

# Тема 1(2) — Исследование примеров модульной структуры программы. Анализ стыковки (coupling)

Исследование провёл студент группы 22107 Гордеев Никита  
Дата выполнения работы над ошибками 31.05.2022 (Версия 4)

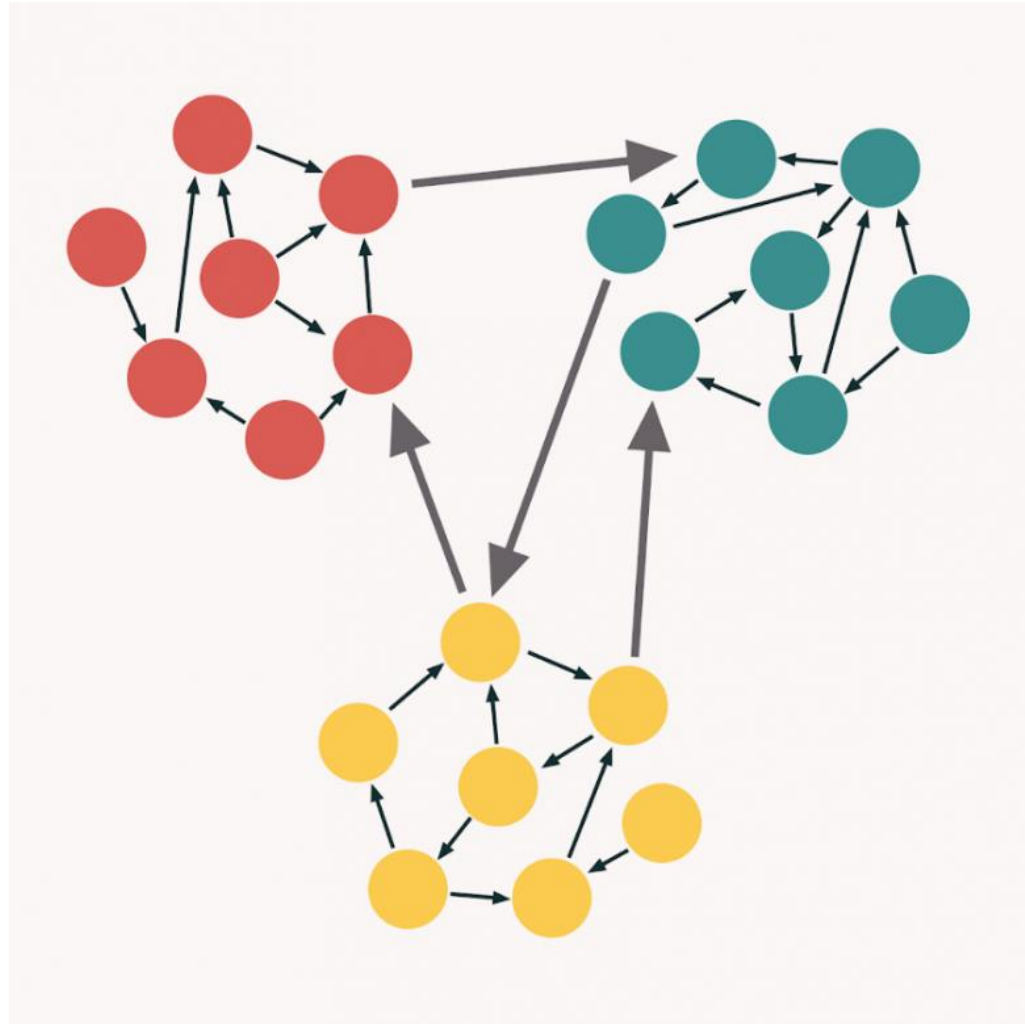
# 1) Постановка задачи

## Определение:

- Стыковка — мера относительной независимости модулей относительно других модулей.
- Чем слабее стыковка, тем проще изменять модуль. Является признаком хорошо структурированной и хорошо спроектированной системы

## Практическая задача:

- Проанализировать стыковку модулей в программе, выяснить нужны ли изменения.



