

Лабораторная работа №5. Первая программа для Android

Цель работы: Ознакомиться с процессом написания программы в Eclipse и ADT (Android Development Tools) для операционной системы Android. Ознакомиться с принципом добавления в программу элементов интерфейса (кнопки, текстовые поля ввода и т.п.) и обработкой событий этих элементов в Android.

Методические указания.

Считаем, что после предыдущих лабораторных работ (№1..№4) у нас на компьютере уже установлены Java, Eclipse и ADT.

1. Запустите Eclipse. Выберите File/New/Android Application Project (рис. 1).

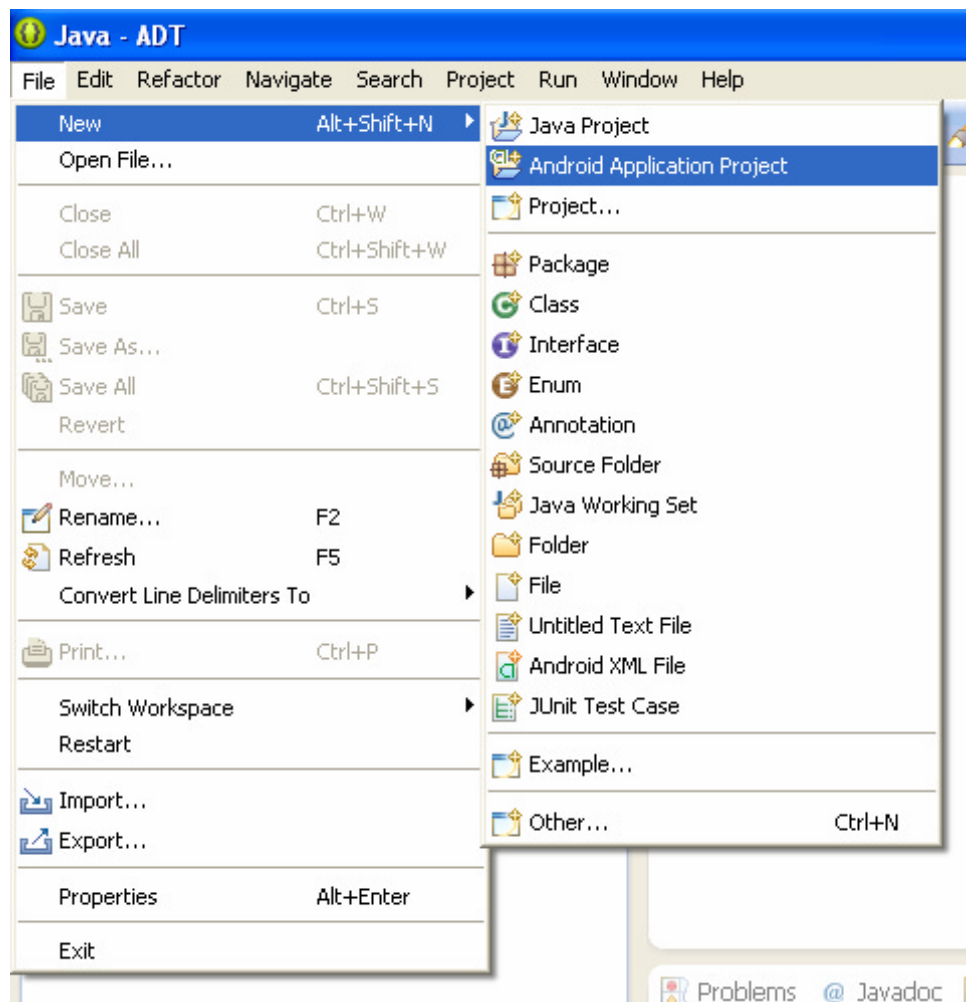


Рис. 1. Создание нового проекта для приложения Android

2. На экране появится окно выбора параметров проекта, приведенное на рис. 2.

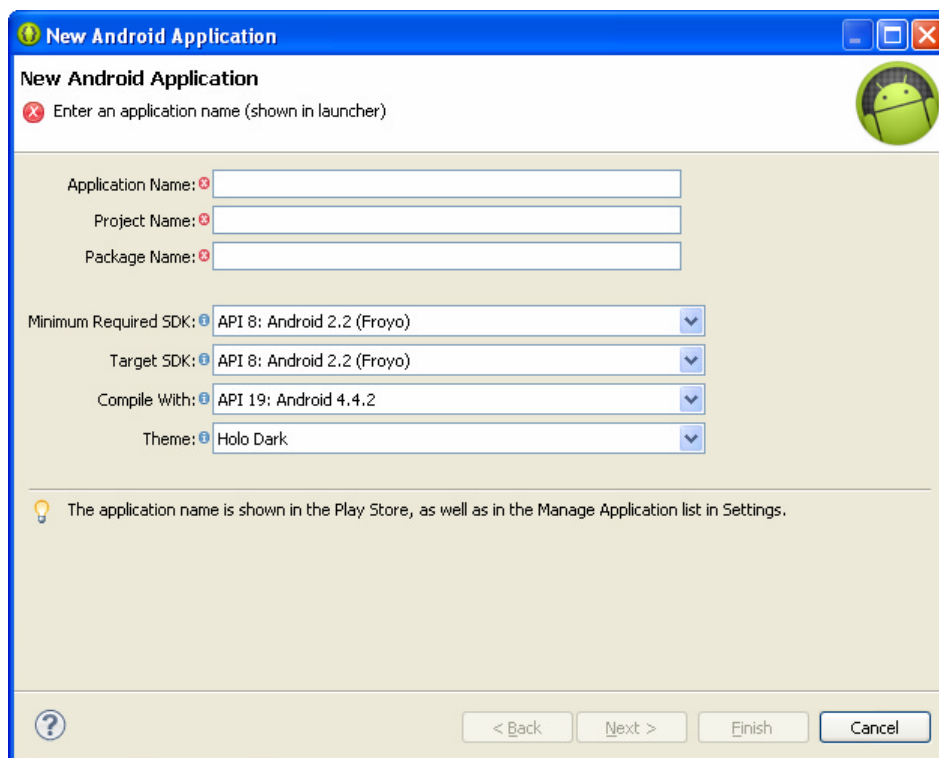


Рис. 2. Окно выбора параметров проекта

В качестве “Application Name” задайте свою фамилию латинскими буквами.

