

Анализ кода программы **FaceTracker**

В ходе анализа исходного кода данной приложения были выделены следующие достоинства и недостатки.

Достоинства:

1. Организация проекта в виде классов.
2. В основном код выдержан в едином стиле написания.
3. Используются стандартные структуры данных.
4. Используются разработанные и принятые сообществом сторонние компоненты.
5. Используются говорящие названия переменных и методов.

Недостатки:

1. Закомментированный участок кода.

```
45     FaceGraphic(GraphicOverlay overlay) {
46         super(overlay);
47         // PathBeard=path;
48         mNotReadyMessage = overlay.getContext().getResources().getString(R.string.not_ready_message);
49         mReadyMessage = overlay.getContext().getResources().getString(R.string.ready_message);
```

2. Малоинформативное название массива объектов, вдобавок еще и не используется в дальнейшем.

```
111         float right = x + xOffset;
112         float bottom = y + yOffset;
113         Object[] L = new Object[100];
114
115         canvas.drawBitmap(HelmetBitmap, left + 15, top, mPaint); // Накладываем шлем
```

3. Избыточная, в дальнейшем неиспользуемая переменная класса и соотв. ей метод.

```
3     class Beard {
4         private String name;
5         private String age;
6         private int photoId;
7
17
18         public int getPhotoId() {
19             return photoId;
20     }
```

4. Нерациональная загрузка файлов из директории, вместо использования файла ресурсов программы.

```
123     private void initializeData() {
124         beards = new ArrayList<>();
125         File mFile = new File("/storage/emulated/0/Download/");
126
127         String[] mFiles = mFile.list(new FilenameFilter() {
128
129             @Override
130             public boolean accept(File folder, String name) {
131                 return name.endsWith(".png");
132             }
133
134         });
135
136         for (int i = 0; i < mFiles.length; i++) {
137             beards.add(new Beard("Борода№" + (i + 1), mFiles[i], R.drawable.emma));
138         }
139     }
```

5. Несоблюдение общего стиля форматирования исходных данных.

```
55     private CameraSource mCameraSource = null;
56     private int Camera_Back_or_Front=CameraSource.CAMERA_FACING_FRONT;
57
```

6. Несоблюдение соглашения о именовании переменных.

```
102     float xOffset = scaleX(face.getWidth() / 2.0f);
103     float yOffset = scaleY(face.getHeight() / 2.0f);
104
105     int widthMask = (int) face.getWidth() * 2;
106     int heightMask = (int) (face.getHeight() * 1.5f);
107     Bitmap helmetBitmap = Bitmap.createScaledBitmap(mBitmap, widthMask, heightMask, false);
108     Bitmap newBitmapBeard = Bitmap.createScaledBitmap(mBitmapBeard, widthMask, heightMask, false);
109
110     GraphicFaceTracker(GraphicOverlay overlay, String path) {
111         mOverlay = overlay;
112         path = path;
113         mFaceGraphic = new FaceGraphic(overlay);
114     }
```

7. Ошибка логического названия переменной. Называется age, хотя используется для хранения пути.

```
106         public void onClick(View view, int position) {
107             Beard beard = beards.get(position);
108             Toast toast = Toast.makeText(getApplicationContext(),
109                 "Выбрано: " + beard.getName(), Toast.LENGTH_SHORT);
110             toast.show();
111             FILE = beard.getAge();
112         }

22     public String getAge() {
23
24         return age;
25     }
```

8. При инициализации объектов, в конструктор передается один и тот же ид ресурса со странным наименованием.

```
135
136         for (int i = 0; i < mFiles.length; i++) {
137             beards.add(new Beard("Борода№" + (i + 1), mFiles[i], R.drawable.emma));
138         }
139     }
```

9. Низкая документированность проекта.

Подсчитаем оценку по формуле:

$$\text{оценка} = \frac{\text{кол} - \text{во плюсов}}{\text{кол} - \text{во плюсов} + \text{кол} - \text{во минусов}} * 10 = \frac{5}{14} * 10 \approx 3.6$$

Путем округления, получим итоговую оценку: **4**