Universiti Malaysia PAHANG
Engineering • Technology • Creativity

BCS3283-Mobile Application Development

Chapter 8
ListView and Embedded Database (SQLite)

Editor
Dr. Mohammed Falah Mohammed

Faculty of Computer Systems & Software Engineering
falah@ump.edu.my

Embedded Database

- Aims

  To learn how to create a ListView and how to build an embedded database, add, edit, delete information using SQLite in Android Studio.

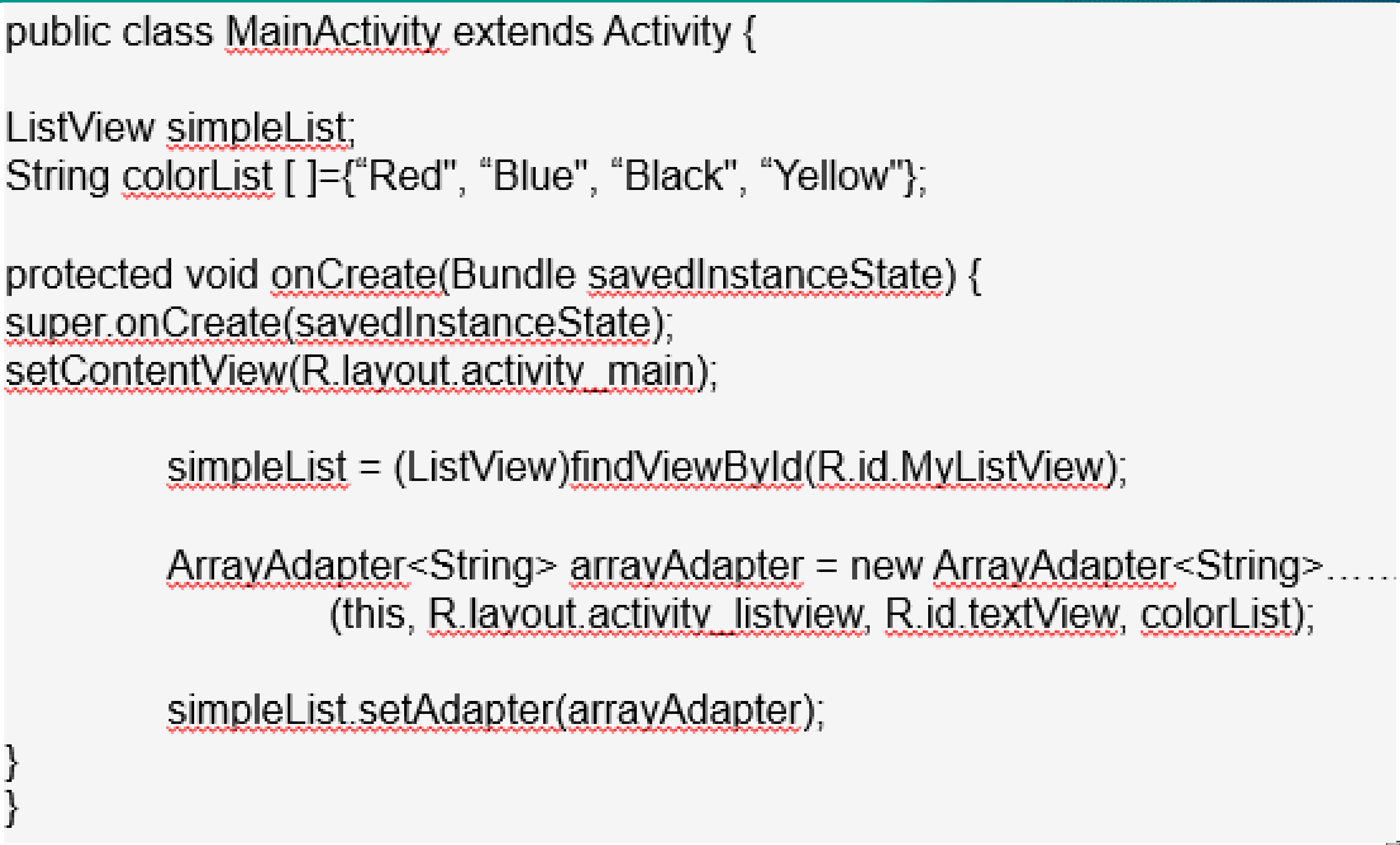
- Expected Outcomes
  - Understanding the concept of ListView and how to create it
  - Understanding the concept of embedded database
  - What are the important classes in SQLite
  - How to use SQLite in Android Apps

- References
  - https://abhiandroid.com/ui/listview
  - http://www.techotopia.com/
  - https://github.com/mitchtabian/SaveReadWriteDeleteSQLite

ListView

- When programmer need to display a data in the form of a scrollable list, ListView can be used to displayed a List of scrollable items in Android.
- ListView helps users to select any list item by clicking on it to do specific action.
- Adapter uses to fill data in a ListView. Where Adapter pulls the items content from a source such as an array or database and converts them into a view then placed into the list.