В полях “Minimum Required SDK:” и “Target SDK:” выберите “API 8: Android 2.2 (Froyo)”. В поле “Compile With:” выберите “API 19: Android 4.4.2”. В поле “Theme:” выберите “Holo Dark” (рис. 3).

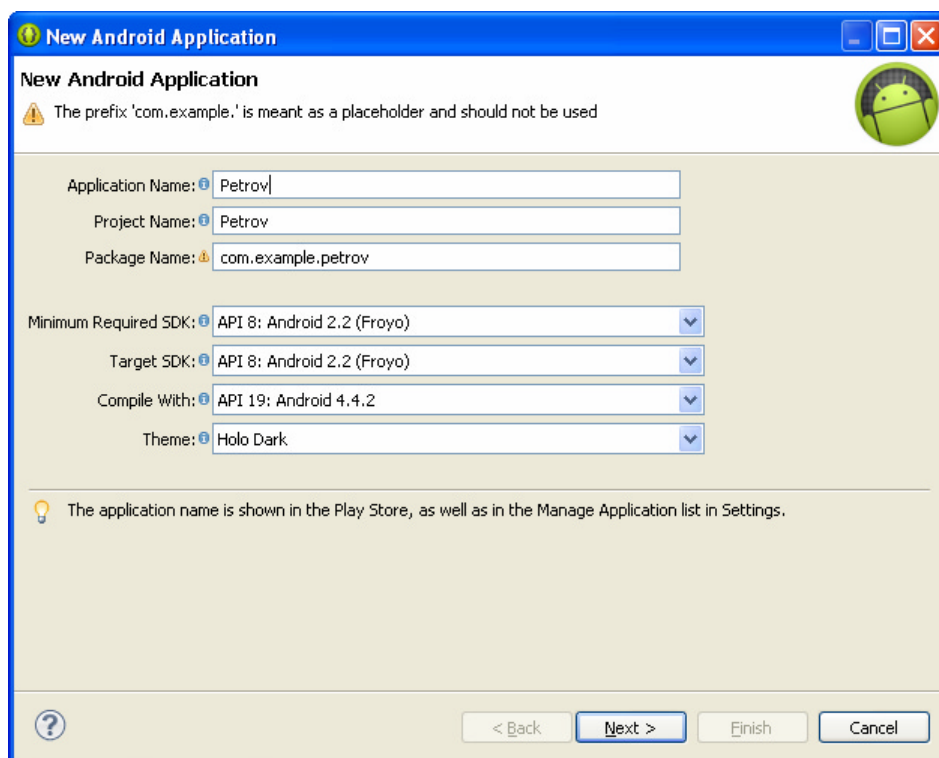


Рис. 3. Окно с установленными параметрами

Появится окно (рис. 4), в котором оставьте все без изменений и нажмите Next.

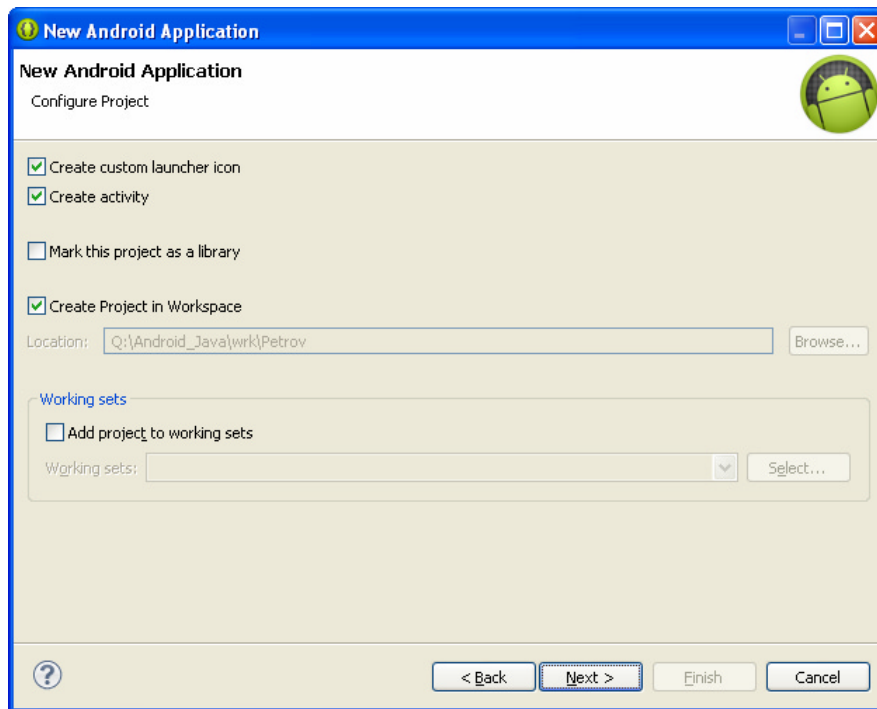


Рис. 4. Окно конфигурации проекта

Появится окно (рис. 5) настройки иконки, в котором тоже оставьте все без изменений и нажмите Next.