## 2) Выбор программы

### Программа:

- Художественные фильтры - реализация преобразования изображения в оттенки серого

### Идея:

- Реализация фильтра представляет собой отдельный файл исходного кода с одной основной функцией фильтра и несколькими вспомогательными функциями.

Рисунок 17.18. Пример применения фильтра «Пикселизация»

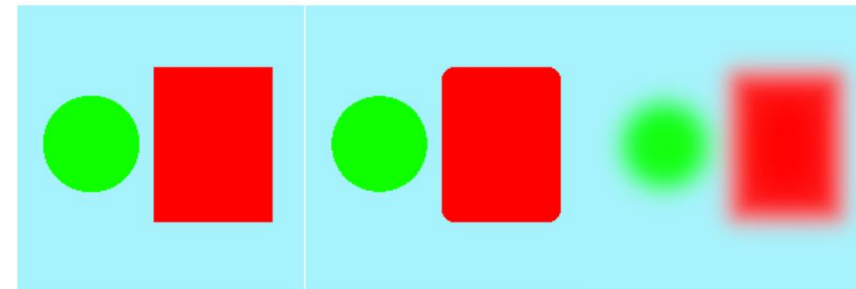


Исходное изображение



Применён фильтр «Пикселизация»

Рисунок 17.17. «Медианный» фильтр vs «Гауссово» размытие



Слева: исходное изображение

В середине: медианный фильтр

Справа: Гаусс

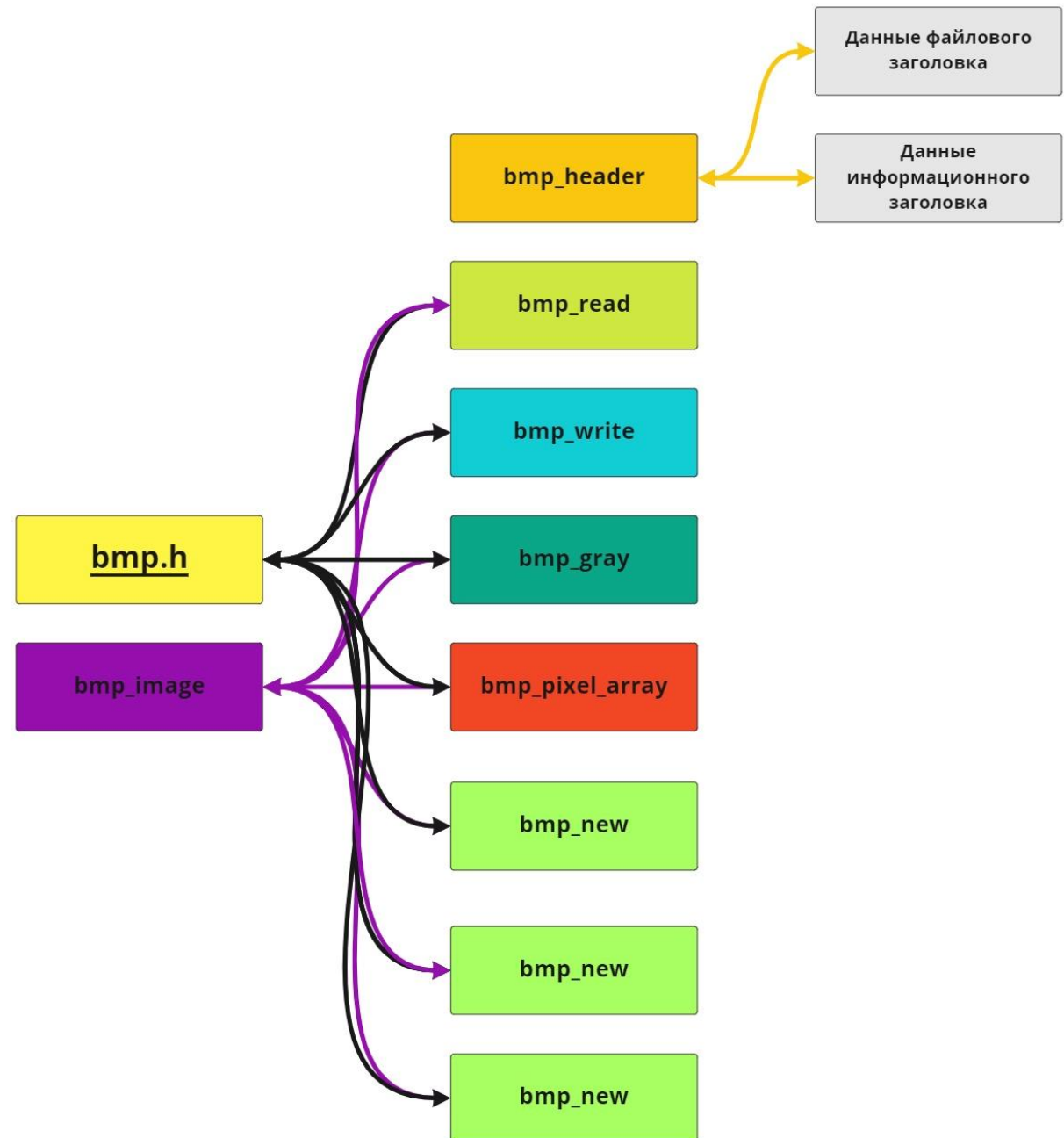
## 3.1) Модуль bmp.h

### Описание:

- bmp.h — заголовочный файл с описаниями форматов.

### Анализ модуля:

- Высокая стыковка функций между собой.
- Каждый новый модуль требует подключения двух родительских



