

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего
образования «Самарский национальный исследовательский университет имени академика
С.П. Королева»
(Самарский университет)

Естественнонаучный институт
Механико-математический факультет
Кафедра информатики и вычислительной математики

ОТЧЕТ ПО ПРАКТИКЕ

Вид практики _____ учебная _____

Тип практики технологическая (проектно-технологическая) практика

Сроки прохождения практики: с 24.06.2024 по 6.07.2024 г.

по направлению подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем
(уровень бакалавриата)

направленность (профиль) «Разработка и администрирование
информационных систем»

Обучающийся группы №4345-020303D

А.В.Архипова

Руководитель практики
от университета старший преподаватель

М.В.Морозова

Дата сдачи 24.06.2024 г.

Дата защиты 6.07.2024 г.

Оценка _____

Самара 2024

СОДЕРЖАНИЕ

ОТЧЕТ ПО ПРАКТИКЕ.....	1
СОДЕРЖАНИЕ	2
ОПИСАТЕЛЬНАЯ ЧАСТЬ	5
1.1 Постановка задачи	5
1.2 Решение задачи	5
1.3 Тестирование приложения.....	7
ВЫВОДЫ	10
ПРИЛОЖЕНИЕ А.....	12
ПРИЛОЖЕНИЕ Б	18
ПРИЛОЖЕНИЕ В.....	19
ПРИЛОЖЕНИЕ Г	20
ПРИЛОЖЕНИЕ Д.....	27

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»
(Самарский университет)

Естественнонаучный институт
Механико-математический факультет
Кафедра информатики и вычислительной математики

Задание по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью

Обучающемуся группы 4345-020303D Архиповой Арине Вячеславовне
Направлен на практику приказом по университету от 19.06.2024 №321-ПР
в Самарский университет, лаборатория администрирования информационных систем
(наименование профильной организации или структурного подразделения университета)

Планируемые результаты освоения образовательной программы	Выполнение определенных видов работ, связанных с будущей профессиональной деятельностью	Результаты практики
ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	Знакомство с методами и способами разработки программ с микросервисной архитектурой, а также оценкой качества программных продуктов.	Знает основные способы применения современного математического аппарата при реализации и оценке качества программных продуктов в различных областях человеческой деятельности.
ОПК-2.2 Применяет современный математический аппарат при реализации и оценке качества программных продуктов и программных комплексов в различных областях человеческой деятельности	Знакомство с инструментами для создания программных продуктов: GitLab, Maven,. Реализация основных алгоритмов, необходимых для функционирования разрабатываемого сервиса на языке Java. Проектирование структуры проекта Android-приложения, реализация и проведение оценки качества Настройка интеграции созданного сервиса с СУБД PostgreSQL. Тестирование разработанного сервиса с использованием Mockito.	Умеет использовать современный математический аппарат при реализации и оценке качества программных продуктов. Имеет навыки использования современного математического аппарата при реализации и оценке качества программных продуктов.

Срок представления на кафедру отчета о практике 6.07.2024 г.

Руководитель практики от университета,

Старший преподаватель

(подпись) М.В.Морозова

Задание принял к исполнению
обучающийся группы № 4345-020303D

(подпись) А.В.Архипова

ОПИСАТЕЛЬНАЯ ЧАСТЬ

1.1 Постановка задачи

Разработать мобильное приложение «Калькулятор цены компьютера» с учётом цен на комплектующие.

1.2 Решение задачи

В качестве среды разработки использовалась Android Studio 2024.1.1. Android Studio - это интегрированная среда разработки (IDE) для создания приложений на платформе Android. Ее основное назначение заключается в обеспечении удобной и эффективной среды для написания кода, отладки и тестирования приложений для устройств на базе Android. Она предлагает разработчикам широкий спектр инструментов, включая визуальные редакторы интерфейсов, системы сборки и управления версиями, что значительно упрощает процесс создания приложений. Кроме того, Android Studio обеспечивает интеграцию с Android SDK, что позволяет быстро получить доступ к последним API и инструментам, необходимым для создания приложений.