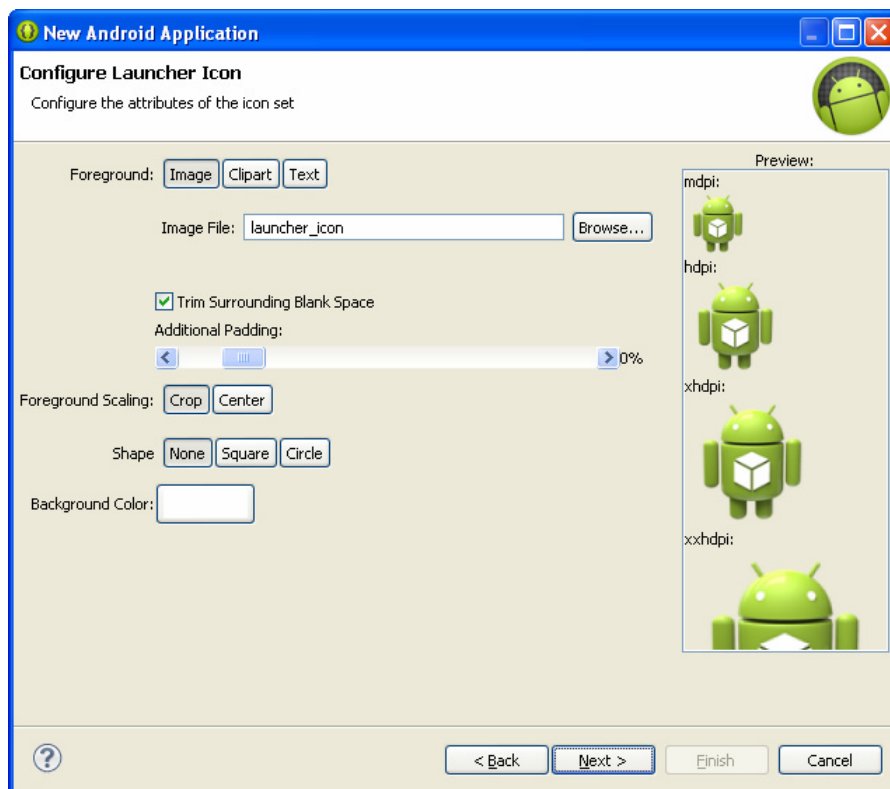


Рис. 5. Окно конфигурации иконки

В появившемся окне (рис. 6) выберите “Blank Activity” и нажмите Next.

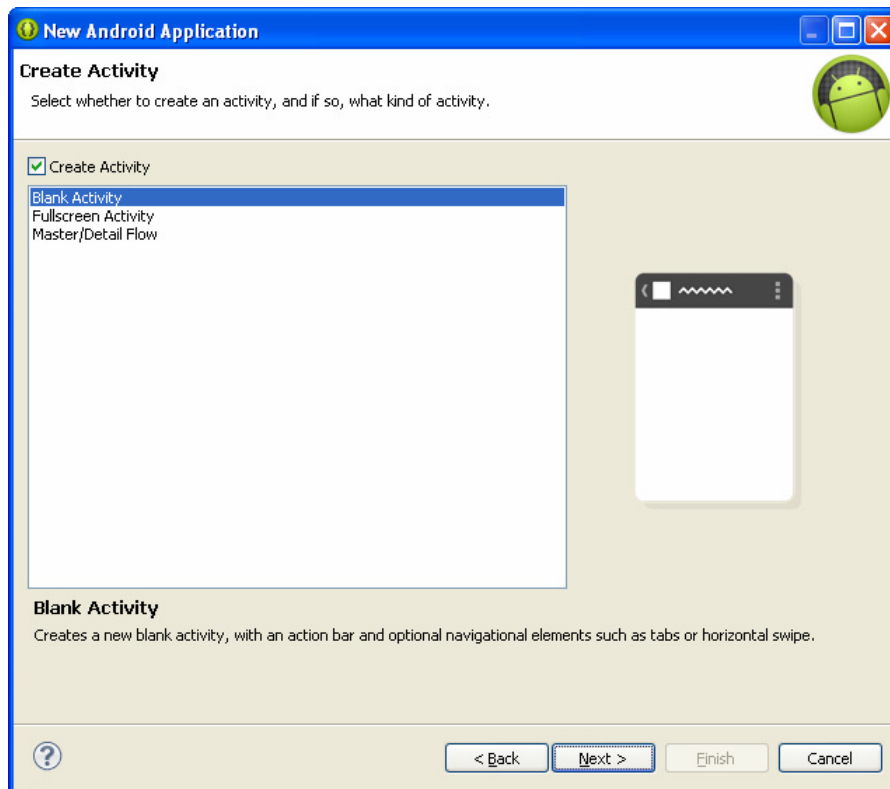


Рис. 6. Окно выбора вида экрана программы

В появившемся окне (рис. 7) оставьте все без изменений и нажмите Finish.

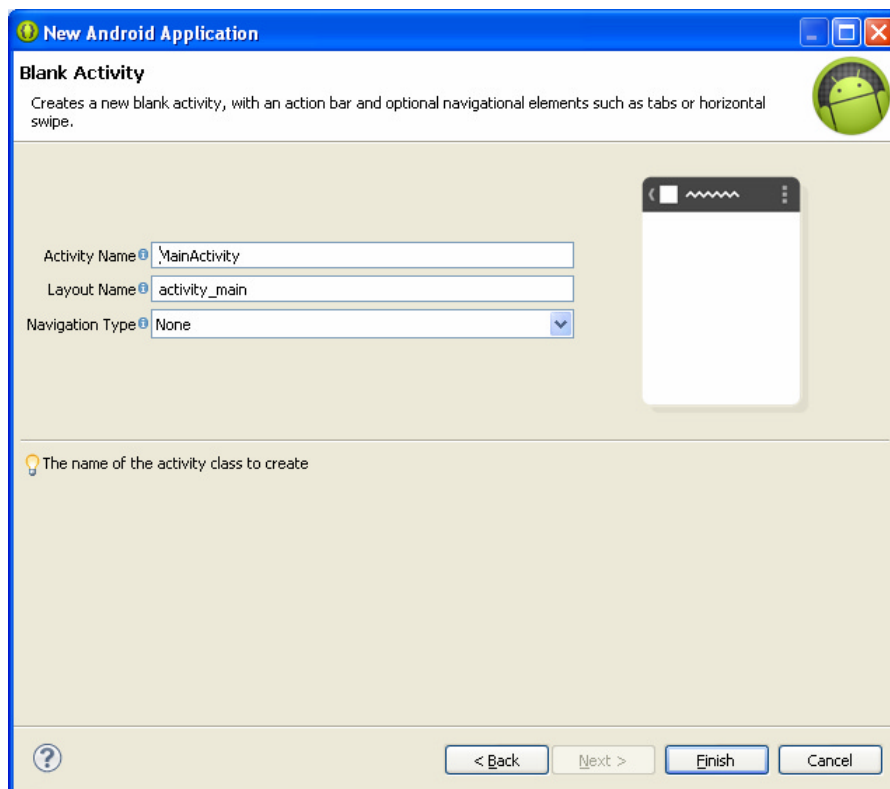


Рис. 7. Завершающее окно настройки проекта

3. После всего сделанного проект (программу) уже можно откомпилировать, установить на телефон или планшет, и запустить.

Чтобы запустить программу на компьютере, нужно вначале создать и запустить виртуальный телефон.

Для того, чтобы создать виртуальный телефон, запустите Android Virtual Device Manager (пункт меню “Window\Android Virtual Device Manager) и в нем нажмите кнопку New. Появится окно, показанное на рис. 8.