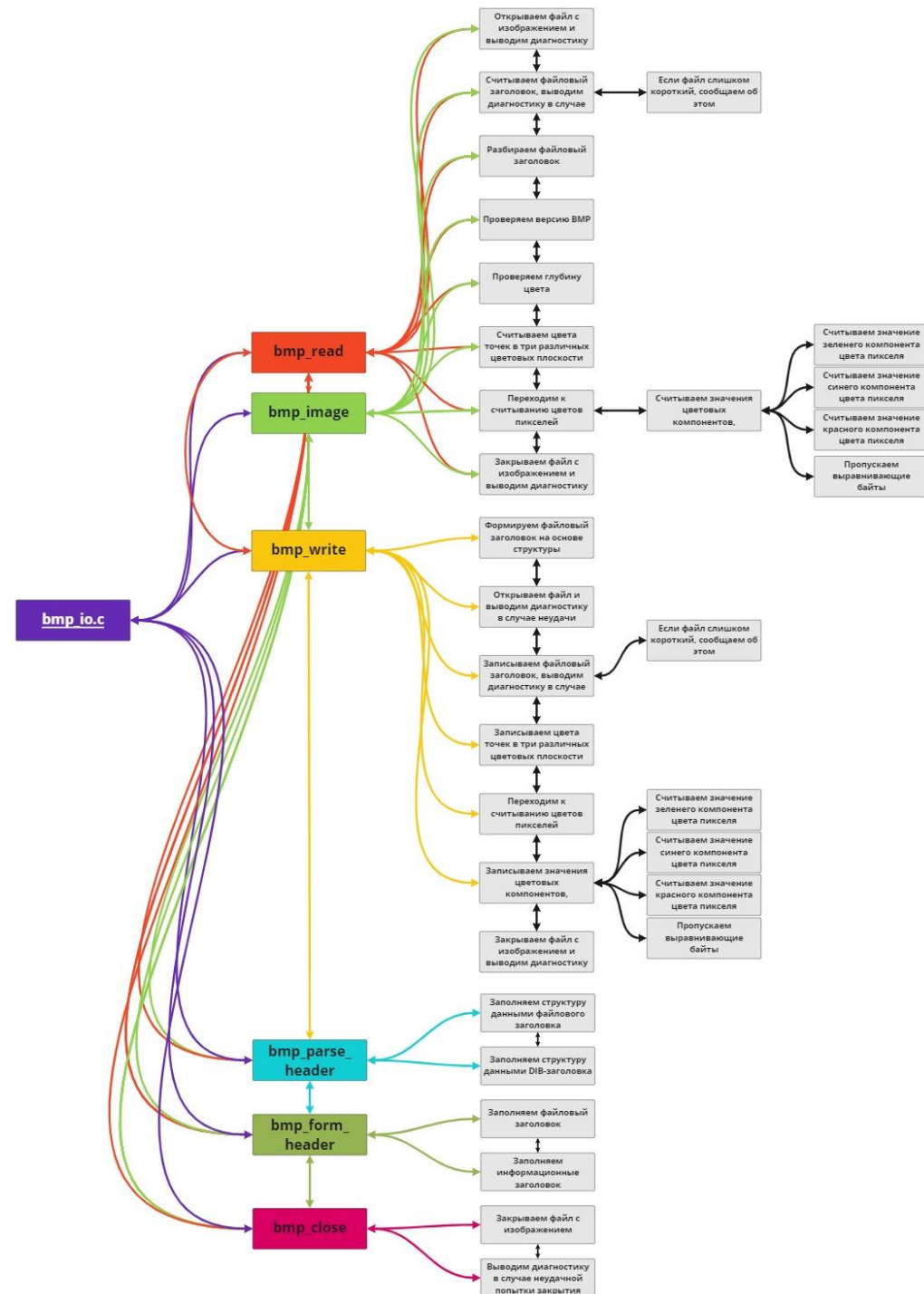
## 3.2) Модуль bmp\_io.c

### Описание:

- bmp\_io.c - функции чтения-записи файлов изображений в формате BMP.

### Анализ модуля:

- Высокая стыковка функций между собой.
- Каждый новый модуль требует подключения всех соседних