Example of ListView.Java



```
public class MainActivity extends Activity {

ListView simpleList;
String colorList [ ]={“Red", “Blue", “Black", “Yellow"};

protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

            simpleList = (ListView)findViewById(R.id.MyListView);

            ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>……
                        (this, R.layout.activity_listview, R.id.textView, colorList);

            simpleList.setAdapter(arrayAdapter);
}
}
```

INTRODUCTION TO SQLite

- Most applications need to store at lest part of information of App’s data.
- A comprehensive data storage strategy is a key factor in the design to avoid data loss.
- Data loss happens when Android runtime system terminates applications component to free up resources.
- SQLite database management system is bundled with the Android operating system

INTRODUCTION TO SQLite

- SQLite named as embedded database because it is forming as a library linked into applications.
- In this case, there will be no standalone database server running in the background.
- All operations of the database are handled internally within the application and a high-level language (Structured Query Language) uses for data accessing.

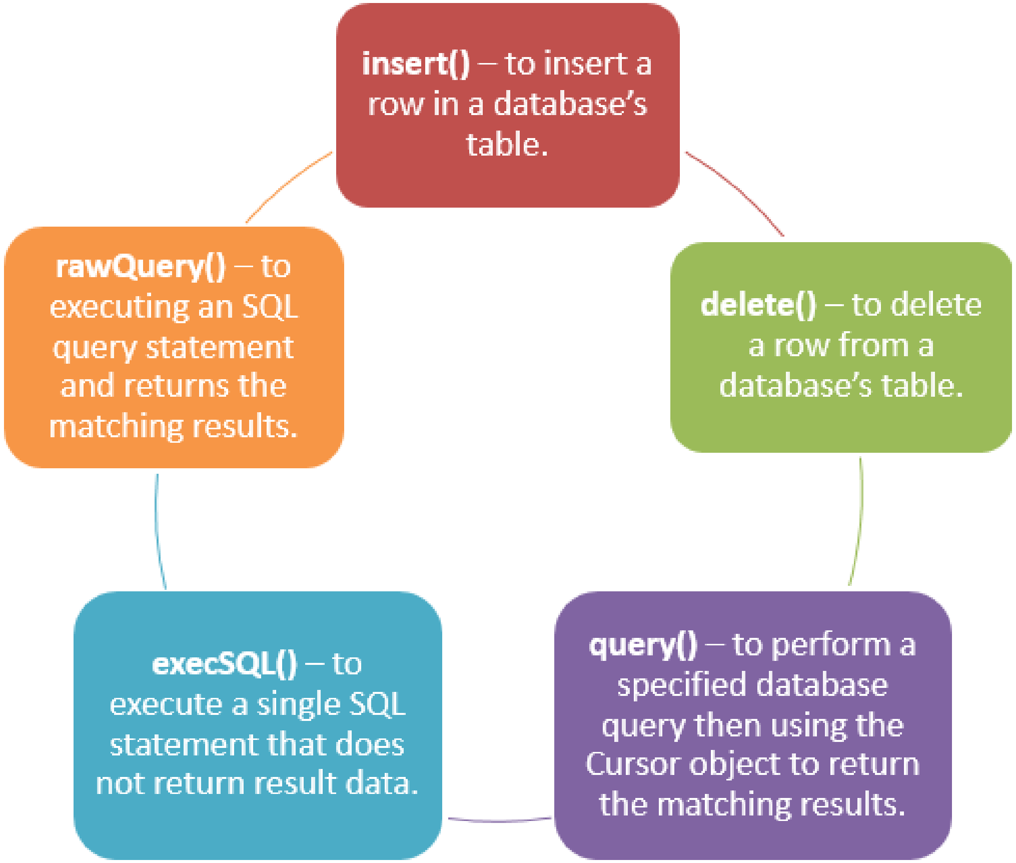
Android SQLite Java Classes

- The C programming language is used to write the SQLite while Android applications are primarily developed using Java.

- The Android SDK includes a set of classes are used to bridge this “language gap”.

Android SQLite Java Classes

Major classes within this category:



insert() – to insert a row in a database's table.

delete() – to delete a row from a database's table.

query() – to perform a specified database query then using the Cursor object to return the matching results.

execSQL() – to execute a single SQL statement that does not return result data.

rawQuery() – to executing an SQL query statement and returns the matching results.