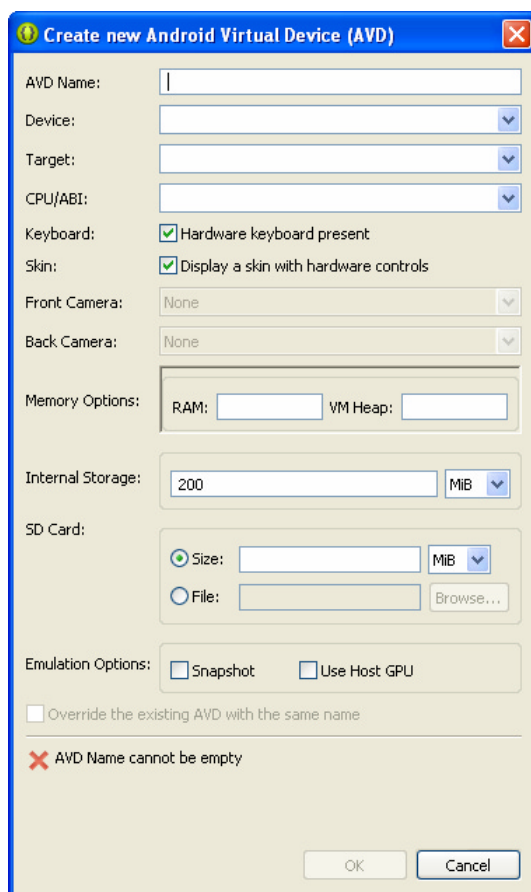


Рис. 8. Окно создания виртуального устройства (телефона, планшета)

В поле “AVD Name” введите имя телефона (любое имя латинскими буквами).

В поле “Device” выбирается тип телефона. Чем больше будет экран и память этого телефона, тем медленнее будет работать виртуальная модель этого телефона и больше потребуется памяти нашего компьютера. Поэтому, для этой лабораторной работы выберите телефон с небольшим экраном - 2.7” QVGA slider (240 x 320: ldpi).

В поле Target укажите версию операционной системы Android, которая будет установлена на этом виртуальном телефоне. Чем более новой будет эта версия, тем медленнее будет загружаться телефон и работать эмулятор. Для данной лабораторной работы укажите “Android 2.2 - API Level 8”.

Поле CPU/ABI заполнится автоматически.

Для поля Back Camera оставьте “None”.

В поле “Memory Options” поставьте “RAM” 512 и “VM Heap” 16. Это оперативная память телефона и, если она будет слишком большой, то эмулятор

телефона будет работать слишком медленно. Если же она будет слишком маленькой, то операционная система и программы могут не запускаться на этом телефоне.

В поле “Internal Storage” ставьте 100 MiB. Это объем встроенной карты памяти телефона.

Поле “SD Card” оставьте пустым.

В итоге заполненное окно должно выглядеть, как это показано на рис. 9.

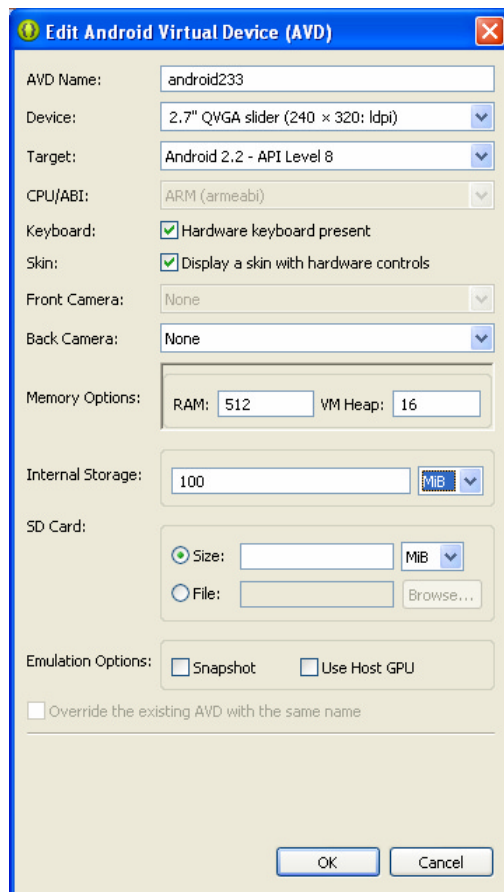


Рис. 9. Выбранные параметры для виртуального телефона

В окне “Android Virtual Device Manager” (рис. 10) нажмите кнопку “Start”.

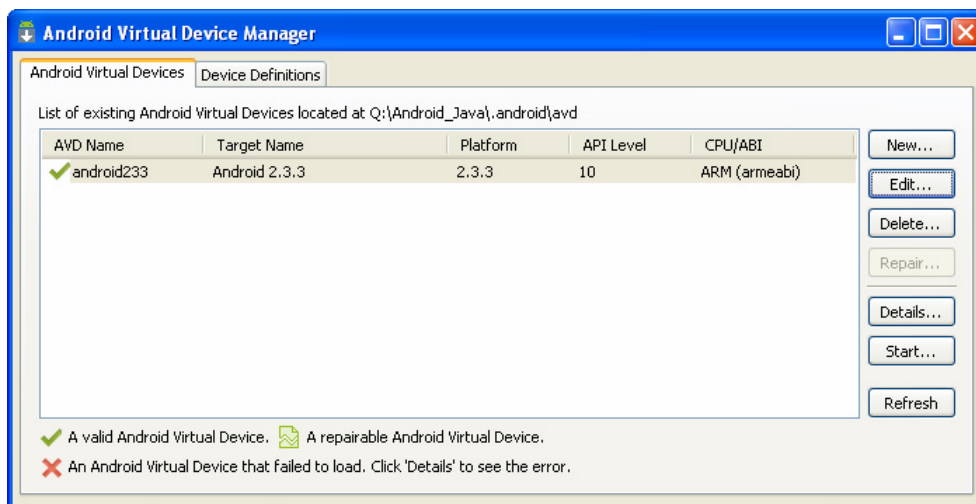


Рис. 10. Окно “Android Virtual Device Manager” с созданным устройством

Появится окно запуска (рис. 11).

