

# Программное средство для Android смартфона, приложение «Финансы»

Автор данного приложения: Гасанов Р.Э.

Сопровождение данного приложения, произвел: Некрасов Э.А.

## Обзор программного средства «Финансы»



Рисунок 1 - Иконка приложения

При запуске приложения нас встречает страница аутентификации пользователя (рисунок 2) с использованием отпечатка пальца (отпечаток пальца берется из системных настроек), если в данном смартфоне отсутствует сканер или отсутствуют данные об отпечатках пальца пользователя, то приложение не будет запущено.

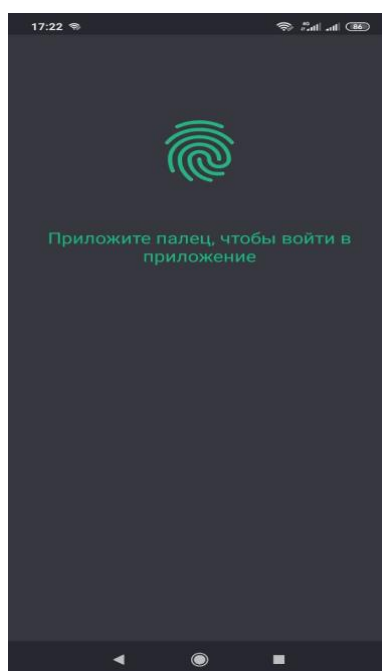


Рисунок 2 – Авторизация пользователя

Далее пользователя встречает меню (рисунок 3), в котором пользователь может выбрать одну из информативных функций данного приложения.

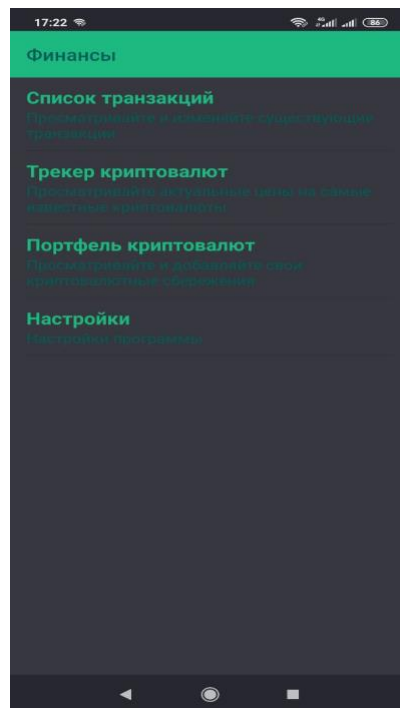


Рисунок 3 – Главное меню

При выборе пользователем пункта меню «Список транзакций», в котором user может просматривать (рисунок 4), добавлять (рисунок 5) и удалять данные о своих покупках или поступлениях. Поля для добавления данных о транзакции в данном приложении никак не ограничены, поэтому пользователь может ввести крайне большое число, что спровоцирует поломку приложения.