Структура разработанного приложения представлена на рисунке 1.

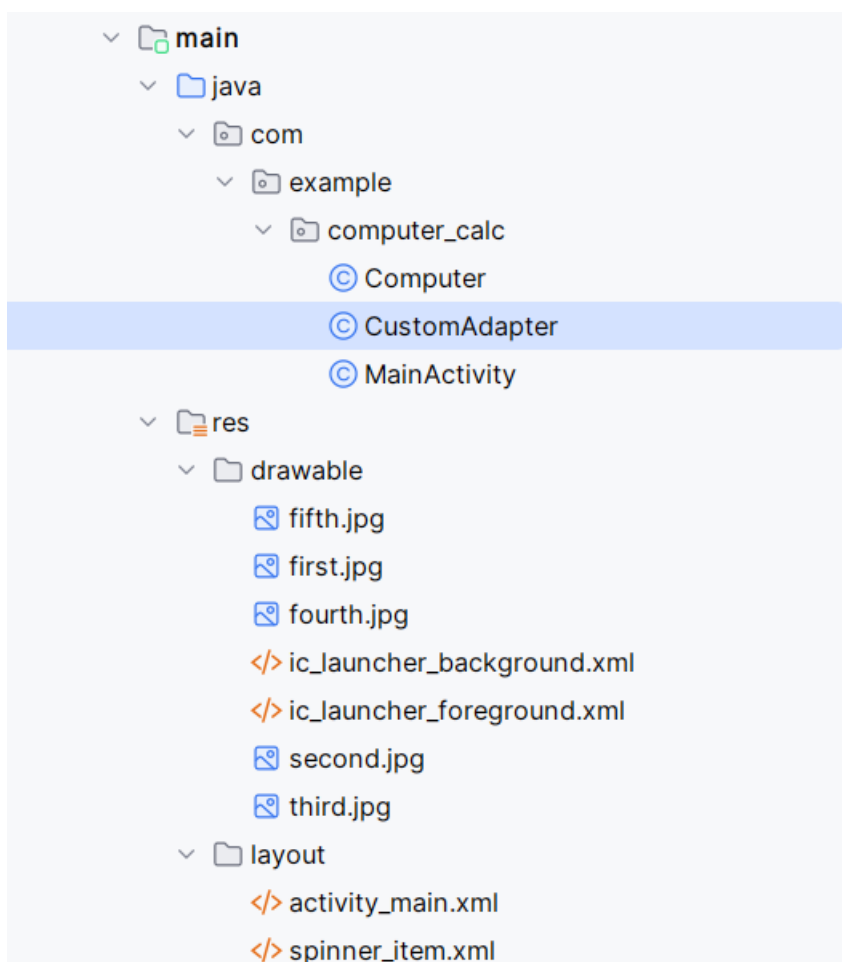


Рисунок 1 – Структура приложения

После применения принципа абстракции ООП к предметной области, были выделены и объединены в отдельный класс цены таких комплектующих компьютера, как процессора, материнской платы, оперативной памяти и периферийных устройств. Разработанный класс `Computer` представлен в приложении Б. В нём был реализован конструктор по умолчанию, а также сгенерированы сеттеры для полей. Суммарная стоимость компьютера с учётом цен на его комплектующие подсчитывается с помощью метода `getResult()`.

По умолчанию выполнение приложения Android начинается с класса `MainActivity`, представленного в приложении А. В нём описано поведение приложения при выборе комплектующих компьютера.

Выбор процессора осуществляется с помощью выпадающего списка `Spinner`, каждый элемент которого содержит изображение и название процессора, описанные в массивах `names` и `images`. Картинки для массива `images` размещены в каталоге `drawable` ресурсов приложения. Для корректного отображения выпадающего списка был реализован адаптер, описанный в приложении В, внешний вид элементов представлен в приложении Г.2. Выбор типа материнской платы осуществляется с помощью переключателей `RadioButton`, объединённых в группу `RadioGroup`. Выбор объёма оперативной памяти производится с помощью слайдера `SeekBar`. Включение в стоимость периферийных устройств определяется с помощью флажка `CheckBox`.