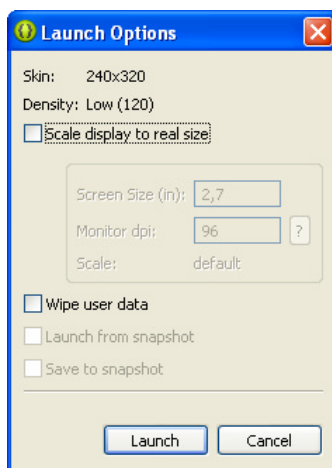


Рис. 11. Окно запуска эмулятора телефона

В поле “Scale display to real size” уберите галочку, если она там есть, и нажмите кнопку Launch. Начнет запускаться эмулятор телефона. При этом для связи Eclipse и этого телефона создается специальный локальный сервер и, если ваш файервол будет в этот момент спрашивать, разрешать ли соединение с ним, то разрешайте.

Появится окно виртуального телефона и начнется загрузка операционной системы на телефоне. Этот процесс займет некоторое время (несколько минут) и завершится примерно таким экраном, как на рис. 12.

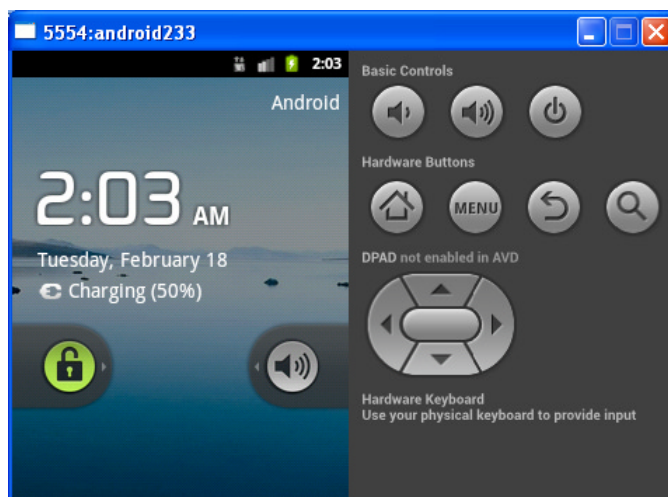


Рис. 12. Экран виртуального телефона после загрузки

Это практически полноценный телефон, на котором установлены самые настоящие программы и, который даже показывает зарядку батареи. После этого окно Android Virtual Device Manager можно закрыть, а окно с виртуальным телефоном должно оставаться открытым.

4. Теперь попробуйте запустить созданный вами проект (программу) на этом виртуальном телефоне. Для этого в меню Project Explorer нажмите правую кнопку

мышью на названии проекта и в выпавшем меню выберите “Run As” и “Android Application” (рис.

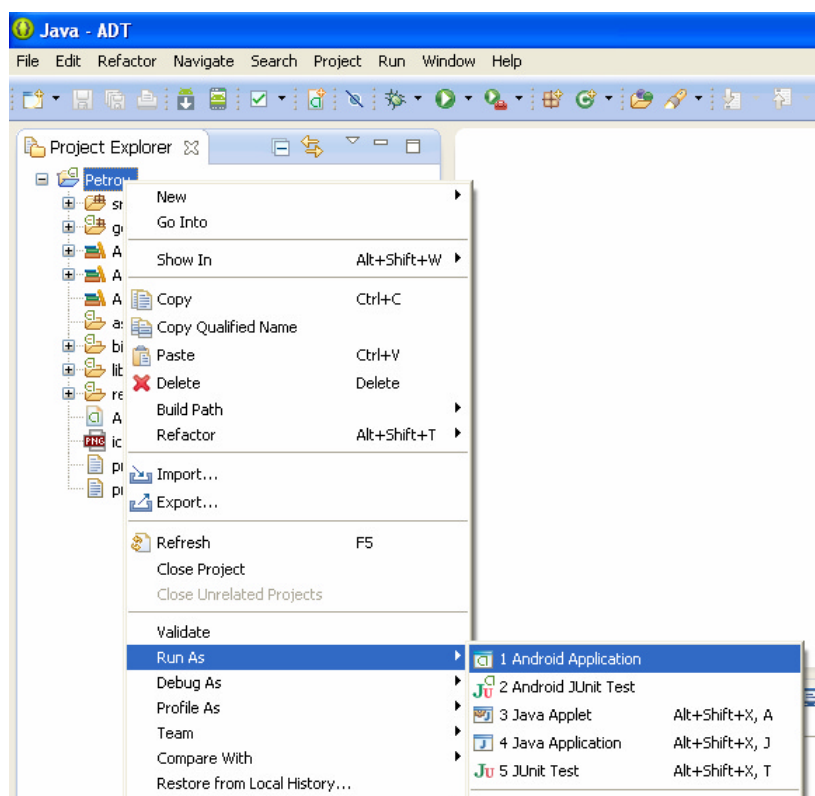


Рис. 13. Запуск приложения

Если проект не содержит ошибок, то пройдет некоторое время (возможно, несколько минут), пока проект откомпилируется и установится на виртуальном телефоне и, разблокировав экран виртуального телефона, Вы увидите запущенное приложение с надписью “Hello, world!” (рис. 14). Это созданная нами программа. В этот момент фаервол так же может беспокоиться и запрашивать разрешение для выполнения запросов в серверу. Разрешайте их.

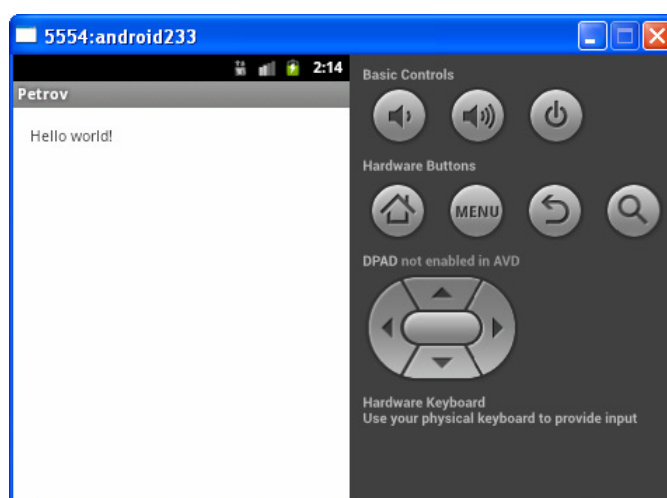


Рис. 14. Откомпилированный и запущенный проект

5. Откомпилированный и запущенный только что созданный проект, как мы убедились, уже содержит надпись “Hello, world!” и больше ничего. Как убрать эту надпись и как вместо нее добавить что-то иное?

