اون متن رو می‌خوایم بفرستیم، برای ورودی دوم هم نوع رشته (String) رو انتخاب می‌کنیم. اسم کلید رو می‌ذارم "matneVaredShode" و برای بازیابی اطلاعاتی که داخل matn (متنی که قرايه کاربر وارد کنه) وجود داره، باید از تابع getText استفاده کنیم و اگه دقت کنید می‌بینید که خروجی این تابع از نوع Editable هست نه از نوع رشته‌ی خام پس با اضافه کردن to_string() اونو به رشته‌ی خام تبدیل می‌کنیم.


```

*Main.java
package ir.AndroidCode.intent;

import android.app.Activity;

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

        EditText matn = (EditText) findViewById(R.id.editText1);
        Button d = (Button) findViewById(R.id.button1);

        d.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent mafhoom = new Intent(Main.this, Dovomi.class);
                mafhoom.putExtra("matneVaredShode", matn.getText());

                public Editable getText ()
                Since: API Level 1
                Return the text the TextView is displaying. If setText() was
                called with an argument of BufferType.SPANNABLE or
                BufferType.EDITABLE, you can cast the return value from this
                method to Spannable or Editable, respectively. Note: The
                content of the return value should not be modified. If you want
                a modifiable one, you should make your own copy first.

            }
        })
    }
}

```

```
mafhoom.putExtra("matneVaredShode", matn.getText().toString());
```

از `matn` ایراد می‌گیره و با زدن `ctrl+1` بهمون پیشنهاد می‌کنه که اونو به `final` تغییر بدیم (چون قراره ارزش تو یه کنترل کننده وقایع ناشناس استفاده کنیم).

```

package ir.AndroidCode.intent;

import android.app.Activity;

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

        EditText matn = (EditText) findViewById(R.id.editText1);
        Button d = (Button) findViewById(R.id.button1);

        d.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent mafhoom = new Intent(Main.this, Dovomi.class);
                mafhoom.putExtra("matneVaredShode", matn.getText().toString());

            }
        });
    }
}

```

AndroidCode.ir

ما اون اطلاعاتی که می‌خواستیم به فعالیت Dovomi بفرستیم رو به mafhoom اضافه کردیم، حالا وقتی می‌خوایم فعالیت جدید رو شروع کنیم، این اطلاعات براش فرستاده می‌شه. تابع startActivity رو استفاده می‌کنیم و mafhoom رو به عنوان ورودی براش می‌فرستیم.

۵. خوب تا اینجا دیگه کار ما با Main.java تموم می‌شه، حالا باید به Dovomi.java بریم و اطلاعاتی که همراه mafhoom فرستاده شده بازبازی کنیم و تو textView نشون بدیم. برای شروع اول به مرجع از فیلد متنی می‌گیریم با این کد :

```
TextView matni = (TextView) findViewById(R.id.textView1);
```