Обработка выбора компонентов производится с помощью слушателей событий – объектов, уведомляемых о некотором событии, произошедшем с подключенным к нему источником. При выборе компонента, слушатели `proc_listener`, `motherboard_listener`, `mem_listener` и `per_listener` получают информацию о событии, определяют цену комплектующей, а затем передают её и индикатор типа комплектующей в метод `setPrice`.

Метод `setPrice` передаёт объекту класса `Computer` цену соответствующего компонента, а также отображает её на экране приложения и вызывает метод `updRes` для расчёта и обновления общей стоимости компьютера.

Переопределённый метод `onCreate` вызывается при создании или перезапуске активности. В нём происходит задание внешнего вида приложения с помощью метода `setContentView`, а также инициализация полей класса, подключение адаптера и слушателей событий к соответствующим элементам. Внешний вид описан в приложении Г.1 с использованием строковых ресурсов, описанных в приложении Г.3.

На рисунке 2 представлены результаты работы приложения.



Рисунок 2 – Результат работы приложения

1.3 Тестирование приложения

Проверка работоспособности программы была проведена с помощью ручного тестирования.

В процессе тестирования приложения были применены такие фреймворки как Mockito и Espresso. Это два популярных фреймворка, используемых для тестирования приложений на платформе Android. Mockito предоставляет возможности для создания и использования макетов в тестах. Макеты позволяют заменить реальные объекты в приложении фиктивными объектами, что упрощает изоляцию и тестирование отдельных компонентов приложения. Mockito также обеспечивает удобные средства для создания заглушек (stubs) и проверки вызовов методов. Espresso представляет собой фреймворк для написания UI-тестов в приложениях Android. Он позволяет создавать автоматизированные тесты пользовательского интерфейса, взаимодействуя с элементами пользовательского

интерфейса.

Перед началом тестирования необходимо подключить следующие зависимости:

```
dependencies {  
    implementation(libs.appcompat)  
    implementation(libs.material)  
    implementation(libs.activity)  
    implementation(libs.constraintlayout)  
    implementation(libs.firebase.sessions)  
    androidTestImplementation(libs.ext.junit)  
    testImplementation(libs.junit.v412)  
    testImplementation(libs.mockito.core)  
    androidTestImplementation("com.android.support:support-annotations:24.0.0")  
    androidTestImplementation("com.android.support.test:runner:0.5")  
    androidTestImplementation("com.android.support.test:rules:0.5")  
    androidTestImplementation("com.android.support.test.espresso:espresso-core:2.2.2")  
}
```

А также настроить параметр gradle:

```
testInstrumentationRunner = "android.support.test.runner.AndroidJUnitRunner"
```

Было написано два юнит-теста, проверяющие класс Computer. Юнит-тесты представлены в приложении Д.1. В них с помощью методов assertEquals и assertEquals были сравнены с ожидаемыми результаты работы конструктора класса и его методов, а также была произведена проверка на соответствие типов. Результат проведённого теста представлен на рисунке 3.

В инструментальном тесте, представленном в приложении Д.2, проверяются ожидаемые результаты работы приложения. Для этого с переменная типа ActivityTestRule. Использование ActivityTestRule позволяет автоматически управлять жизненным циклом активности во время выполнения тестов. Это включает её создание перед запуском теста и завершение после его выполнения. Затем с помощью методов onView и onData был произведён поиск компонентов для совершения над ними действий с помощью метода perform. После этого произвелась проверка соответствия ожидаемому результату с помощью методов matches и withText. Для установки слайдера в желаемое положение был также реализован метод setProgress. Результат проведённого теста представлен на рисунке 4.