В Project Explorer выберите ваш проект и нажмите res\layout\activity_main.xml (рис. 15).

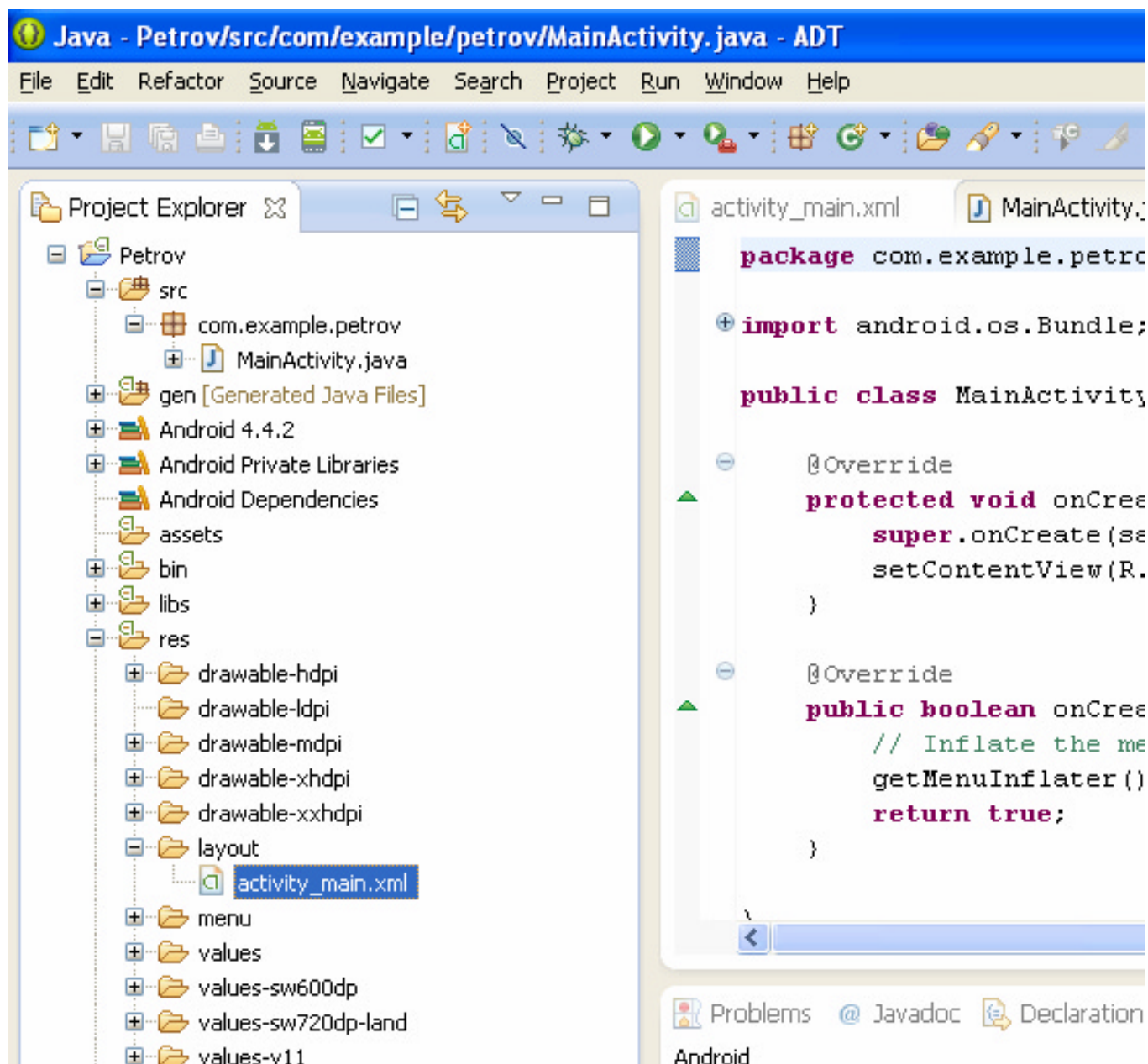


Рис. 15. Окно Project Explorer

Откроется окно визуального редактирования содержимого экрана нашей программы (приложения) (рис. 16). С расположенного в левой части окна Palette перетащите на окно программы кнопку “Button”. Нажмите на эту кнопку и в окне Properties в правой части экрана в поле текст введите новую надпись для нее “Нажми меня!” (рис. 17).