Рисунок 4 – Список транзакций

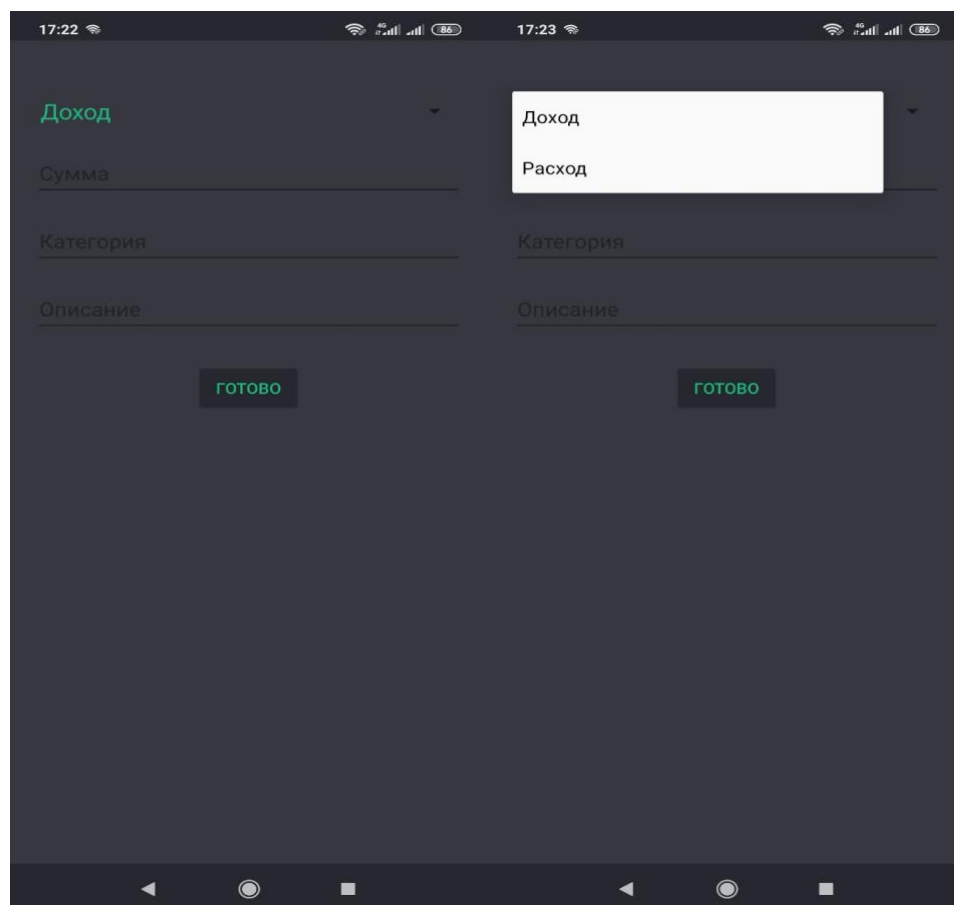


Рисунок 5 – Добавление транзакции

Если же пользователь выбирает пункт меню «Трекер криптовалют», то открывается окно со списком топ 10 криптовалют и данными об их стоимости, а также о капитализации. Но из-за изменения API-ключа используемого сервиса, приложение выдает ошибку и вылетает.

При выборе пользователем пункта «Портфель криптовалют», пользователю открывается окно с выбором определенной криптовалюты и добавлением ее количества в портфель пользователя, данный портфель обновляется после каждого выхода из данного окна. Так-как присутствует та же проблема с API-ключом, у пользователя закрывается приложение и выдается ошибка.

Пункт меню «Настройки» никакого функционала не несет, на данный пункт нет возможности нажать, поэтому данный пункт висит просто для видимости.

## Описание приложения, после сопровождения

После сопровождения, название приложения было решено изменить на «Бумажник», что более ясно отражает суть приложения, а также стандартная иконка приложения была изменена на новую иконку (рисунок 6).



Рисунок 6 – Иконка приложения «Бумажник»

При запуске приложения нас встречает страница аутентификации пользователя (рисунок 7) с использованием отпечатка пальца (отпечаток пальца берется из системных настроек), если в данном смартфоне отсутствует сканер или отсутствуют данные об отпечатках пальца пользователя, то приложение не будет запущено.



Рисунок 7 – Авторизация пользователя

Далее пользователя встречает меню (рисунок 8), в котором пользователь может выбрать одну из информативных функций данного приложения.