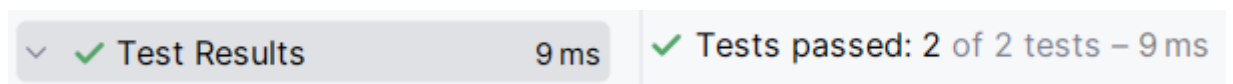


Рисунок 3 – Результаты локальных тестов

Status	<div><div></div></div>	1 passed	1 tests, 1 m 32 s 76 ms
Filter tests:	<div><div>✓</div><div>⊗</div><div>↕</div><div>✕</div><div>↑</div><div>↓</div><div>🕒</div><div>📄</div><div>🔗</div></div>		
Tests		Duration	INFINIX Infinix X6831
✓ Test Results		3 s	1/1
✓ MainActivityTest		3 s	1/1
✓ TestCase		3 s	✓

Рисунок 4 – Результат инструментального теста

В результате проведённого тестирования не было обнаружено отклонений от ожидаемой работы приложения.

ВЫВОДЫ

В процессе прохождения технологической (проектно-технологической) практики:

- обобщена и систематизирована информация о технологии разработки программ с микросервисной архитектурой;
- обобщена и систематизирована информация о технологии разработки мобильных приложений;
- были изучены инструменты для создания мобильных приложений: Android Studio;
- спроектировано, реализовано на языке Java и протестировано приложение, которое реализует калькулятор цены компьютера.

Таким образом, в ходе выполнения индивидуального задания технологической (проектно-технологической) практики были сформированы систематические знания основных технологий программирования; основных методов и средств автоматизации проектирования, создания, тестирования и оценки качества программных продуктов; основных концептуальных положений объектно-ориентированного и визуального программирования, методов, способов и средств разработки программ в рамках этих направлений. Сформированы умения выбирать технологию программирования, позволяющую решить поставленную задачу с учетом имеющихся ограничений; умения использовать методы и средства автоматизации проектирования, создания тестирования и оценки качества программных продуктов; умения использовать методы, способы и средства разработки программ в рамках объектно-ориентированного и визуального программирования при решении практических задач. Также приобретены навыки использования технологий программирования, позволяющих решить поставленную задачу с учетом имеющихся ограничений, навыки проектирования, реализации, испытания и оценки качества программных продуктов, навыки разработки программ в рамках объектно-ориентированного и визуального программирования что свидетельствует о том, что компетенции ОПК-2 и ОПК-2.2 освоены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Что такое Android Studio и как ей пользоваться [Электронный ресурс]. URL: <https://skillbox.ru/media/code/hto-takoe-android-studio-i-kak-ey-polzovatsya/> (Дата обращения: 24.06.2024)
- 2 Изучение Java для разработки приложений для Android – полное руководство [Электронный ресурс] / translated.turbopages.org – 2021. URL: https://translated.turbopages.org/proxy_u/en-ru.ru.3bb439c0-64a44189-7b934c03-74722d776562/https/www.geeksforgeeks.org/learn-java-for-android-app-development-a-complete-guide/ (Дата обращения 05.07.2024)
- 3 Android Developers [Электронный ресурс]. URL: <https://developer.android.com> (дата обращения 3.07.2024)
- 4 Тестирование активности в Android Studio / Testing activity in Android Studio [Электронный ресурс]. URL: <https://evgenii.com/blog/testing-activity-in-android-studio-tutorial-part-2/> (Дата обращения 06.07.2024)
- 5 Тестирование приложения [Электронный ресурс]. URL: <https://developer.alexanderklimov.ru/android/debug/testing.php> (Дата обращения 4.07.2024)

