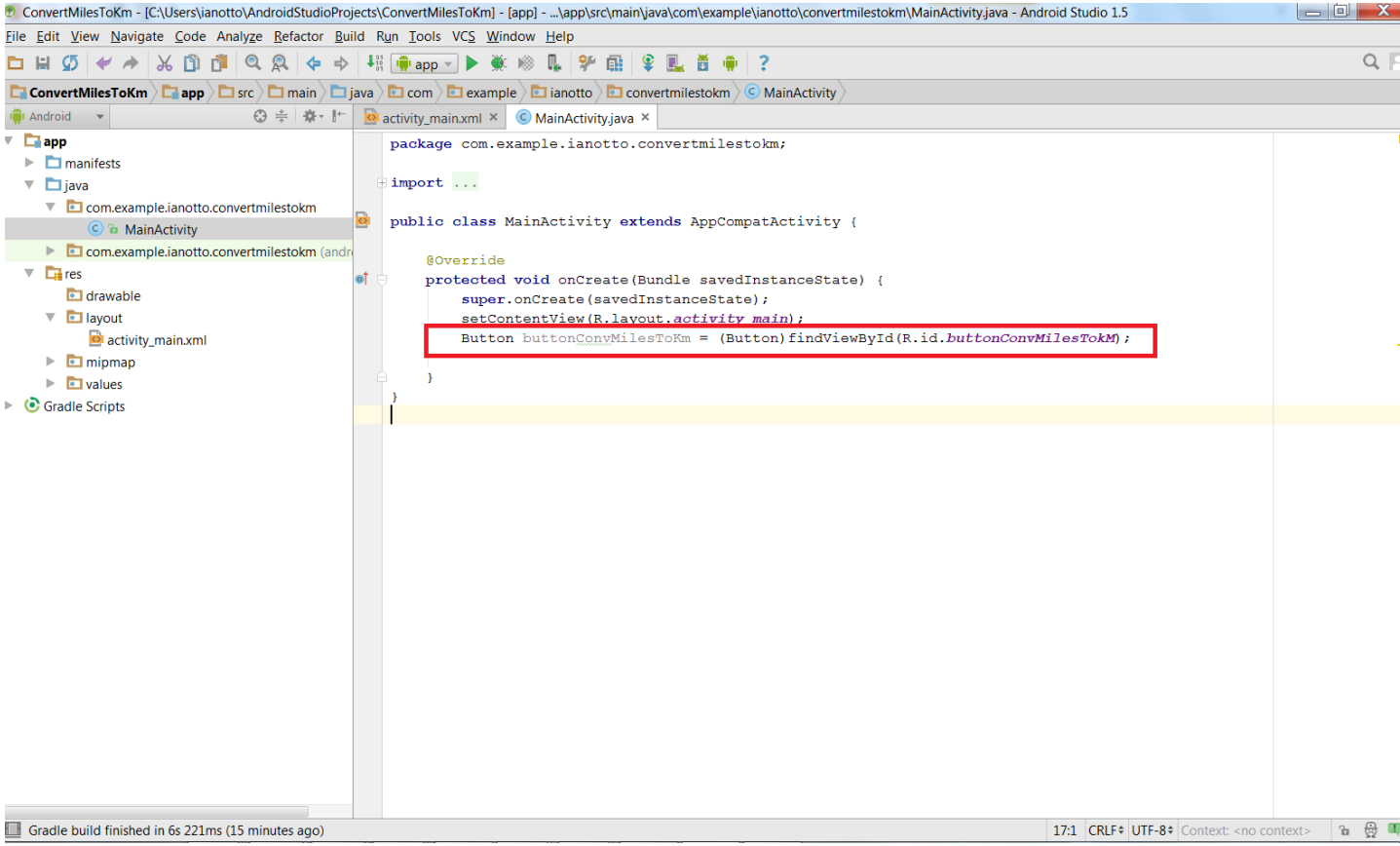
# Coding and testing the application

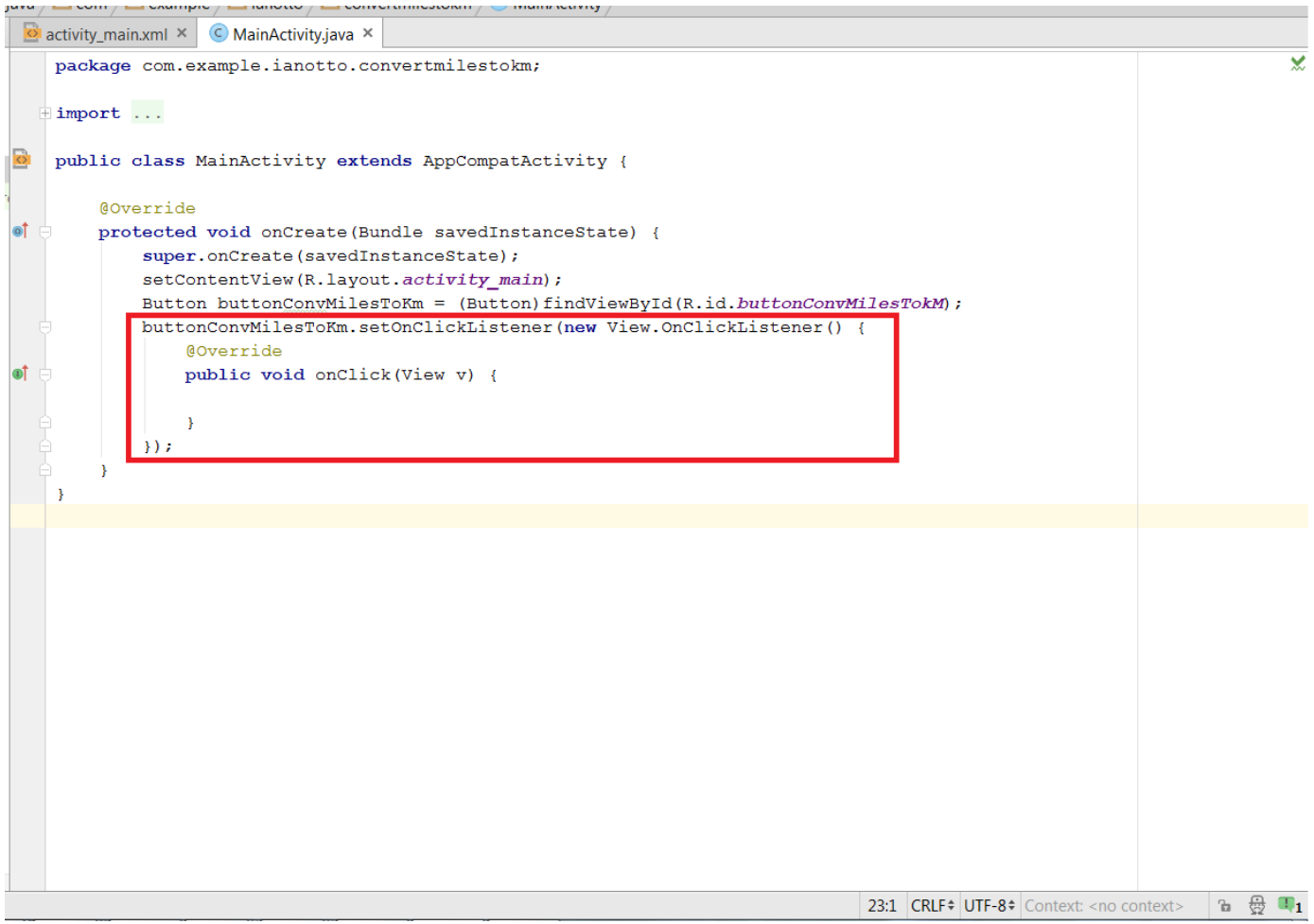
- When an application is compiled a class named **R** is created.
- It contains references to the application resources.
- To get a reference to the button object we can make a call to the findViewById method :

```
Button buttonConvMilesToKm =
    (Button)findViewById(R.id.buttonConvMilesTokM);
```