Android SQLite Java Classes: **SQLiteOpenHelper**

A helper class is designed to manage the created database and update it.

The following callback methods implement within that subclass:

(MANDATORY)

onCreate() – is a method calls when we need to create a database for the first time, i.e., initializing the database by creating a table and inserting any initial data rows.

onUpgrade() – is a method used when we need to update a database. i.e., an application need to be updated on the device and that requires handling additional data storage by updating the app's database schema.

Android SQLite Java Classes: **SQLiteOpenHelper**

(POPULAR)

getWritableDatabase(): Reading and writing from opened or created database.

getReadableDatabase(): Reading only from created or opened database.

close() : used to close the database.

Android SQLite Java Classes: **Cursor**

This class is provided to enable user access to the database query results and the following chart represents the class key methods:

close() – Releasing all resources used by the cursor then closes it.

getCount() – Returning the rows number in the result set.

moveToFirst() – Moving to the first row in the result set.

moveToLast() – Moving to the last row in the result set.

moveToNext() – Moving to the next row in the result set.

move() – Moving from the current position in the result set by a specified offset .

get<type>() – Returning the value of the specified <type> contained at the specified column index of the row.

Implementing the SQLite

We are going to creating DB, Table, adding and deleting data from an SQLite database

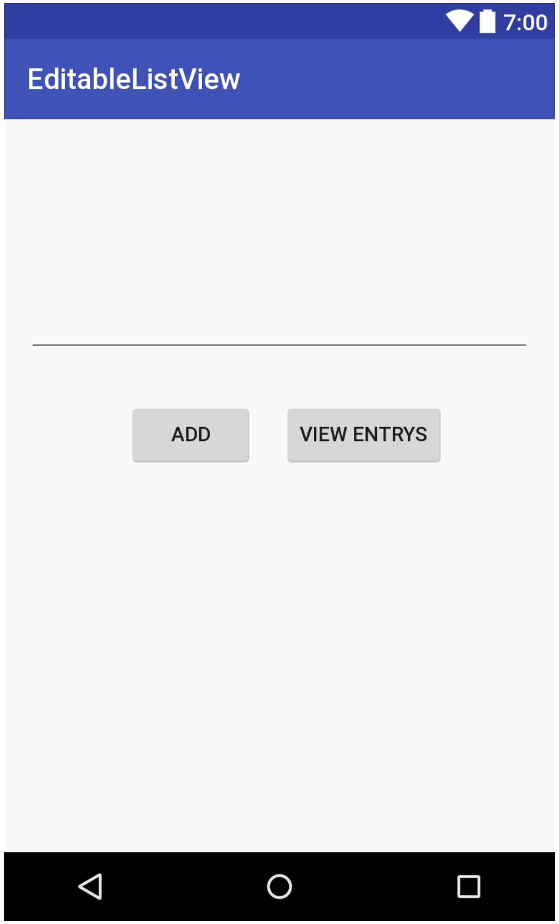
The application consists of three Activities

- MainActivity
- DatabaseHelper
- ViewListContents

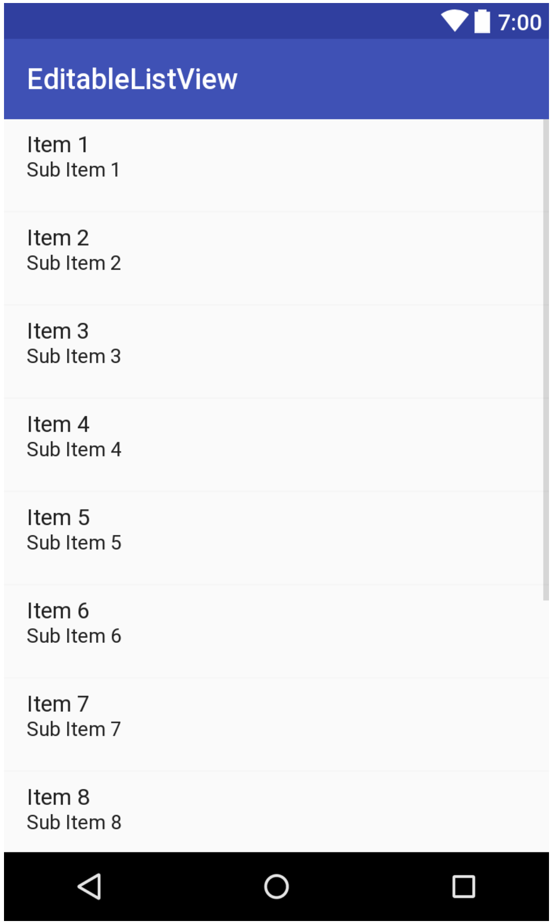
The Source code description can be found on the following links:

- https://www.youtube.com/watch?v=nY2bYJyGty8
- https://github.com/mitchtabian/SaveReadWriteDeleteSQLite

EditableListView App



MainActivity GUI



Item 1
Sub Item 1

Item 2
Sub Item 2

Item 3
Sub Item 3

Item 4
Sub Item 4

Item 5
Sub Item 5

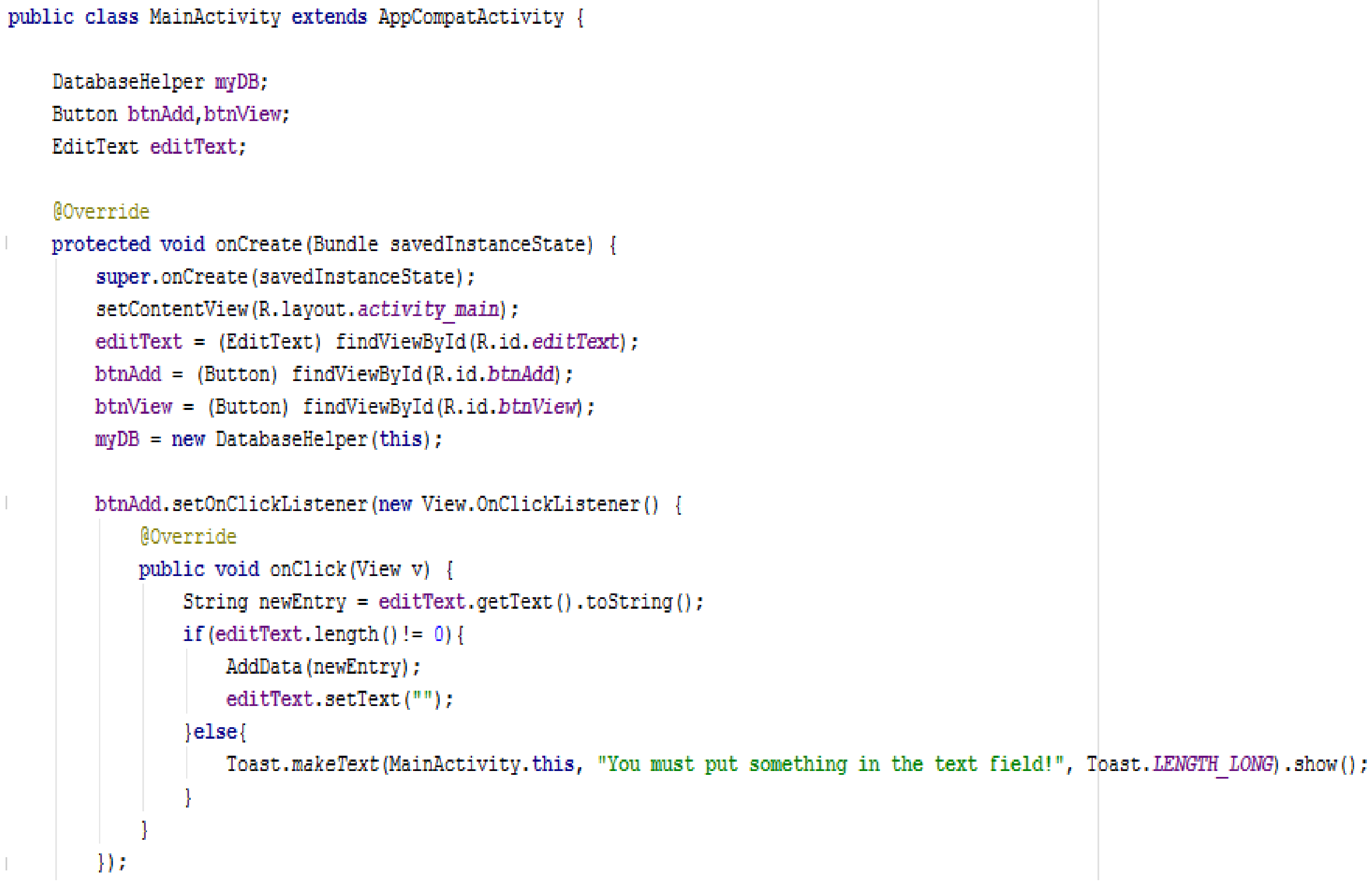
Item 6
Sub Item 6

Item 7
Sub Item 7

Item 8
Sub Item 8

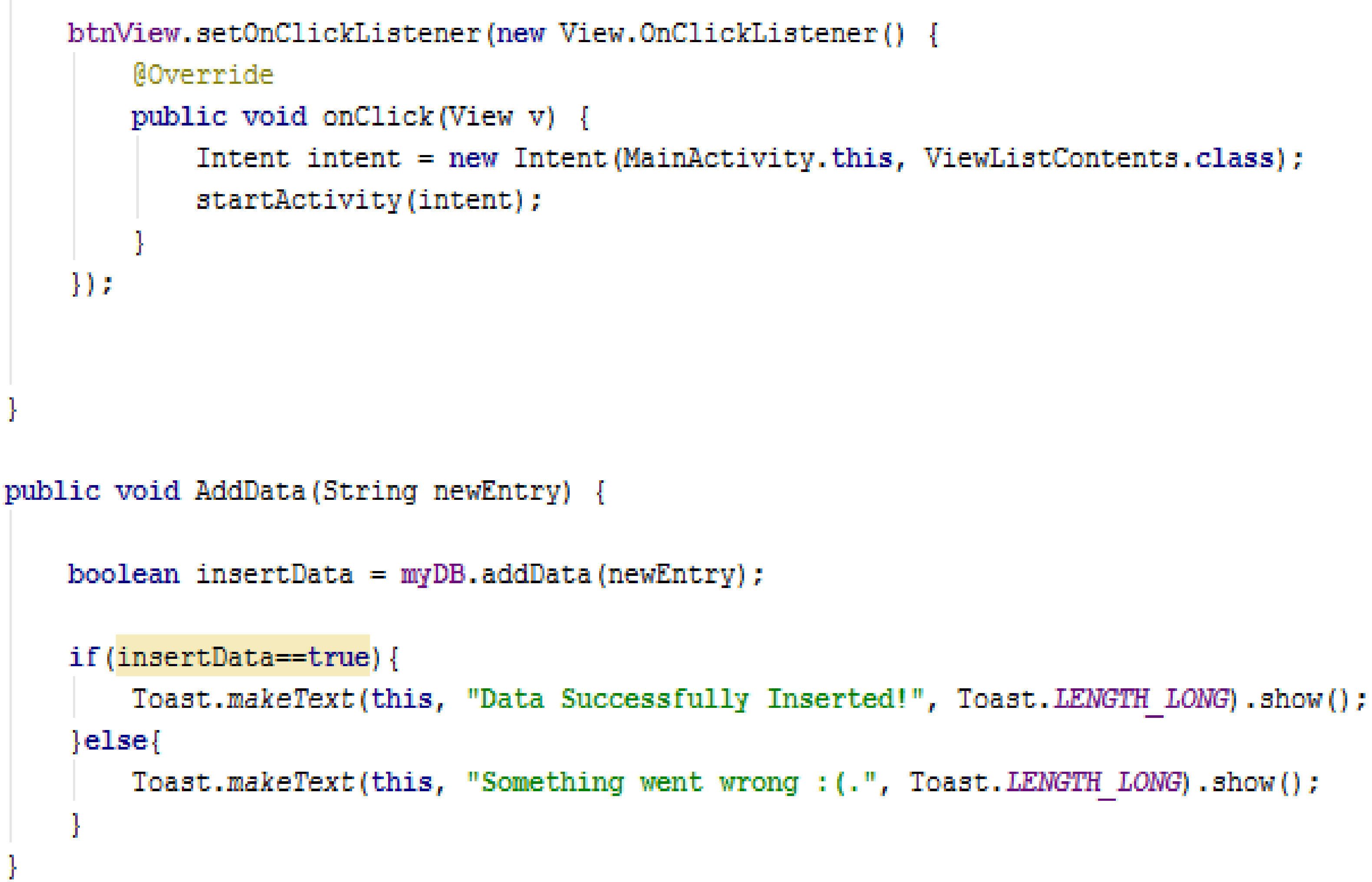
ViewListContents Activity GUI