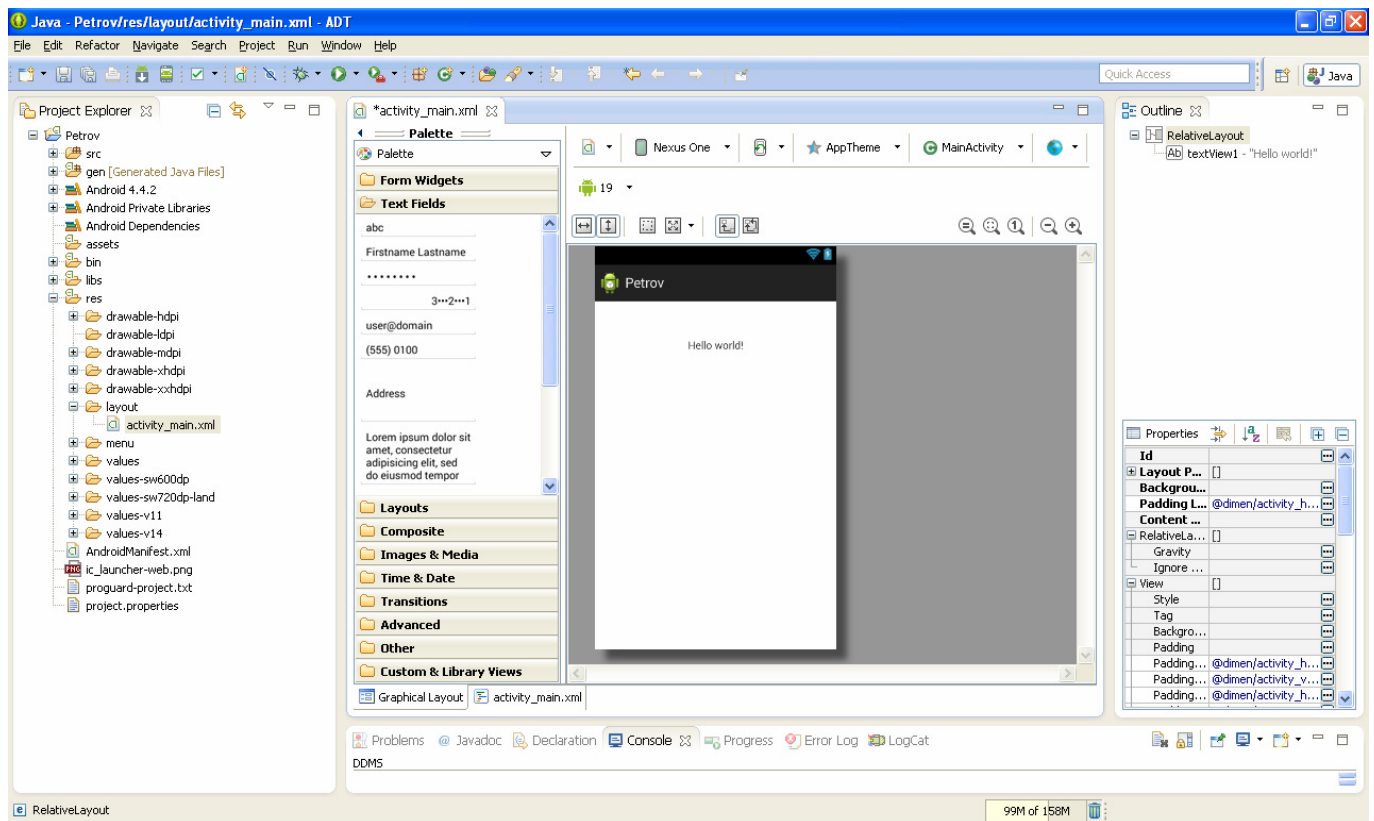


Рис. 16. Окно визуального редактирования содержимого экрана программы

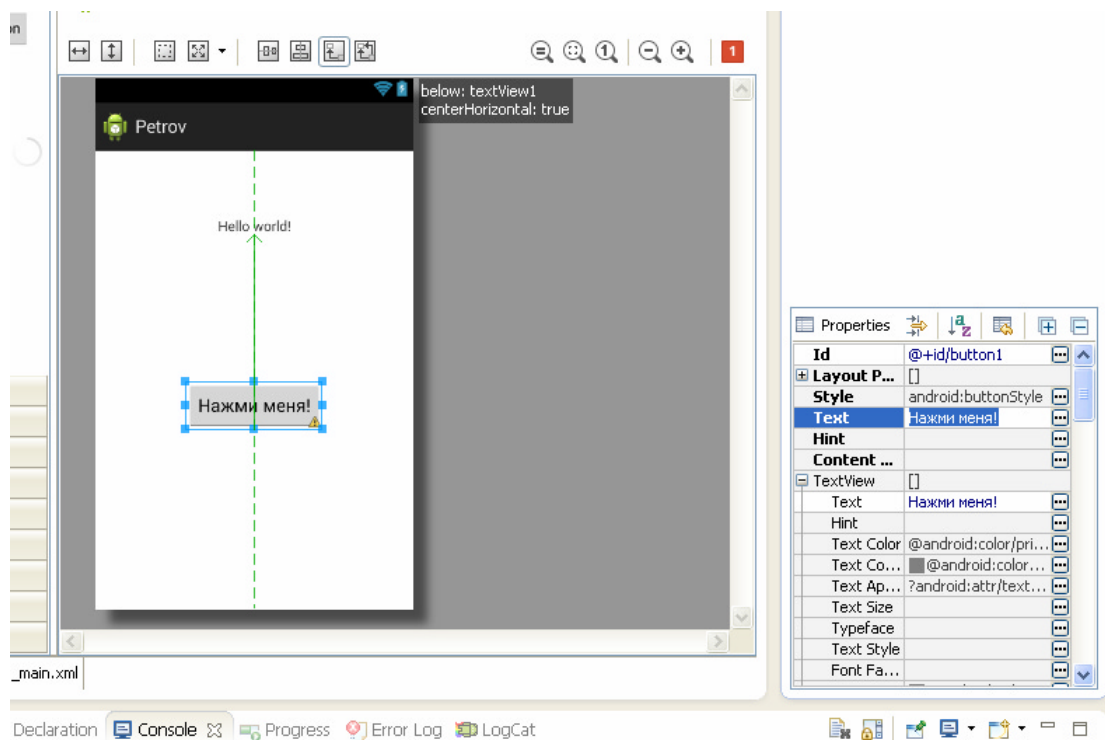


Рис. 17. Добавление кнопки и изменение надписи на ней

Таким же образом измените и текст надписи “Hello, world!” на свои имя и фамилию, например, на Владимир Петров. Снова запустите проект на выполнение и в окне виртуального телефона появится новый вариант нашей программы (рис. 18).

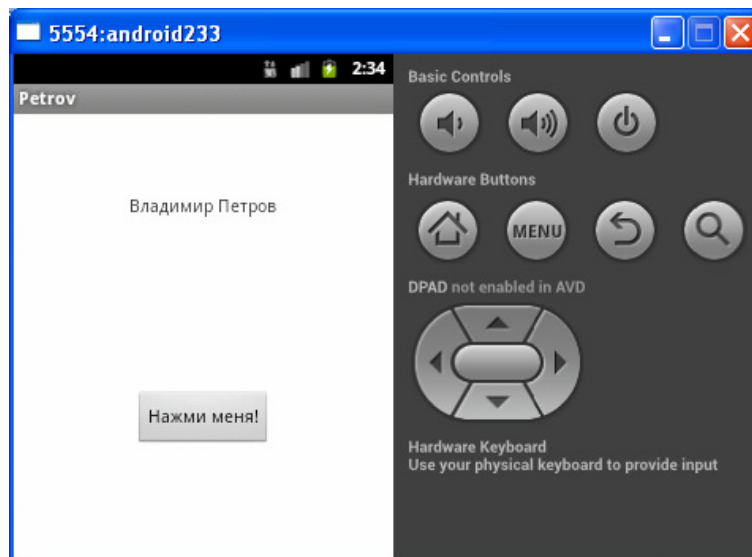


Рис. 18. Измененная программа

Можно попробовать понажимать на эту кнопку, но ничего не произойдет. Почему? Мы пока не написали никакого текста программы, который бы реагировал на это нажатие.

