

Разделение логики и пользовательского интерфейса. Система плагинов.

Никитин Михаил

Срок сдачи: 7 октября (23:55).

1 Обзор задания

В рамках первого задания практикума Вам необходимо будет отработать два навыка:

1. Разделение логики и пользовательского интерфейса;
2. Разработка системы плагинов.

2 Разделение логики и пользовательского интерфейса (3 балла)

В рамках данного подзадания Вам необходимо пострить программу в соответствии с архитектурой **MVC** с **активной моделью**. Таким образом, на верхнем уровне приложение должно быть разбито на три модуля:

- модель — основная логика приложения;
- представление — способ отображения результатов работы;
- контроллер — способ интерпретации пользовательского ввода.

Вам потребуется реализовать два пользовательских интерфейса — графический и консольный. **То есть результатом сборки должны быть два исполняемых файла.** Классы модели включаются в оба модуля на этапе компиляции.

Модель должна уведомлять представления о следующих событиях:

- изображение загружено;
- произведено разделение изображения на цветовые каналы;
- удалены рамки пленки (если реализовано);
- произведено совмещение каналов;
- к изображению применен фильтр постобработки (если реализовано).

И в консольном, и в графическом варианте программы должно быть 2 представления:

1. текстовое представление; **в смысле представление для текста**
2. графическое представление. **в смысле представление для графики**

Текстовое представление используется для вывода лога по ходу работы вашей программы (с указанием времени прихода оповещений о соответствующих изменениях в модели). В случае консольного интерфейса лог работы выводится непосредственно в консоль, а в случае графического интерфейса вся информация должна отображаться в текстовом поле интерфейса.

Под графическим представлением понимается сохранение в файл полученного изображения в случае консольного приложения, или отображение его на графическом интерфейсе. **Если реализованы фильтры постпроцессинга, то после их применения должно происходить автоматическое пересохранение изображения (консольный интерфейс) или обновление его отображения (графический интерфейс).**

Обратите внимание, что модель не должна зависеть ни от представления, ни от контроллера. Чтобы добиться этого, воспользуйтесь шаблоном **Наблюдатель**.

2.1 Описание графического интерфейса

Минимальный графический интерфейс должен включать следующие элементы:

- меню:
 - *Файл:*
 - * *Открыть файл;*
 - * *Выход.*
 - *Обработка:*
 - * *Совместить каналы;*
 - * *...*
 - *Плагины:*
 - * *Обновить плагины;*
 - * имя первого найденного плагина;
 - * имя второго найденного плагина;
 - *
- поле для отображения состояния модели на изображении;
- текстовое поле для отображения лога уведомлений от модели.

При выборе плагина из списка, реализованный в нем фильтр должен быть применен к изображению.

2.2 Расширение консольного интерфейса

Для проверки работоспособности плагинов в консольном варианте программы предлагается добавить еще один необязательный параметр `-filter` в качестве аргументов командной строки.

Таким образом, если выставлен этот параметр, то программа должна:

1. Найти все доступные плагины;
2. Вывести найденные фильтры на экран;
3. Попросить пользователя выбрать фильтр (+ уточнить параметры, если требуется);
4. Применить его к изображению;
5. Записать результат в файл.

Пример взаимодействия:

```
$ ./align src.bmp dst.bmp -filter
1 dll is found, 2 filters are loaded.
[1] Median
[2] Gray world
Choose filter:
> 2
"Gray_world" was applied.
$
```

3 Разработка системы плагинов (2 балла)

В рамках второго подзадания Вам необходимо будет разработать систему плагинов вашего приложения. Плагины должны содержать фильтры постобработки, которые предложены вам для реализации в бонусной части задания машграфа.

К данному заданию предъявляются следующие требования:

- Необходимо реализовать как минимум 2 плагина (например, автоконтраст и медианный фильтр);
- Регистрация плагинов должна происходить с помощью менеджера плагинов;

- Регистрация плагинов должна происходить при запуске приложения;
- В случае графического интерфейса регистрация плагинов должна проходить также и по запросу пользователя;
- Компоненты должны лежать в подкаталоге plugins (относительно каталога с исполняемым файлом программы);
- Должна быть реализована динамическая загрузка фильтров путем поиска нужных плагинов во всех динамических библиотеках (dll/so), которые лежат в папке с плагинами;
- Программа не должна зависеть от имен библиотек и от их количества. В случае отсутствия библиотек программа должна сообщать об этом пользователю и корректно работать в базовом режиме без фильтров постобработки;
- В графическом режиме список доступных плагинов должен быть отображен в интерфейсе.