MainActivity



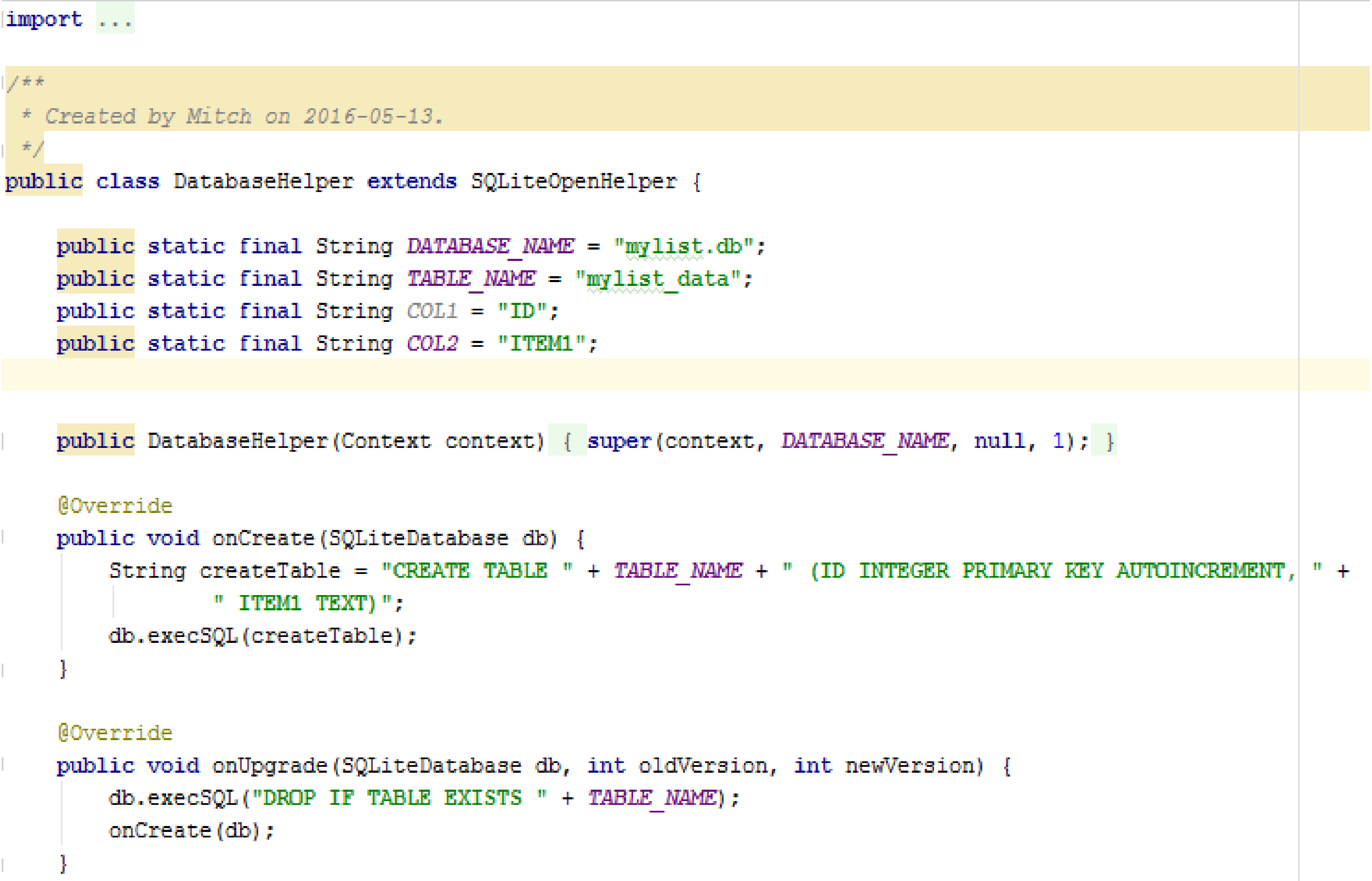
```
public class MainActivity extends AppCompatActivity {

    DatabaseHelper myDB;
    Button btnAdd,btnView;
    EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = (EditText) findViewById(R.id.editText);
        btnAdd = (Button) findViewById(R.id.btnAdd);
        btnView = (Button) findViewById(R.id.btnView);
        myDB = new DatabaseHelper(this);

        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String newEntry = editText.getText().toString();
                if(editText.length()!= 0){
                    AddData(newEntry);
                    editText.setText("");
                }else{
                    Toast.makeText(MainActivity.this, "You must put something in the text field!", Toast.LENGTH_LONG).show();
                }
            }
        });
```