6. В редакторе нажмите на кнопку и посмотрите в окне Properties значение поля Id. Допустим, там написано “@+id/button1”. Это значит, что Id кнопки - “button1”. По этому Id к ней можно обращаться из нашей программы.

Аналогично, нажав на надпись, можно узнать и ее Id. Допустим, в поле Id надписи написано «@+id/textView1». Значит Id надписи - “textView1”.

В Project Explorer для своего проекта выберите src\MainActivity.java (рис. 19).

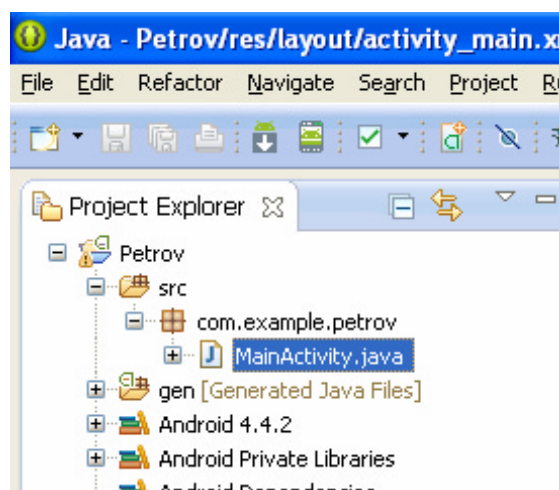


Рис. 19. Выбор для редактирования MainActivity.java

MainActivity.java откроется в окне редактора.

Чтобы работать с текстовой надписью и кнопкой, для начала нужно создать поля (переменные) этих типов. В Java для Android тип данных для кнопок называется Button, а тип данных для надписей - TextView.

Добавьте в класс MainActivity поле кнопки и поле текстовой строки, придумав для них любые названия, например katya и olesya:

```
TextView katya;  
Button olesya;
```

Eclipse подчеркнет TextView и Button как ошибки. Чтобы устранить ошибку, добавьте в программу import обоих типов данных.

Теперь в методе onCreate (он будет запускаться при инициализации окна программы) добавьте следующие две строки:

```
katya = (TextView) findViewById(R.id.textView1);  
olesya = (Button) findViewById(R.id.button1);
```

Смысл этих двух строк в том, что по известным нам ранее Id текстовой строки и кнопки (textView1 и button1) мы находим их в окне программы и запоминаем в специально для этого созданных нами полях katya и olesya.

Теперь остается только назначить обработчик для события нажатия на кнопку. Для этого в этом же методе onCreate добавьте строку:

```
olesya.setOnClickListener(обработка);
```

В класс добавьте новый экземпляр класса OnClickListener следующим образом:

```
OnClickListener обработка = new OnClickListener() {  
    public void onClick(View v) {  
        katya.setText("Кнопка была нажата!");  
    }  
};
```

Здесь действие katya.setText("Кнопка была нажата!"); будет выполняться при нажатии кнопки. Вместо этого действия можно по аналогии прописать в качестве реакции на нажатие кнопки любое другое действие.

Eclipse подчеркнет OnClickListener как ошибку и нужно его импортировать в программу. Eclipse предложит два варианта импорта (рис. 20).

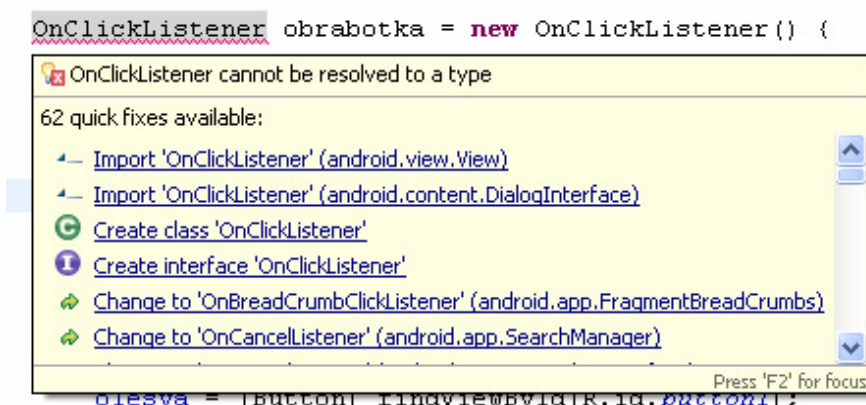


Рис. 20. Для импорта OnClickListener нужно выбрать android.view.View

Так же придется импортировать тип данных View, подведя к нему мышку в Eclipse. После этого ошибок не останется и можно снова откомпилировать и запустить программу. В этот раз после нажатия на кнопку текст с “Владимир Петров” изменится на “Кнопка была нажата!” (рис. 21).

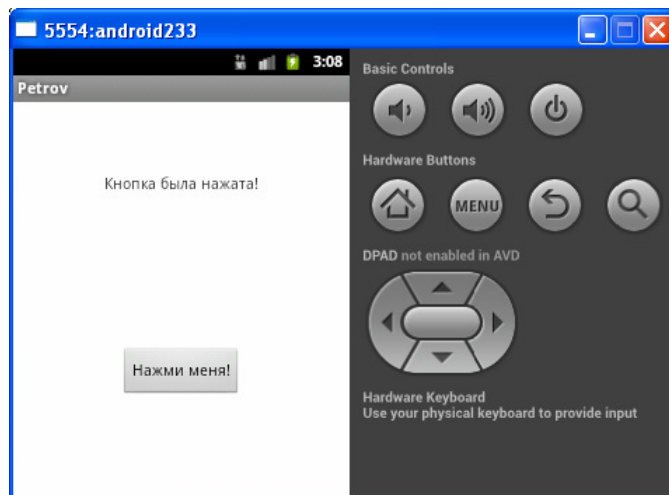


Рис. 21. Результат нажатия на кнопку в запущенной программе

7. Покажите откомпилированную и запущенную программу преподавателю.
8. Если у вас есть реальный планшет или телефон, работающий под управлением операционной системы Android, то попробуйте установить туда написанную вами программу и запустить ее. Для этого из папки bin вашего проекта возьмите файл с именем проекта и расширением “.apk”. Это стандартное расширение для установочных программ в операционной системе Android. Самостоятельно разберитесь или почитайте в Google, как устанавливать программу из apk-файла.
9. Дома самостоятельно попробуйте модифицировать написанную Вами программу, добавляя другие элементы интерфейса, реакции на нажатия кнопки и т.д.