برای اینکه این فیلد متنی، متن دلخواه ما رو نشون بده باید از تابع setText استفاده کنیم (هرچی به عنوان ورودی بفرستیم رو نشون می‌ده)، برای گرفتن همون متنی رو که توسط mafhoom فرستاده شده، اول باید با getIntent مفهوم می‌گیریم که این فعالیت رو اجرا کرده رو به دست بیاریم، و بعد از تابع getIntent برای گرفتن اطلاعات داخل intent استفاده می‌کنیم، که در واقع تمام اطلاعاتی که بهش داده بودیم رو برمیگردونه. اما ما فقط اون قسمتی از اطلاعاتش رو می‌خوایم که با کلید "matneVaredShode" مشخص شده (اگرچه در این تمرین فقط همین بدونه رو همراه mafhoom فرستادیم). و چون اون اطلاعاتی که می‌خوایم از نوع رشته است، از تابع getString استفاده می‌کنیم و کلید رو بهش می‌دیم.

```
matni.setText(getIntent().getExtras().getString("matneVaredShode"));
```

```

package ir.AndroidCode.intent;

import android.app.Activity;

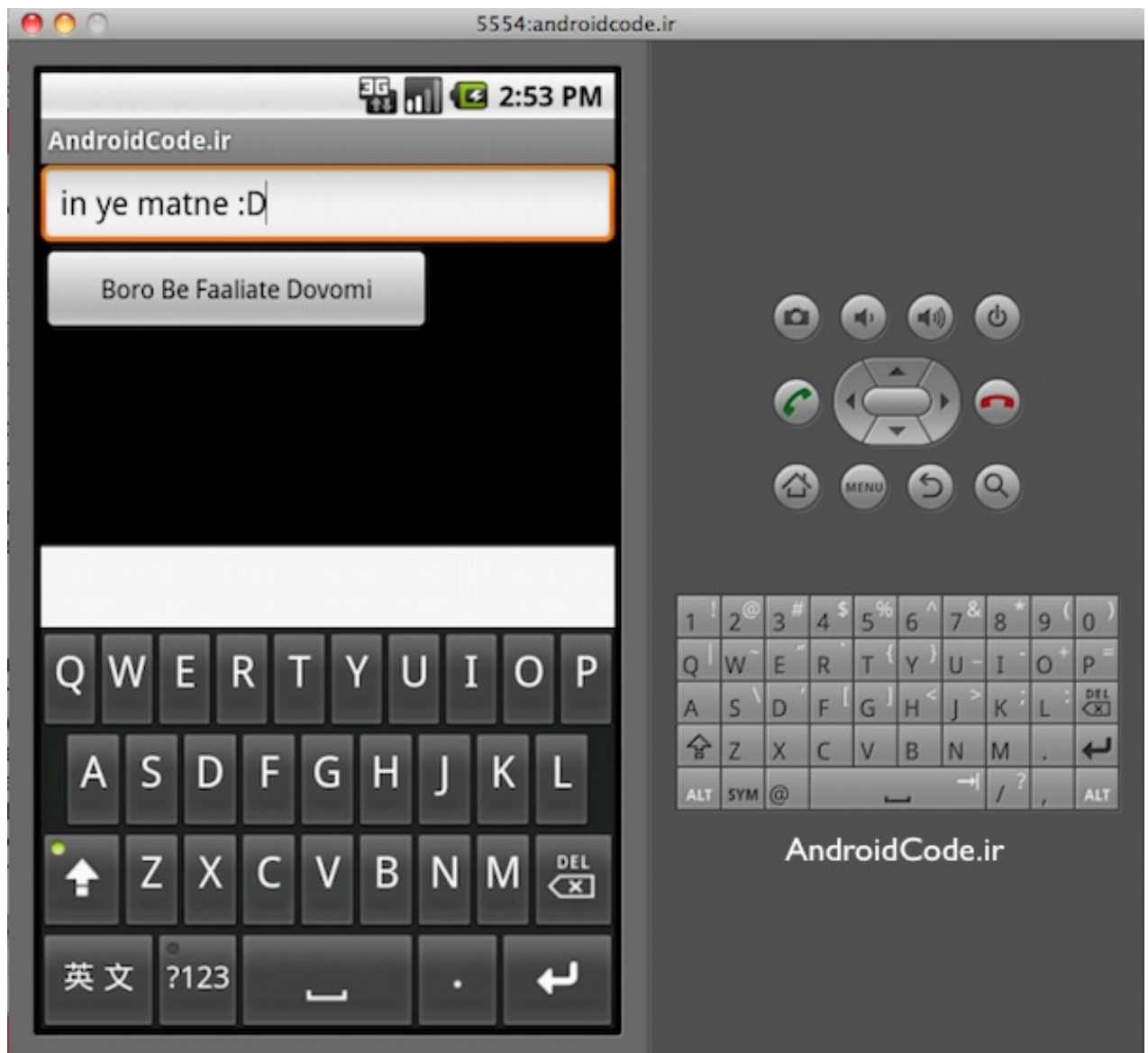
public class Dovomi extends Activity {

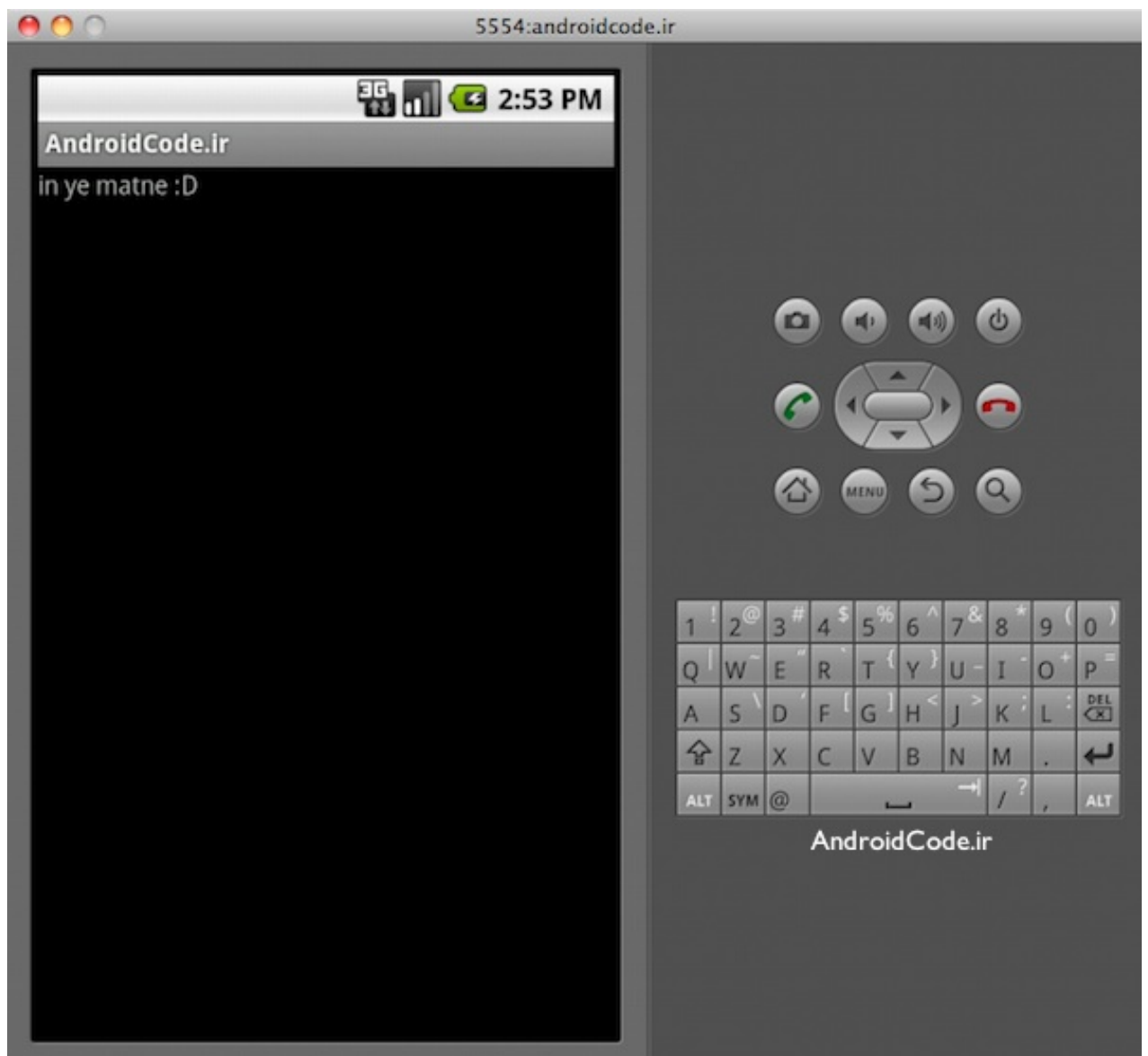
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dovomi);

        TextView matni = (TextView) findViewById(R.id.textView1);
        matni.setText(getIntent().getExtras().getString("matneVaredShode"));
    }
}

```

۶. حالا برنامه رو اجرا می‌کنیم و می‌بینیم همون متنی که تو فعالیت اصلی وارد کردیم تو فعالیت دوم نشون داده می‌شه :





نوشته شده توسط وجدانی

تمامی حقوق برای سرویس بلاگ بیان محفوظ است

آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

- [خانه](#)
- [دانلود JDK و SDK برای ایرانیان](#)
- [سوالات متداول](#)
- [جدول محتوا](#)