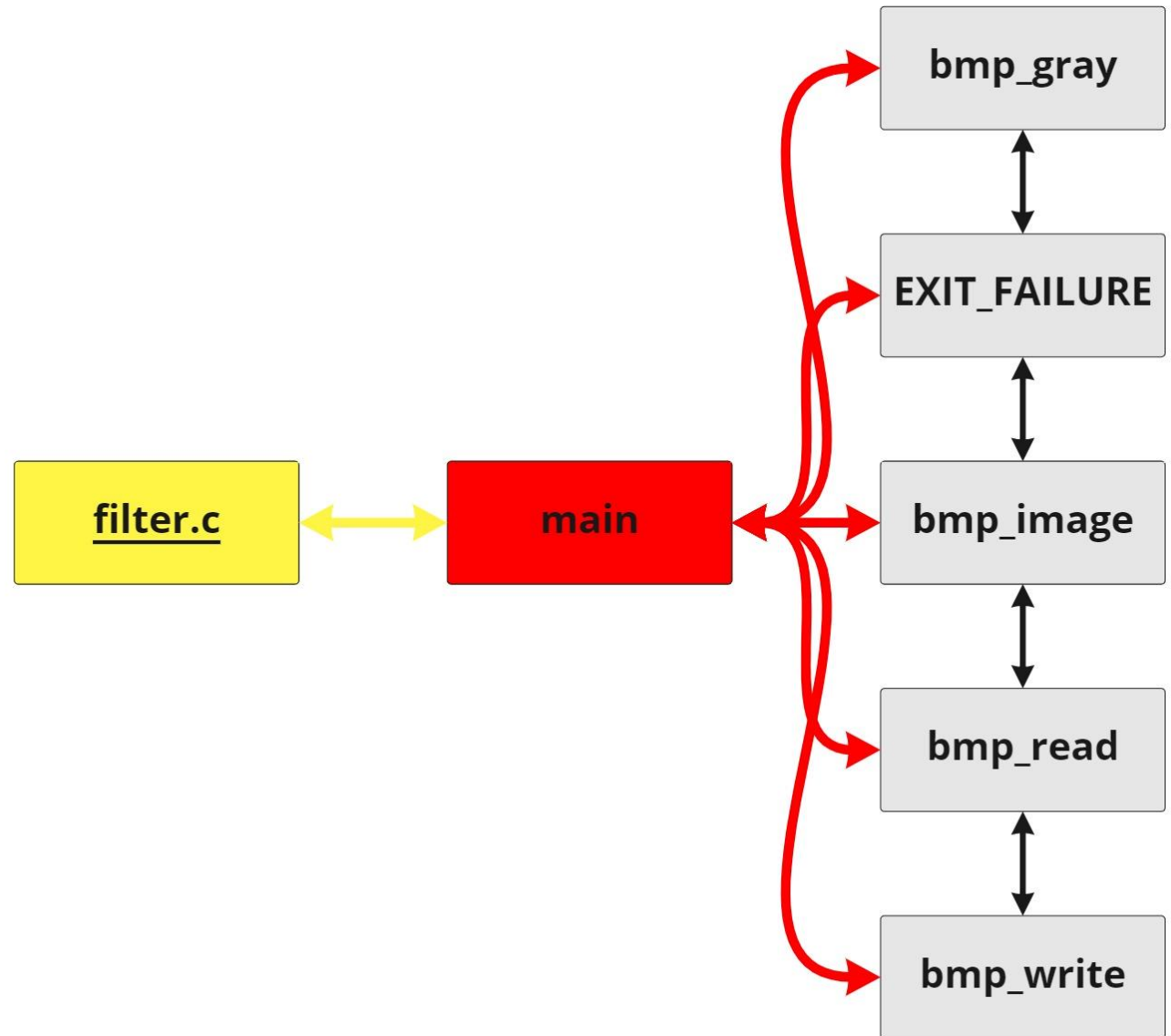
## 3.2) Модуль filter.c

### Описание:

- filter.c — программа для наложения фильтров на изображение в формате BMP

### Анализ модуля:

- Одна функция в модуле
- Низкая стыковка



### 3.3) Модуль bmp.grey.c

#### Описание:

bmp.grey — программа преобразует цвета пикселей в черно-белую гамму

#### Анализ модуля:

- Низкая стыковка функций между собой.
- Новый модуль требует подключения предыдущего