ПРИЛОЖЕНИЕ А

Листинг файла MainActivity.java, реализующего основное поведение приложения.

```
package com.example.computer_calc;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.SeekBar;
import android.widget.Spinner;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    //Объявление полей класса
    TextView procPriceLabel, motherboardPriceLabel, memoryPriceLabel, periferalPriceLabel,
resultLabel;
    Computer c;
    Spinner sp1;
    CheckBox cb;
    RadioButton am4, am5, lga1151, lga1200, lga1700;
    SeekBar sb;
    CustomAdapter adapter;
    String price;
    String[] names = {"Процессор AMD A6-7480 OEM [FM2+, 2 x 3.5 ГГц, L2 - 1 МБ, 2 x
DDR3-2133 МГц, AMD Radeon R5, TDP 65 Вт]",
        "Процессор AMD FX-4300 BOX [AM3+, 4 x 3.8 ГГц, L2 - 4 МБ, L3 - 4 МБ, 2 x DDR3-
1866 МГц, TDP 95 Вт, кулер]",
        "Процессор AMD Athlon 200GE OEM [AM4, 2 x 3.2 ГГц, L2 - 1 МБ, L3 - 4 МБ, 2 x
DDR4-2667 МГц, AMD Radeon Vega 3, TDP 35 Вт]",
        "Процессор AMD Athlon 3000G OEM [AM4, 2 x 3.5 ГГц, L2 - 1 МБ, L3 - 4 МБ, 2 x
DDR4-2666 МГц, AMD Radeon Vega 3, TDP 35 Вт]",
        "Процессор Intel Celeron G5905 OEM [LGA 1200, 2 x 3.5 ГГц, L2 - 0.5 МБ, L3 - 4 МБ,
2 x DDR4-2666 МГц, Intel UHD Graphics 610, TDP 58 Вт]"};
```

```

int[] images = {R.drawable.first,
                R.drawable.second,
                R.drawable.third,
                R.drawable.fourth,
                R.drawable.fifth};

float[] prices = {1199, 1850, 3599, 4499, 4499};

//Переопределение слушателей
//Для выпадающего списка
AdapterView.OnItemClickListener proc_listener = new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        setPrice(prices[i], 1);
    }
    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {}
};

//Для переключателя
View.OnClickListener motherboard_listener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        RadioButton rb = (RadioButton) v;
        float p = 0;
        if (rb.getId() == R.id.am4) {
            p = 5999;
        } else if (rb.getId() == R.id.am5) {
            p = 9899;
        } else if (rb.getId() == R.id.lga_1151) {
            p = 6999;
        } else if (rb.getId() == R.id.lga_1200) {
            p = 5799;
        } else if (rb.getId() == R.id.lga_1700) {
            p = 6699;
        }
    }
}

```

```

        setPrice(p, 2);
    }
};

//Для слайдера
SeekBar.OnSeekBarChangeListener mem_listener = new SeekBar.OnSeekBarChangeListener()
{
    @Override
    public void onProgressChanged(SearchBar seekBar, int i, boolean b) {
        int selected = seekBar.getProgress();
        int selected1 = i;
        double p = 0;
        if (selected1 == 0) {
            p = 1099;
        } else if (selected1 == 1) {
            p = 899;
        }
        else if (selected1 == 2) {
            p = 1499;
        }
        else if (selected1 == 3) {
            p = 3299;
        }
        else if (selected1 == 4) {
            p = 8099;
        }
        setPrice(p, 3);
    }
    @Override
    public void onStartTrackingTouch(SearchBar seekBar) {}
    @Override
    public void onStopTrackingTouch(SearchBar seekBar) {}
};

//Для флажка
View.OnClickListener per_listener = new View.OnClickListener() {
    @Override

```

```

public void onClick(View view) {
    double p = 0;
    if (((CheckBox) view).isChecked()) {
        p = 15000;
    }
    setPrice(p, 4);
}
};

```

//Переопределение метода onCreate

@Override

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //Инициализация полей
    c = new Computer();
    sp1 = findViewById(R.id.spinner);
    cb = findViewById(R.id.periferal);
    am4 = findViewById(R.id.am4);
    am5 = findViewById(R.id.am5);
    lga1151 = findViewById(R.id.lga_1151);
    lga1200 = findViewById(R.id.lga_1200);
    lga1700 = findViewById(R.id.lga_1700);
    sb = findViewById(R.id.seekbar);
    adapter = new CustomAdapter(this, names, images);
    procPriceLabel = findViewById(R.id.proc_price);
    motherboardPriceLabel = findViewById(R.id.motherboard_price);
    memoryPriceLabel = findViewById(R.id.mem_price);
    periferalPriceLabel = findViewById(R.id.per_price);
    resultLabel = findViewById(R.id.res);