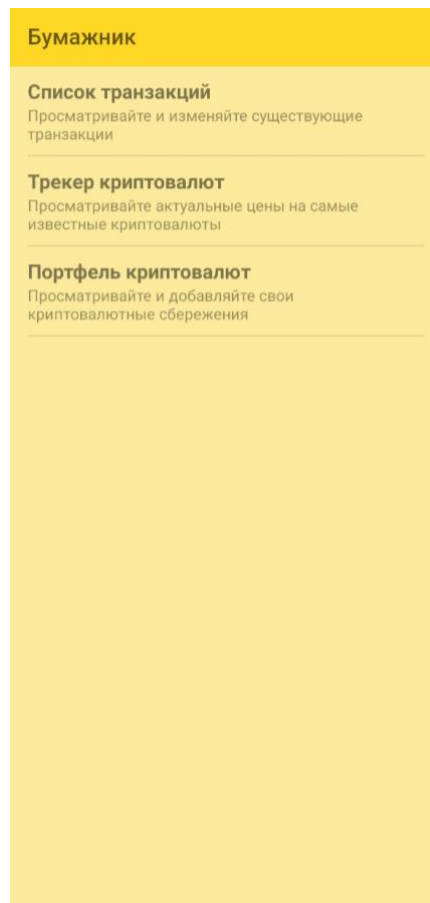


Рисунок 8 – Главное меню

При выборе пользователем пункта меню «Список транзакций», в котором user может просматривать (рисунок 9), добавлять (рисунок 10), удалять данные о своих покупках или поступлениях, а также просматривать баланс (рисунок 11), сколько потрачено и сколько получено (рисунок 12). Поля для добавления данных о транзакции в данном приложении ограничены, что позволяет уберечь пользователя от вылета программы.