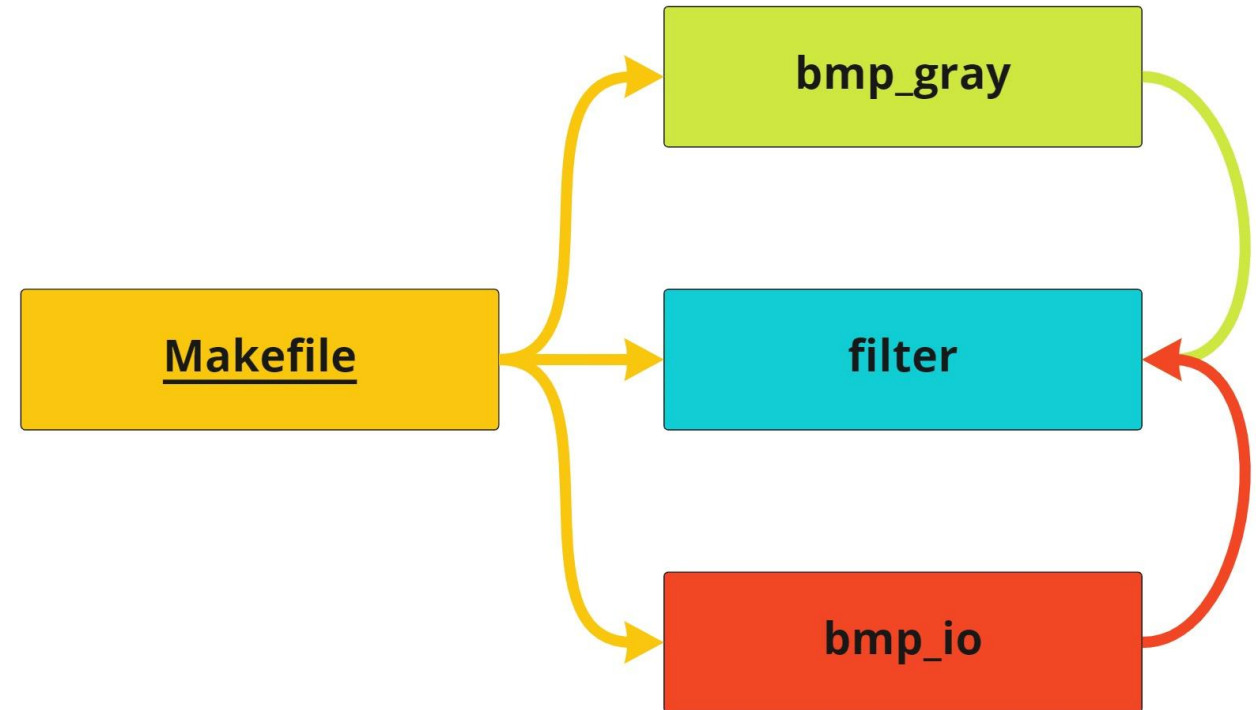
## 3.5) Модуль Makefile

### Описание:

Makefile — это набор инструкций для программы make, которая помогает собирать программный проект.

### Анализ модуля:

- Высокая стыковка функций между собой.