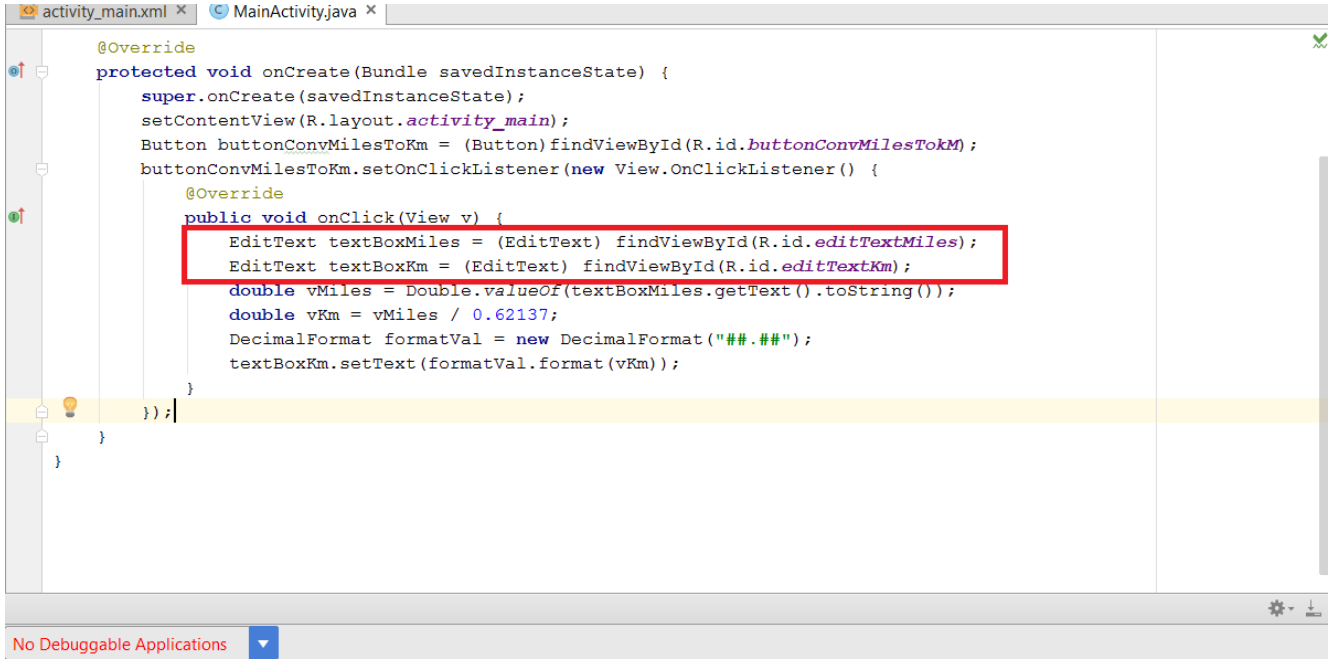
- The button object must be defined in the *OnCreate* function contained in the file *MainActivity.java* as we can see in the figure.
- We obtain a reference to the *ButtonConvertMilesToKm* component and assign it to the button object using the *findViewById* function.



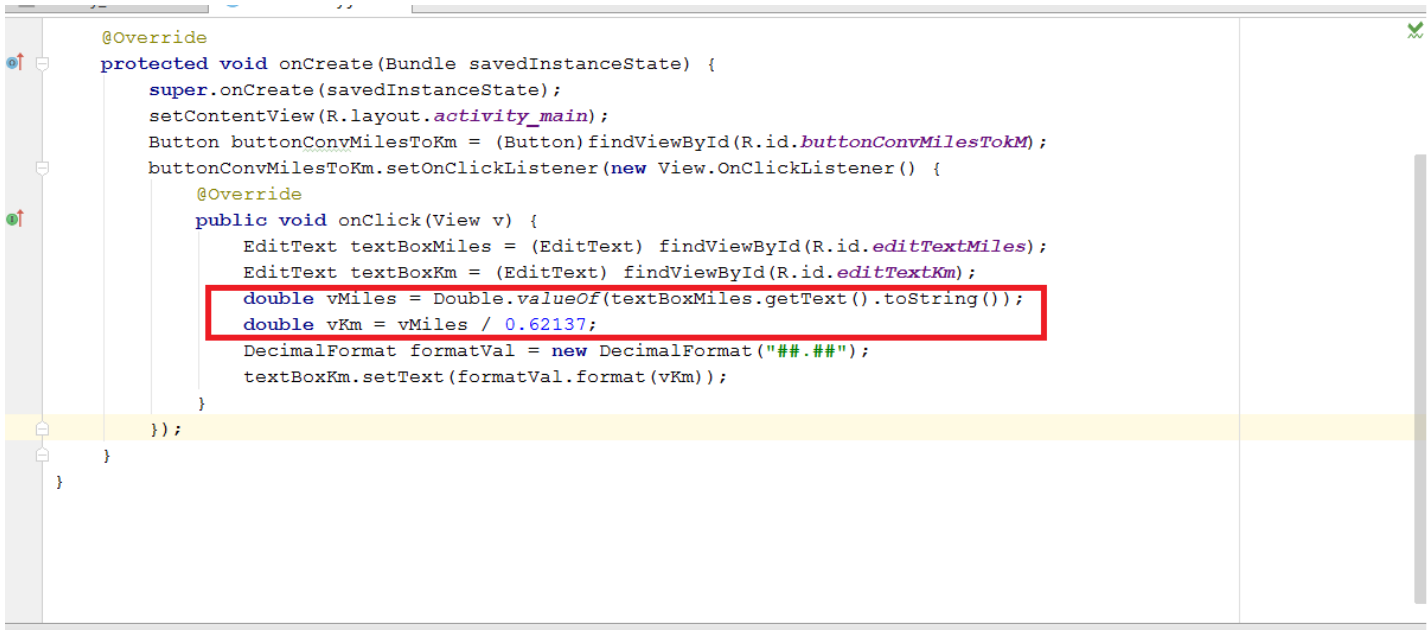
- if a button is to respond to a click event (the equivalent to the user touching and releasing the button view as though clicking on a physical button) it must both register the *View.onClickListener* event listener (via a call to the target view's *setOnClickListener()* method) and implement the corresponding *onClick()* callback method.
- In the event that a "click" event is detected on the screen at the location of the button view, the Android framework will call the *onClick()* method of that view when that event is removed from the event queue.
- It is within the implementation of the *onClick()* callback method that any tasks should be performed or other methods called in response to the button click.



