

Варианты реализации

CLI (Command Line Interface)

Программу требуется реализовать в виде утилиты, подобной стандартным linux-утилитам с которыми Вы уже имели дело: управление осуществляется посредством аргументов командной строки. Конкретный список команд зависит от решаемой задачи, но в общем случае обязательно:

- Наличие справки, которая распечатывается при вызове утилиты без аргументов или стандартными ключами (-h, -help)
- Обязательная обработка всех возможных исключительных ситуаций (даже бьющийся в агонии по клавиатуре пользователь не должен “уронить” вашу программу неверными аргументами)
- Для всех(для которых это имеет смысл) ключей должны быть как полные так и сокращенные версии(-h, -help)
- По-умолчанию последним аргументов утилите должно передаваться имя bmp файла, в который и будут записаны изменения, однако это можно переопределить соответствующими ключами, как и имя выходного файла.
- Утилита должна иметь функцию печати подробной информации о bmp-файле
- В случае, если программой будут поддерживаться не все версии bmp-файлов, программа должна выводить об этом внятное сообщение, а не крашиться.
- Для каждого инструмента должен быть соответствующий ключ и ключи для его конфигурирования. Например, рисование прямоугольника может выглядеть как-то так:

```
mypaint --rectangle --start 0 0 --end 100 50 --color red picture.bmp
```

Реализация интерфейса должна быть с использованием **getopt**

Ссылки на ресурсы

- Кратко о getopt: https://www.gnu.org/software/libc/manual/html_node/Getopt.html
- Подробно о getopt: <https://www.gnu.org/software/libc/manual/pdf/libc.pdf> В основном раздел 25.1.2 Parsing Program Arguments, но ctrl+f тоже принесет плоды

GUI (Graphical User Interface)

Программу требуется реализовать в виде оконного графического приложения с использованием C++ фреймворка Qt (надо быть готовым в изучению C++ самостоятельно). Конкретный функционал приложения зависит от решаемой задачи, но в общем случае должно быть:

- Окно отображения изображения
- Некоторая палитра инструментов и возможность их настройки (цвет, например)(см. Paint, PhotoShop и т.д.)
- Пользователь должен иметь возможность, выбрав инструмент, на рисунке выделить

область его применения

- Некоторое меню, позволяющее открыть/сохранить изображение, посмотреть подробную информацию об открытом bmp-файле, посмотреть информацию об авторе и краткую справку по приложению.
- Никакие действия пользователя не должны приводить к “падению” программы.
- В случае, если программой будут поддерживаться не все версии bmp-файлов, программа должна выводить об этом внятное сообщение, а не крашиться.

Ссылки на ресурсы

- Официальный сайт Qt: <https://www.qt.io>
- Откуда скачать Qt Creator: <https://www.qt.io/qt-features-libraries-apis-tools-and-ide/>
- “Первые шаги” для начинающих на Qt wiki: https://wiki.qt.io/Qt_for_Beginners
- Видеолекции Кирилла Владимировича по Qt:
<https://www.youtube.com/playlist?list=PL754BBE9A89BA9D3B>

Нюансы реализации

Qt в том числе имеет средства для работы с изображениями. Этими средствами (за исключением средств визуального отображения) пользоваться запрещено. Что это значит:

- Вами должен быть реализован класс, хранящий в себе представление изображения.
- Должны быть реализованы методы для загрузки изображения из файла и выгрузки его в файл.
- Заголовки изображения должны быть представлены в виде отдельных структур для каждого заголовка, если их несколько.
- Все операции из задания должны быть реализованы в виде отдельных public методов и реализованы должны быть самостоятельно (нельзя использовать стандартные средства, который сами все нарисуют)

From:
<http://se.moevm.info/> - **se.moevm.info**

Permanent link:
http://se.moevm.info/doku.php/courses:programming:rules_extra_kurs?rev=1519648854 

Last update: **2022/12/10 09:08**