    //Назначение адаптера и слушателей
    sp1.setAdapter(adapter);
    sp1.setOnItemClickListener(proc_listener);
    am4.setOnClickListener(motherboard_listener);
}

```

```

am5.setOnClickListener(motherboard_listener);
lga1151.setOnClickListener(motherboard_listener);
lga1200.setOnClickListener(motherboard_listener);
lga1700.setOnClickListener(motherboard_listener);
sb.setOnSeekBarChangeListener(mem_listener);
cb.setOnClickListener(per_listener);
}

//Вспомогательные методы
//Для установки значений и вывода цены
public void setPrice(double p, int i) {
    price = p + " руб.";
    switch (i) {
        case 1:
            procPriceLabel.setText(price);
            c.setProc(p);
            break;
        case 2:
            motherboardPriceLabel.setText(price);
            c.setMotherboard(p);
            break;
        case 3:
            memoryPriceLabel.setText(price);
            c.setMem(p);
            break;
        case 4:
            periferalPriceLabel.setText(price);
            c.setPer(p);
            break;
    }
    updRes();
}

//Для подсчёта и обновления результата
public void updRes() {
    String s = "Итоговая цена: " + c.getResult() + " рублей.";

```



```
resultLabel.setText(s);  
}
```

ПРИЛОЖЕНИЕ Б

Листинг файла Computer.java, содержащего класс «Компьютер», стоимость которого подсчитывается.

```
package com.example.computer_calc;

public class Computer {
    //Поля
    public double procPrice, motherboardPrice, memPrice, perPrice;
    //Конструктор
    Computer() {
        procPrice = 0;
        motherboardPrice = 0;
        memPrice = 1099;
        perPrice = 0;
    }
    //Сеттеры
    public void setProc(double p) {
        procPrice = p;
    }
    public void setMotherboard(double p) {
        motherboardPrice = p;
    }
    public void setMem(double p) {
        memPrice = p;
    }
    public void setPer(double p) {
        perPrice = p;
    }
    //Метод подсчёта результата
    public double getResult() {
        return procPrice + motherboardPrice + memPrice + perPrice;
    }
}
```

ПРИЛОЖЕНИЕ В

Листинг файла CustomAdapter.java - адаптера выпадающего списка.

```
package com.example.computer_calc;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;

public class CustomAdapter extends ArrayAdapter<String> {
    //Поля
    Context context;
    String[] names;
    int[] images;
    //Конструктор
    public CustomAdapter(@NonNull Context context, String[] names, int[] images) {
        super(context, R.layout.spinner_item, names);
        this.context = context;
        this.names = names;
        this.images = images;
    }
    //Переопределение методов
    // Отвечает за внешний вид выпадающего списка в развёрнутом состоянии
    @Override
    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = inflater.inflate(R.layout.spinner_item, null);
        TextView t1 = (TextView) row.findViewById(R.id.textView);
        ImageView i1 = (ImageView) row.findViewById(R.id.imageView);
        t1.setText(names[position]);
        i1.setImageResource(images[position]);
        return row;
    }
    //Отвечает за внешний вид выпадающего списка в свёрнутом состоянии
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = inflater.inflate(R.layout.spinner_item, null);
        TextView t1 = (TextView) row.findViewById(R.id.textView);
        ImageView i1 = (ImageView) row.findViewById(R.id.imageView);
        t1.setText(names[position]);
        i1.setImageResource(images[position]);
        return row;
    }
}
```

ПРИЛОЖЕНИЕ Г

Листинги файлов формата xml.

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:id="@+id/scroll"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content">

    <LinearLayout

        android:id="@+id/main"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        tools:context=".MainActivity"

        android:orientation="vertical">

        <TextView

            android:layout_height="wrap_content"

            android:layout_width="wrap_content"

            android:text="@string/your_config"/>

        <TextView

            android:layout_height="wrap_content"

            android:layout_width="wrap_content"

            android:text="@string/choose_proc"/>

        <Spinner

            android:id="@+id/spinner"

            android:layout_width="409dp"

            android:layout_height="wrap_content"
```

```

tools:layout_editor_absoluteX="5dp"

tools:layout_editor_absoluteY="40dp" />

<TextView

    android:id="@+id/proc_price"

    android:layout_height="match_parent"

    android:layout_width="wrap_content"

    android:padding="10dp"/>

<TextView

    android:layout_height="wrap_content"

    android:layout_width="wrap_content"

    android:text="@string/choose_mothe"/>

<RadioGroup

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/motherboard">

    <RadioButton

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="@string/lga_1700"

        android:id="@+id/lga_1700"/>

    <RadioButton

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="@string/am4"

        android:id="@+id/am4"/>

    <RadioButton

        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"

        android:text="@string/am5"

        android:id="@+id/am5"/>
    <RadioButton

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="@string/lga_1200"

        android:id="@+id/lga_1200"/>

    <RadioButton

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="@string/lga_1151"

        android:id="@+id/lga_1151"/>

</RadioGroup>

<TextView

        android:id="@+id/motherboard_price"

        android:layout_height="match_parent"

        android:layout_width="wrap_content"

        android:padding="10dp"/>

<TextView

        android:layout_height="wrap_content"

        android:layout_width="wrap_content"

        android:text="@string/mem_choose"/>

<SeekBar

        android:id="@+id/seekbar"

        style="@style/Widget.AppCompat.SeekBar.Discrete"

        android:layout_height="match_parent"