- In the *OnClick()* callback method we code the instructions that allow to convert the miles value into kilometers value.
- To do this conversion, first a reference to the *EditTextMiles* component and the *EditTextKm* component are obtained with the following instructions :
  - **`EditText textBoxMiles = (EditText) findViewById(R.id.editTextMiles);`**
  - **`EditText textBoxKm = (EditText) findViewById(R.id.editTextKm);`**



- Then the value entered in the *EditTextMiles* component is converted to a double value with the following instruction :
  - **`double vMiles = Double.valueOf( textBoxMiles.getText().toString());`**
- The miles value is then converted into kilometers value with the instruction :
  - **`double vKm = vMiles / 0.62137;`**



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button buttonConvMilesToKm = (Button)findViewById(R.id.buttonConvMilesTokM);
    buttonConvMilesToKm.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText textBoxMiles = (EditText) findViewById(R.id.editTextMiles);
            EditText textBoxKm = (EditText) findViewById(R.id.editTextKm);
            double vMiles = Double.valueOf(textBoxMiles.getText().toString());
            double vKm = vMiles / 0.62137;
            DecimalFormat formatVal = new DecimalFormat("##.##");
            textBoxKm.setText(formatVal.format(vKm));
        }
    });
}
}
```

- Finally the kilometers value is displayed in the *EditTextKm* component with the following instructions :
  - **`DecimalFormat formatVal = new DecimalFormat("##.##");`**
  - **`textBoxKm.setText( formatVal.format(vKm));`**

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button buttonConvMilesToKm = (Button)findViewById(R.id.buttonConvMilesTokM);
        buttonConvMilesToKm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText textBoxMiles = (EditText) findViewById(R.id.editTextMiles);
                EditText textBoxKm = (EditText) findViewById(R.id.editTextKm);
                double vMiles = Double.valueOf(textBoxMiles.getText().toString());
                double vKm = vMiles / 0.62137;
                DecimalFormat formatVal = new DecimalFormat("##.##");
                textBoxKm.setText(formatVal.format(vKm));
            }
        });
    }
}
```

```
            public void onClick(View v) {
                EditText textBoxMiles = (EditText) findViewById(R.id.editTextMiles);
                EditText textBoxKm = (EditText) findViewById(R.id.editTextKm);
                double vMiles = Double.valueOf(textBoxMiles.getText().toString());
                double vKm = vMiles / 0.62137;
                DecimalFormat formatVal = new DecimalFormat("##.##");
                textBoxKm.setText(formatVal.format(vKm));
            }
        });
        Button buttonConvKmToMiles = (Button)findViewById(R.id.buttonConvKmToMiles);
        buttonConvKmToMiles.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText textBoxMiles = (EditText) findViewById(R.id.editTextMiles);
                EditText textBoxKm = (EditText) findViewById(R.id.editTextKm);
                double vKm = Double.valueOf(textBoxKm.getText().toString());
                double vMiles = vKm * 0.62137;
                DecimalFormat formatVal = new DecimalFormat("##.##");
                textBoxMiles.setText(formatVal.format(vMiles));
            }
        });
    }
}
```

- To convert kilometers into miles we have to code the *OnClick()* callback method of the button *buttonConcKmToMiles.*

# Testing the application on a Virtual Device (AVD)

- Now, we test the application on the Android Virtual Device emulator.
- Once the application is launched on the emulator, we can enter a value in the *EditTextMiles* component and click on the button « *Convert Miles to Km* ». The result is displayed in the *EditTextKm* component.