- Главный модуль требует подключения остальных





## 4) Вывод по структуре программы

### Недостатки:

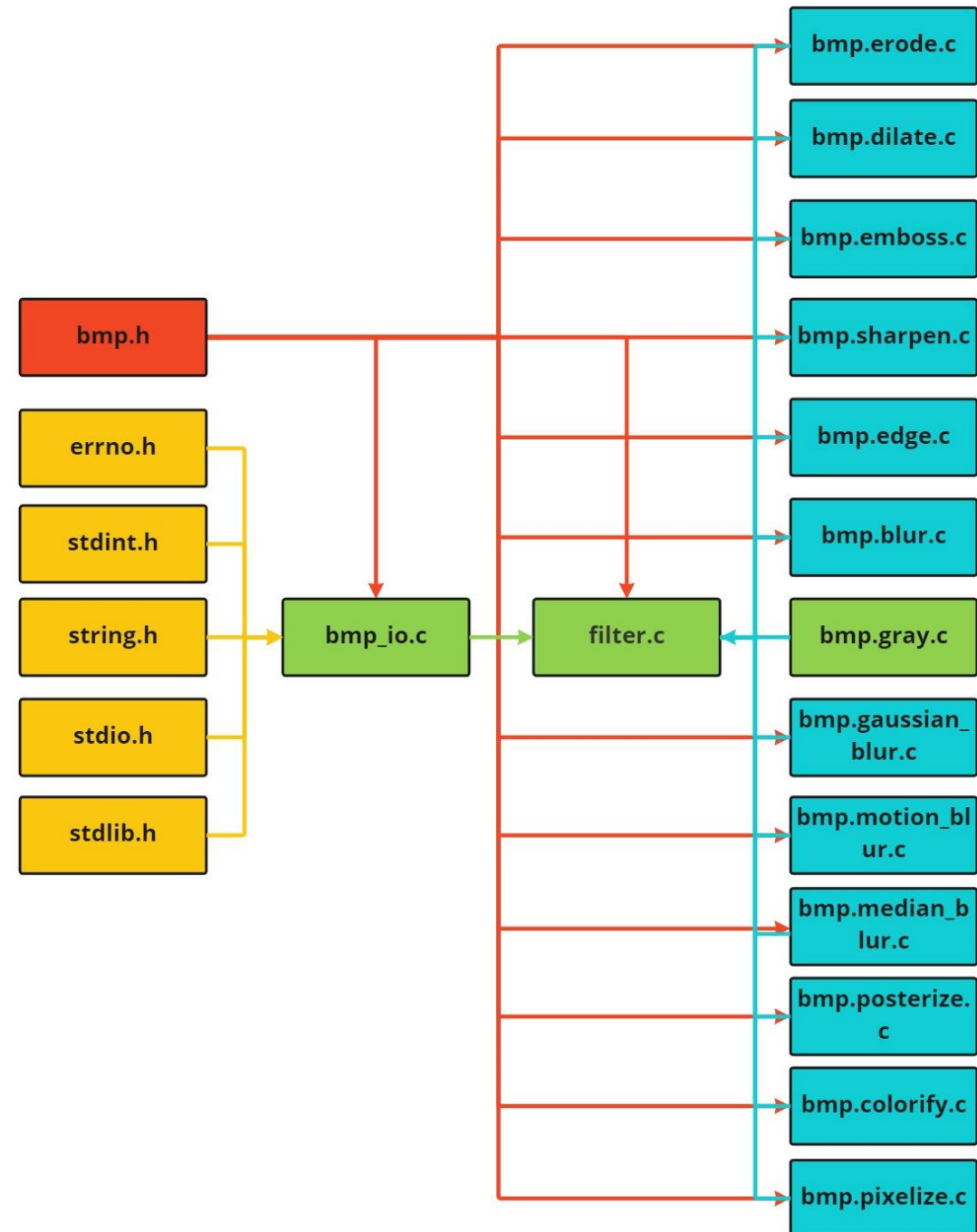
- Данный код имеет высокую стыковку
- Плохо структурированная и спроектированная системы