```

```
android:layout_width="300dp"
```

```
android:max="4"
```

```
android:progress="0"/>
```

```
<LinearLayout
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:orientation="horizontal">
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:paddingLeft="13dp"
```

```
        android:text="@string/_2"/>
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:paddingLeft="60dp"
```

```
        android:text="@string/_4"/>
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:paddingLeft="60dp"
```

```
        android:text="@string/_8"/>
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:paddingLeft="55dp"
```

```
        android:text="@string/_16"/>
```

```

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:paddingLeft="55dp"

    android:text="@string/_32"/>

</LinearLayout>

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/mem_price"

    android:text="@string/_1099_0"/>

<CheckBox

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Включить периферийные устройства в стоимость"

    android:id="@+id/periferal"/>

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/per_price"

    android:text="@string/_0"/>

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/res"/>

</LinearLayout>

</ScrollView>

```


Листинг Г.1 – Файл main_activity.xml, основной вид приложения

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="horizontal">

    <ImageView

        android:id="@+id/imageView"

        android:layout_width="0dp"

        android:layout_weight="1"

        android:layout_height="68dp"

        android:padding="10dp"

        android:src="@drawable/first" />

    <TextView

        android:id="@+id/textView"

        android:layout_width="0dp"

        android:layout_height="61dp"

        android:layout_weight="1"

        android:padding="10dp"

        android:text="@string/proc1" />

</LinearLayout>
```

Листинг Г.2 – Файл spinner_item.xml, элемент выпадающего списка

```
<resources>

    <string name="app_name">Computer_calc</string>

    <string name="proc1">Процессор AMD A6-7480 OEM [FM2+, 2 x 3.5 ГГц, L2 - 1 МБ, 2 x
```

```

DDR3-2133 МГц, AMD Radeon R5, TDP 65 Вт]</string>
<string name="your_config">Ваша конфигурация</string>
<string name="lga_1700">На сокете LGA 1700</string>
<string name="am4">На сокете AM4</string>
<string name="am5">На сокете AM5</string>
<string name="lga_1200">На сокете LGA 1200</string>
<string name="lga_1151">На сокете LGA 1151</string>
<string name="choose_proc">Выберите процессор</string>
<string name="choose_mothe">Выберите материнскую плату</string>
<string name="mem_choose">Выберите объем оперативной памяти (ГБ)</string>
<string name="_2">2</string>
<string name="_4">4</string>
<string name="_8">8</string>
<string name="_16">16</string>
<string name="_32">32</string>
<string name="_1099_0">1099.0 руб.</string>
<string name="_0">0.0 руб.</string>
</resources>

```

Листинг Г.3 – Файл strings.xml, строковые ресурсы приложения

ПРИЛОЖЕНИЕ Д

Тестирование приложения.