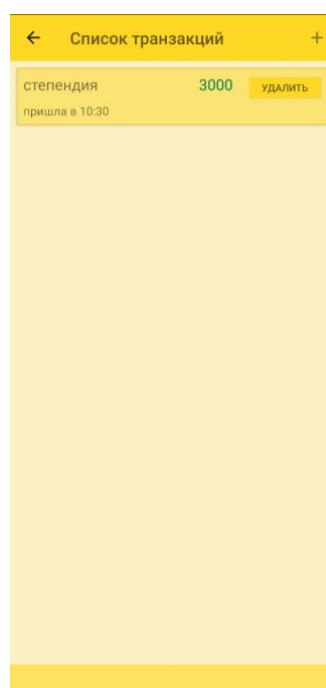


Рисунок 9 – Список транзакций

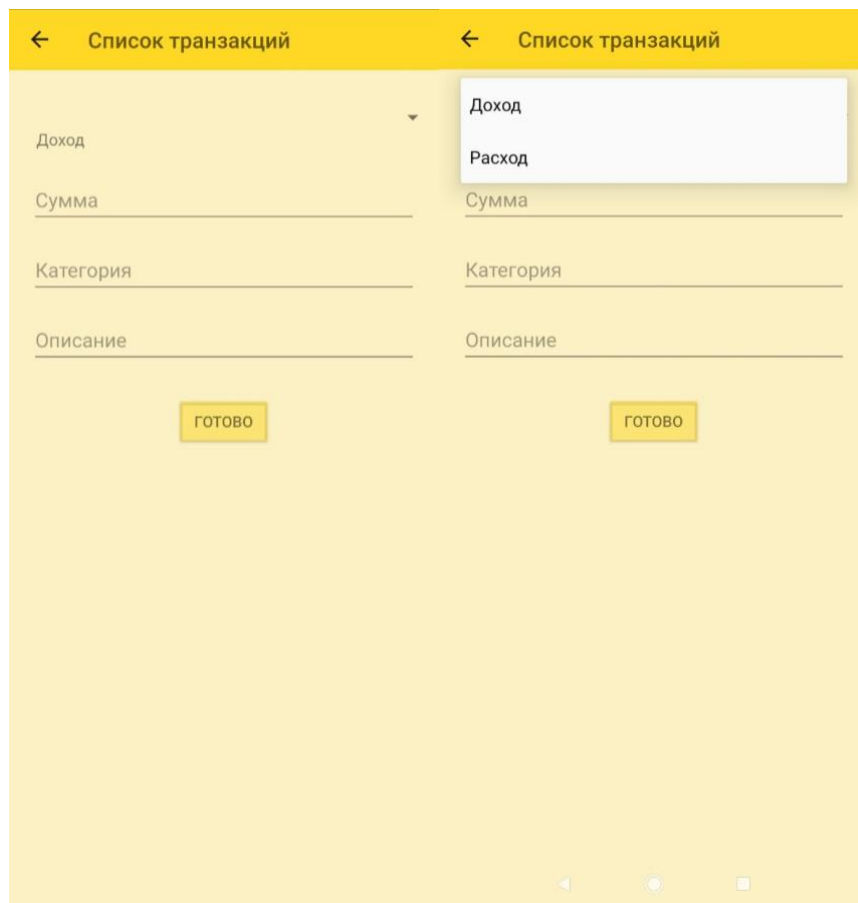


Рисунок 10 – Добавление транзакции

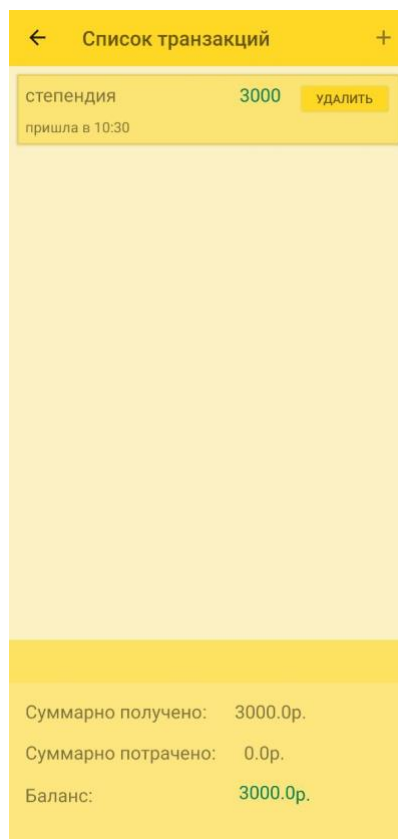


Рисунок 11 – Данные пользователя

Если же пользователь выбирает пункт меню «Трекер криптовалют», то открывается окно (рисунок 12) со списком топ 10 криптовалют и данными об их стоимости, а также о капитализации. Что позволяет пользователю следить за тенденциями на рынке.

Символ	Название	Цена (\$)	Изменение 24ч (%)	Изменение 7дн (%)	Изменение 30дн (%)
BTC	"Bitcoin"	8723.75	0.187831%	-3.84353%	12.9572%
ETH	"Ethereum"	202.016	-0.043322%	-6.32372%	3.13395%
XRP	"XRP"	0.21344	-0.0446447%	-4.48085%	8.22338%
USD	"Tether"	1.00586	-0.190058%	-0.052489%	0.120356%
BCH	"Bitcoin Cash"	243.093	-0.00518033%	-7.02538%	0.600185%
BSV	"Bitcoin SV"	201.489	0.0220204%	-6.32261%	3.38433%
LTC	"Litecoin"	46.3921	-0.26412%	-5.84628%	4.32575%
BNB	"Binance Coin"	16.5768	0.160786%	-5.33676%	1.42646%
EOS	"EOS"	2.72646	0.0497747%	-7.07994%	-0.706284%

Рисунок 12 – Трекер криптовалют

При выборе пользователем пункта «Портфель криптовалют», пользователю открывается окно (рисунок 13) с выбором определенной криптовалюты и добавлением ее количества в портфель пользователя, данный портфель обновляется после каждого выхода из данного окна.

← Портфель криптовалют

Tether ▼

Сумма: 10

ДОБАВИТЬ

"Bitcoin": 0.0  
 "Ethereum": 0.0  
 "XRP": 0.0  
 "Tether": 10.0  
 "Bitcoin Cash": 0.0  
 "Bitcoin SV": 0.0  
 "Litecoin": 0.0  
 "Binance Coin": 0.0  
 "EOS": 0.0  
 "Tezos": 0.0

← Портфель криптовалют

Tether ▼

Сумма: 10

ДОБАВИТЬ

"Bitcoin": 0.0  
 "Ethereum": 0.0  
 "XRP": 0.0  
 "Tether": 10.0  
 "Bitcoin Cash": 0.0  
 "Bitcoin SV": 0.0  
 "Litecoin": 0.0  
 "Binance Coin": 0.0  
 "EOS": 0.0  
 "Tezos": 0.0

Рисунок 13 – Портфель криптовалют



## Перечень изменений и добавлений в функционале программного средства

1) Добавлено ограничение на вводимые данные пользователем, а также возможность ввода копеек (дробная часть числа) в пункте – Добавление транзакции (рисунок 14);

Рисунок 14 – Вводимые данные

2) Добавлено отображение баланса пользователя, а также количество полученных средств и потраченных в пункте – Список транзакций. Данное меню вызывается свайпом снизу вверх (рисунок 15);

Рисунок 15 – Баланс пользователя

3) Отлажена работа приложения с API, что позволяет пользователю работать с пунктами – Трекер криптовалют и Портфель криптовалют (рисунок 12,13).

