

Достоинства:

+1 Единый стиль именования переменных, методов, классов

```
class Beard {
    private String name;
    private String age;
    private int photoId;

    Beard(String name, String age, int photoId) {
        this.name = name;
        this.age = age;
        this.photoId = photoId;
    }

    public String getName() {
        return name;
    }

    public int getPhotoId() {
        return photoId;
    }

    public String getAge() {
        return age;
    }
}
```

+2 Информативное название полей

+3 Достаточное комментирование кода

+4 Использование объектно-ориентированного подхода к написанию кода

Недостатки:

-1 Нереализованные/неиспользуемые методы

```
private FileUtils() {
}
```

-2 Закомментированный код

```
FaceGraphic(GraphicOverlay overlay) {
    super(overlay);
    // PathBeard=path;
    mNotReadyMessage = overlay.getContext().getResources().getString(R.string.not_ready_message);
    mReadyMessage = overlay.getContext().getResources().getString(R.string.ready_message);
    mPaint = new Paint();
    mPaint.setColor(VALID_COLOR);
    mPaint.setStyle(Paint.Style.STROKE);
    mPaint.setStrokeWidth(BOX_STROKE_WIDTH);
    mPaint.setTextSize(ID_TEXT_SIZE);
}
```

-3 Поддержка единственного расширения изображений

```
private void initializeData() {
    beards = new ArrayList<>();
    File mFile = new File("/storage/emulated/0/Download/");

    String[] mFiles = mFile.list(new FilenameFilter() {
        @Override
        public boolean accept(File folder, String name) {
            return name.endsWith(".png");
        }
    });
}
```

Итоговая оценка: 6/10