

Уважаемый проверяющий, здравствуйте. Посчитал нужным загрузить некоторые свои мысли вместе с кодом решения задачи. Возможно, код выглядит не очень правильно, я буду рад его исправить. Но годовой курс по Java, и настоятельные рекомендации моего практика, Мартина Райлы, приучили думать сначала об архитектуре. Возможно, она покажется неявной, поэтому хотелось бы отразить ее написание. Все описанные файлы я сложил по директориям, как того требует ТЗ.

1. Я рассматриваю задачу с нескольких уровней абстракции. Сперва файл рассматривается «на уровне операционной системы», если так можно выразиться. Сейчас для программы он представляет собой некоторую сущность, хранящуюся на жестком диске. В идеальном случае, хотелось бы верить, что это bmp файл с нужными характеристиками. С нужными ли – проверяют функции модуля `file_manipulator.c`. Разумеется, сначала я написал заголовочный файл с перечислением тех функций, которые точно буду реализовывать в ресурсном файле. Итак, файл проверяется на соответствие заданным ограничениям, выдает результат данной проверки, а также способен открывать и закрывать файл. Допустим, файл соответствует всем ограничениям. Тогда программа переходит к следующему пункту.
2. Файл «на уровне содержимого». Теперь не нужно переживать за характеристики файла и можно начать изучать непосредственно его содержимое. Поэтому заголовочный файл `bmp_manipulator.h` и соответствующий ему модуль позволяют программе выделять память и считывать на это место содержимое файла, высчитывать необходимое количество памяти, `padding-bytes` и так далее. Какое-то содержимое из другой области памяти (заранее указанной), функция, наоборот, позволяет записать в указанный файл.
3. Логику поворота картинки посчитал полезным тоже вынести в отдельный модуль, так как показалось полезным подумать о расширяемости, ведь поворачивать в теории можно любую абстрактную картинку, не обязательно bmp. К сожалению, я не придумал, как поворачивать картинку против часовой стрелки, но понял, что 3 раза по часовой – это то же самое, ведь +270 градусов и -90 градусов эквиваленты (:
4. Модуль `main` собирает всю описанную логику в кучу, но перед этим считывает из аргументов командной строки файл картинки и файл, в который нужно будет сохранить перевернутую картинку. После всех манипуляций память очищается.

Надеюсь, этот файл и написанные особые комментарии к каждой функции (которые в Java позволяют формировать Javadoc. Экспериментальный метод показал, что Ciliop позволяет писать что-то подобное). Спасибо за прочтение!