```
package com.example.computer_calc;

import org.junit.Test;

import static org.hamcrest.core.IsInstanceOf.instanceOf;
import static org.junit.Assert.*;

public class ComputerTest {
    Computer c = new Computer();

    @Test
    public void createComputer() {
        //Проверяем типы полей
        assertThat(c.procPrice, instanceOf(double.class));
        assertThat(c.motherboardPrice, instanceOf(double.class));
        assertThat(c.memPrice, instanceOf(double.class));
        assertThat(c.perPrice, instanceOf(double.class));
        //Проверяем значения, назначаемые конструктором
        assertEquals(0, c.procPrice, 0.1);
        assertEquals(0, c.motherboardPrice, 0.1);
        assertEquals(1099, c.memPrice, 0.1);
        assertEquals(0, c.perPrice, 0.1);
    }

    @Test
    public void setPrices() {
        //Устанавливаем новые значения
        c.setMem(100);
        c.setMotherboard(100);
        c.setPer(100);
        c.setProc(100);
        //Проверяем правильность установки
        assertEquals(100, c.procPrice, 0.1);
    }
}
```

```

    assertEquals(100, c.motherboardPrice, 0.1);
    assertEquals(100, c.memPrice, 0.1);
    assertEquals(100, c.perPrice, 0.1);
    //Проверяем правильность результата
    assertEquals(400, c.getResult(), 0.1);
}
}

```

Приложение Д.1 – Локальное юнит-тестирование класса Computer

```

package com.example.computer_calc;

import static android.support.test.espresso.Espresso.onData;
import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static android.support.test.espresso.assertion.ViewAssertions.matches;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withText;

import static org.hamcrest.Matchers.allOf;
import static org.hamcrest.Matchers.instanceOf;
import static org.hamcrest.Matchers.is;

import android.support.test.espresso.UiController;
import android.support.test.espresso.ViewAction;
import android.support.test.espresso.matcher.ViewMatchers;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.view.View;
import android.widget.SeekBar;

import org.hamcrest.Matcher;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

```

```

@RunWith(AndroidJUnit4.class)
public class MainActivityTest {
    @Rule
    public ActivityTestRule<MainActivity> mActivityRule
        = new ActivityTestRule<>(MainActivity.class);

    @Test
    public void TestCase() throws InterruptedException {
        String item = "Процессор AMD Athlon 3000G OEM [AM4, 2 x 3.5 ГГц, L2 - 1 МБ, L3 - 4
МБ, 2 x DDR4-2666 МГц, AMD Radeon Vega 3, TDP 35 Вт]";
        onView(withId(R.id.spinner)).perform(click());
        onData(allOf(is(instanceOf(String.class)), is(item))).perform(click());
        onView(withId(R.id.proc_price)).check(matches(withText("4499.0 руб.")));
        onView(withId(R.id.am5)).perform(click());
        onView(withId(R.id.motherboard_price)).check(matches(withText("9899.0 руб.")));
        onView(withId(R.id.seekbar)).perform(setProgress(3));
        onView(withId(R.id.mem_price)).check(matches(withText("3299.0 руб.")));
        onView(withId(R.id.periferal)).perform(click());
        onView(withId(R.id.per_price)).check(matches(withText("15000.0 руб.")));
        onView(withId(R.id.res)).check(matches(withText("Итоговая цена: 32697.0 рублей.")));
    }

    public static ViewAction setProgress(final int progress) {
        return new ViewAction() {
            @Override
            public void perform(UiController uiController, View view) {
                ((SeekBar) view).setProgress(progress);
            }

            @Override
            public String getDescription() {
                return "Set a progress";
            }
        }
    }
}

```

```
@Override
public Matcher<View> getConstraints() {
    return ViewMatchers.isAssignableFrom(SeekBar.class);
}
};
}
}
```

Приложение Д.2 – Инструментальное тестирование класса MainActivity