MainActivity



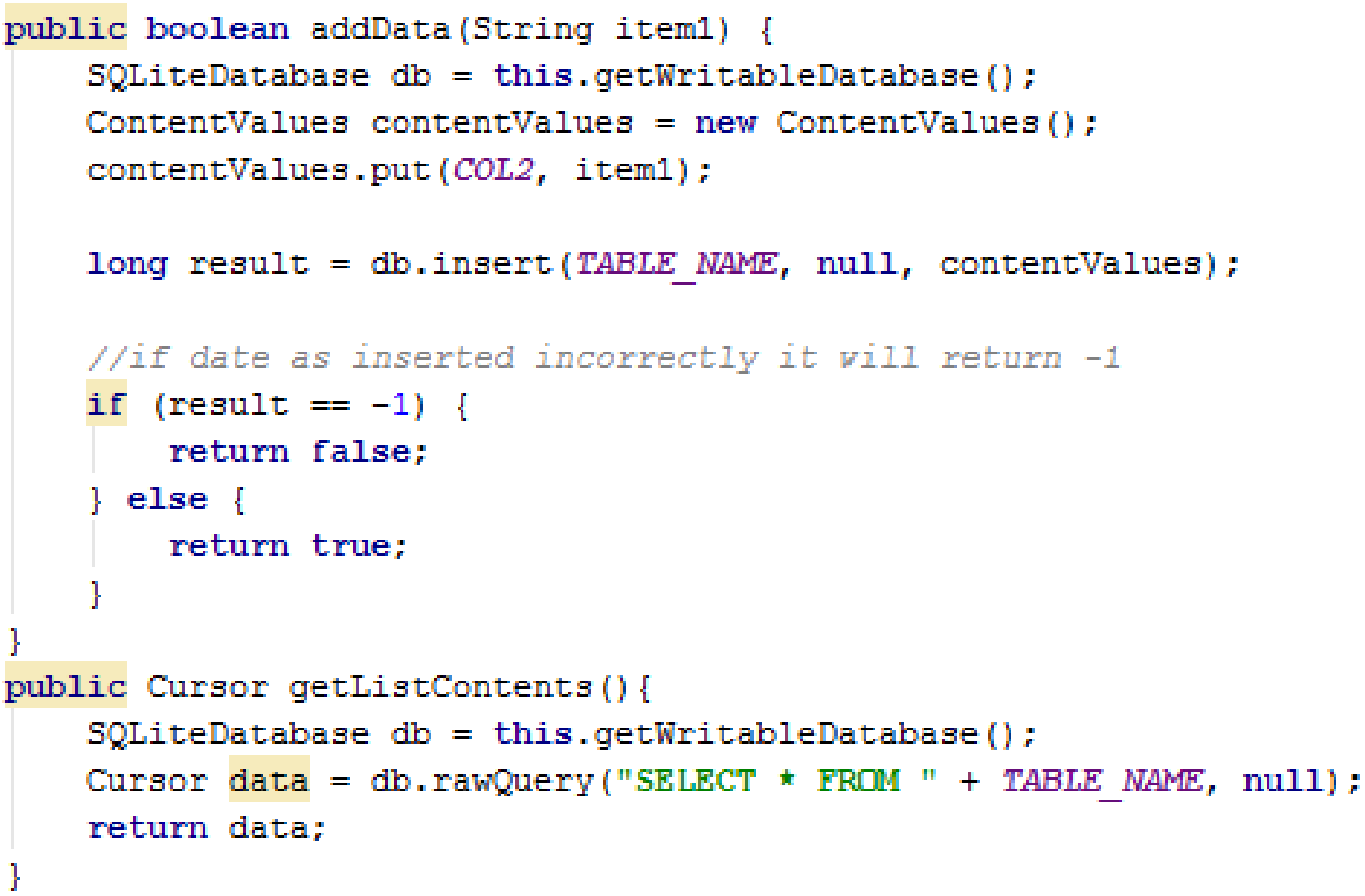
```
    btnView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, ViewListContents.class);
            startActivity(intent);
        }
    });


}

public void AddData(String newEntry) {

    boolean insertData = myDB.addData(newEntry);

    if(insertData==true){
        Toast.makeText(this, "Data Successfully Inserted!", Toast.LENGTH_LONG).show();
    }else{
        Toast.makeText(this, "Something went wrong :(.", Toast.LENGTH_LONG).show();
    }
}
```