## Перечень изменений в интерфейсе программного средства:

- 1) Изменение названия приложения: «Финансы» -> «Бумажник»;
- 2) Изменение иконки приложения (рисунок 16).



Рисунок 16 – Было-Стало. Иконка приложения

- 3) Скрыта неиспользуемая кнопка «Настройки» для того, чтобы не вводить в заблуждение пользователя (рисунок 17).

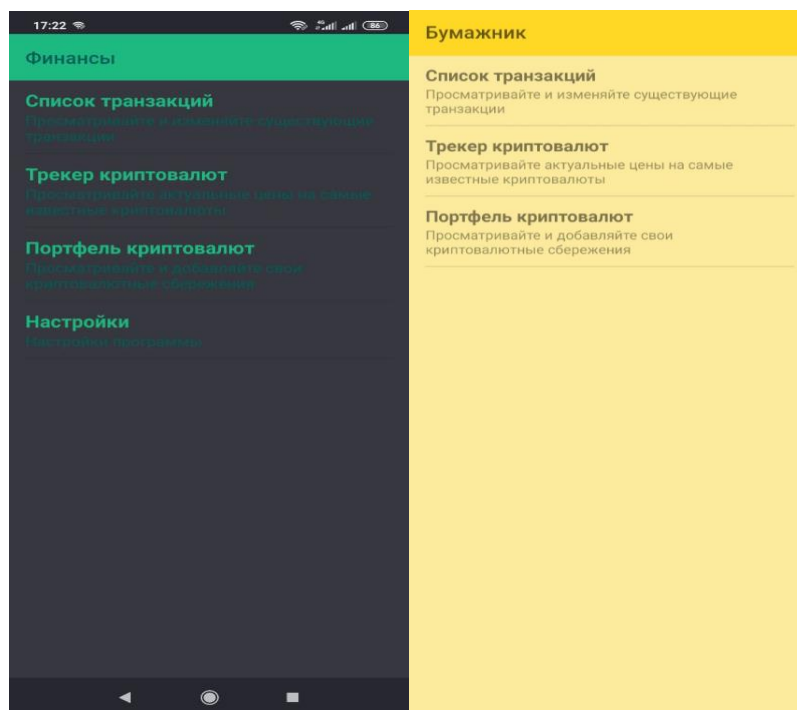


Рисунок 17 – Было-Стало. Главное меню

- 4) Добавлены кнопки «назад» в ToolBar, для перехода в предыдущий пункт меню (рисунок 18).



Рисунок 18 – Было-Стало. ToolBar

- 5) Скрыты нижние системные кнопки (рисунок 19);



Рисунок 19 – Было-Стало. Системные кнопки

- 6) Замена цветовой схемы на желтую во всем приложении, более приятную глазу (рисунок 20).

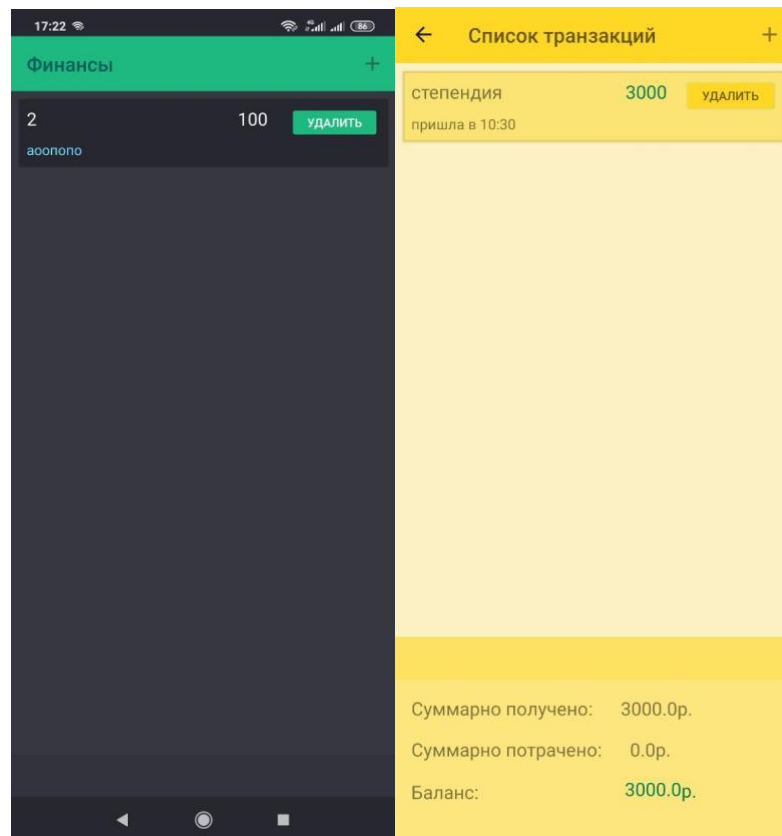


Рисунок 20 – Было-Стало. Цветовая схема

## Рефакторинг кода программного средства:

1) Был заново переписан не работающий модуль, работы с API CoinMarketCup. Данное API изменилось на приватное, а также теперь все запросы, которые приходят с сервера, выстроены в другой форме. Изменения в коде показаны на рисунке 21;

```
private void loadNext10Coin(int index) {
    client=new OkHttpClient();
    request=new Request.Builder().url(String.format("https://api.coinmarketcap.com/v1/ticker/?start=kd&limit=10",index))
        .build();

    swipeRefreshLayout.setRefreshing(true);
    client.newCall(request)
        .enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                Toast.makeText(Tracker.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                String body=response.body().string();
                Gson gson=new Gson();
                final List<CoinModel> newItems=gson.fromJson(body,new TypeToken<List<CoinModel>>().getType());
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        items.addAll(newItems);
                        adaptor.setLoaded();
                        adaptor.updateData(items);
                        swipeRefreshLayout.setRefreshing(false);
                    }
                });
            }
        });
}

request= new Request.Builder().url("https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest?CMC_PRO_API_KEY=e477882e-2d4e-42f7-bd92-6e4b4b4b4b4b");
Log.d("Request", request.toString());

swipeRefreshLayout.setRefreshing(false);
client.newCall(request)
    .enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            Toast.makeText(context, Tracker.this, e.getMessage(), Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            String body=response.body().string();
            Gson gson=new Gson();
            Log.d("body", body);

            Gson googleGson = new Gson();
            JsonObject data = googleGson.fromJson(body, JsonObject.class);

            Log.d("body_data", data.toString());

            JSONArray jsonArray = data.getAsJSONArray( "data");
            ArrayList<JsonObject> jsonObjList = googleGson.fromJson(jsonArray, ArrayList.class);

            final List<CoinModel> newItems = new ArrayList<>();
            for(int i=0; i< jsonObjList.size(); i++) {
                CoinModel item = new CoinModel();

                item.setId(jsonObjList.get(i).getAsJsonObject().get("id").toString());
                item.setName(jsonObjList.get(i).getAsJsonObject().get("name").toString());
                item.setSymbol(jsonObjList.get(i).getAsJsonObject().get("symbol").toString());

                JsonObject quote = googleGson.fromJson(jsonObjList.get(i).getAsJsonObject().get("quote"), JsonObject.class);
                JsonElement jsonArrQuote = quote.getAsJsonObject().get("quote");

                JsonObject usd = googleGson.fromJson(jsonArrQuote.toString(), JsonObject.class);
                JsonElement jsonArrUsd = usd.getAsJsonObject("USD");

                item.setPercent_change_1h(jsonArrUsd.getAsJsonObject().get("percent_change_1h").toString());
                item.setPercent_change_24h(jsonArrUsd.getAsJsonObject().get("percent_change_24h").toString());
                item.setPercent_change_7d(jsonArrUsd.getAsJsonObject().get("percent_change_7d").toString());
                item.setPrice_usd(jsonArrUsd.getAsJsonObject().get("price").toString());

                newItems.add(item);
            }
        }
    });
```