DatabaseHelper Activity



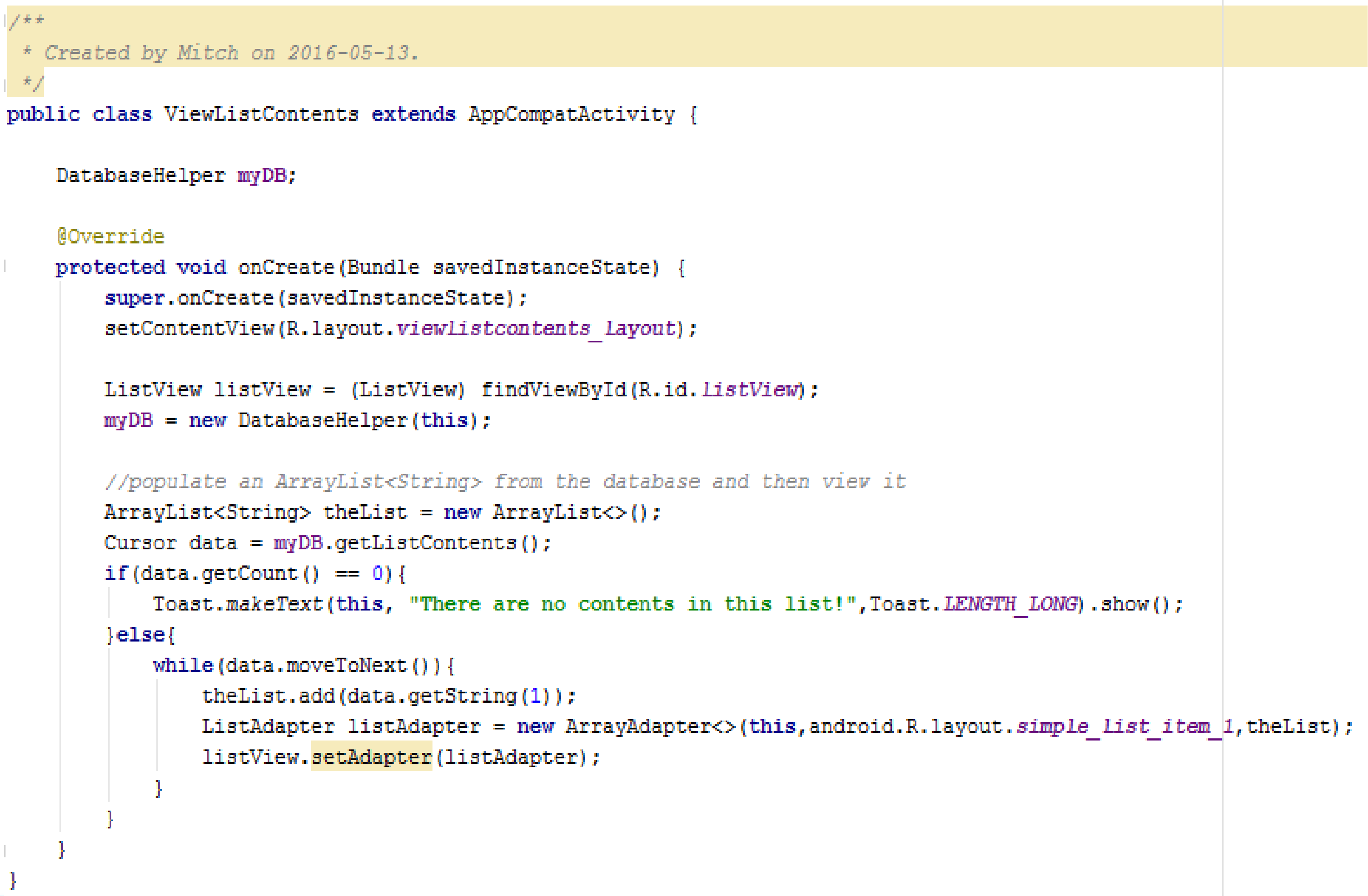
```
import ...

/**
 * Created by Mitch on 2016-05-13.
 */
public class DatabaseHelper extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "mylist.db";
    public static final String TABLE_NAME = "mylist_data";
    public static final String COL1 = "ID";
    public static final String COL2 = "ITEM1";


    public DatabaseHelper(Context context) { super(context, DATABASE_NAME, null, 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + " (ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                " ITEM1 TEXT)";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP IF TABLE EXISTS " + TABLE_NAME);
        onCreate(db);
    }
```

DatabaseHelper Activity



```
public boolean addData(String item1) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL2, item1);

    long result = db.insert(TABLE_NAME, null, contentValues);

    //if date as inserted incorrectly it will return -1
    if (result == -1) {
        return false;
    } else {
        return true;
    }
}
public Cursor getListContents(){
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor data = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    return data;
}
```

ViewListContents Activity



```
/**
 * Created by Mitch on 2016-05-13.
 */
public class ViewListContents extends AppCompatActivity {

    DatabaseHelper myDB;

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

        ListView listView = (ListView) findViewById(R.id.listView);
        myDB = new DatabaseHelper(this);

        //populate an ArrayList<String> from the database and then view it
        ArrayList<String> theList = new ArrayList<>();
        Cursor data = myDB.getListContents();
        if(data.getCount() == 0){
            Toast.makeText(this, "There are no contents in this list!",Toast.LENGTH_LONG).show();
        }else{
            while(data.moveToNext()){
                theList.add(data.getString(1));
                ListAdapter listAdapter = new ArrayAdapter<>(this,android.R.layout.simple_list_item_1,theList);
                listView.setAdapter(listAdapter);
            }
        }
    }
}
```