Рисунок 23 – До-После. API CoinMarketCup

2) Добавлены комментарии о работе главных классов (рисунок 22, 23);

```
/*
    Класс Portfolio отвечает за отображение данных в портфеле криптовалют,
    запрос этих данных на сервере CoinMarketCup, а также отображение и обработку
    нажатий на элементы Activity
*/

public class Portfolio extends AppCompatActivity {
```

Рисунок 22 – Описание классов

```

/*
 * Класс Tracker отвечает за отображение данных в трекаре криптовалют,
 * запрос этих данных на сервере CoinMarketCup, а также отображение и обработку
 * нажатий на элементы Activity
 */

public class Tracker extends AppCompatActivity {

/*
 * Класс ChooseActivity отвечает за отображение списка меню в Activity,
 * а также обработку элементов списка.
 */

public class ChooseActivity extends AppCompatActivity {
    private List<Choose> CL = new ArrayList<>();
    private ChooseAdapter adapter;

/*
 * Класс FingerprintActivity отвечает за проверку отпечатка пальца у пользователя.
 */

public class FingerprintActivity extends Activity {

    private KeyStore keyStore;
    // Variable used for storing the key in the Android Keystore container
    private static final String KEY_NAME = "androidHive";
    private Cipher cipher;
    private TextView textView;

/*
 * Класс MainActivity отвечает за отображение списка транзакций, удаления транзакций,
 * обработку элементов списка и отображение баланса пользователя.
 */

public class MainActivity extends AppCompatActivity implements PostsAdapter.OnDeleteButtonClickListener {
    private PostsAdapter postsAdapter;

```

Рисунок 23 – Описание классов

3) Исправлены строчные переносы, а также отсутствие табов и пробелов (рисунок 24).

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (data == null) {return;}
    String title = data.getStringExtra( name: "TITLE");
    String content = data.getStringExtra( name: "CONTENT");
    String amount = String.valueOf(data.getStringExtra( name: "AMOUNT"));
    String type = data.getStringExtra( name: "TYPE");
    postViewModel.savePost(new Post(title, content, amount, type));

    if (type.equals("Расход")) {totalSpent += Integer.parseInt(amount); updateTextView(Integer.toString(totalSpent)); } else {totalEarned +=Integer
    SharedPreferences.Editor editor = mSettings.edit();
    editor.putInt(APP_PREFERENCES_TOTALSPENT, totalSpent);
    editor.putInt(APP_PREFERENCES_TOTALEARNED, totalEarned);
    editor.apply();
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (data == null) return;

    String title = data.getStringExtra( name: "TITLE");
    String content = data.getStringExtra( name: "CONTENT");
    String amount = String.valueOf(data.getStringExtra( name: "AMOUNT"));
    String type = data.getStringExtra( name: "TYPE");

    postViewModel.savePost(new Post(title, content, amount, type));

    if (type.equals("Расход")) {
        totalSpent += Float.parseFloat(amount);
        updateTextView(Float.toString(totalSpent));
        updateSumAmount(Float.toString( f: totalEarned - totalSpent));
    } else
    {
        totalEarned +=Float.parseFloat(amount);
        updateTotalEarned(Float.toString(totalEarned));
        updateSumAmount(Float.toString( f: totalEarned - totalSpent));
    }

    SharedPreferences.Editor editor = mSettings.edit();

    editor.putFloat(APP_PREFERENCES_TOTALSPENT, totalSpent);
    editor.putFloat(APP_PREFERENCES_TOTALEARNED, totalEarned);
    editor.apply();
}

```

Рисунок 24 – До-После. На примере функции MainActivity.onActivityResult

## **Ссылки:**

1) github репозиторий проекта: <https://github.com/edward-nek/FinancePro>