### Преимущества:

- Соответствует общим показателям хорошей читаемости и сопровождаемости.

### Итог:

Плохая связи всех модулей, программу нужно доработать.



## 5) Материалы

1. **6. Художественные фильтры (12.12.2020) // Курсы кафедры ИМО URL:**  
**<https://moodle.cs.petrso.ru/mod/assign/view.php?id=999> (дата обращения: 14.03.2022).**
2. **Cohesion и Coupling: отличия // Хабр URL:**  
**<https://habr.com/ru/post/568216/> (дата обращения: 12.03.2022).**

Курсы кафедры ИМО Русский [ru]

### 6. Художественные фильтры (12.12.2020)

Базовая оценка: 8  
Срок работы: 12.12.2020

#### Постановка задачи

Художественные фильтры в растровых графических редакторах представляют собой не что иное, как реализацию алгоритмов обработки двумерного массива цветов пикселей изображения. Далее будем рассматривать изображение в формате BMP с глубиной цвета 24 бита на пиксель. В этом случае цвет каждого пикселя представлен тремя байтами, задающими, соответственно, значение красной, синей и зеленой компонентой цвета.

Так как по ряду причин существует несколько разновидностей формата BMP, ограничимся версией заголовка BMPINFOHEADER. Такой файл состоит из двух заголовков размером 14 и 40 байт соответственно, содержащих служебную информацию и массива цветов пикселей, представляющих построчную развертку изображения, начиная с нижней. Байты цвета пикселя идут в порядке BGR (синий, зеленый, красный).

Для целей оптимизации скорости обработки, авторами формата было предусмотрено, что каждая строка пикселей начинается с адреса, кратного четырем байтам, для чего справа добавляются "псевдопиксели", дополняющие ширину изображения до кратной четырем, что обрабатывается в приложенной к заданию программе. Также, в этой программе выполняется распределение цветовых компонентов пикселей по отдельным цветовым плоскостям, т.е. сначала мы храним все синие компоненты, затем все зеленые, затем все красные компоненты.

Подробнее о структуре файла можно прочитать [здесь](#).

В приложенной программе предложена реализация преобразования изображения в оттенки серого. Вам необходимо реализовать собственный фильтр в соответствии с заданием. По ссылке для каждого из заданий можно посмотреть небольшое описание и, в некоторых случаях, пример работы фильтра.

#### Обратите внимание!

Фильтр для реализации назначается преподавателем! Реализация фильтра должна представлять собой отдельный файл исходного кода с одной основной функцией фильтра и, возможно, несколькими вспомогательными функциями. Прототип основной функции фильтра должен быть помещен в заголовочный файл bmp.h.

Не забудьте нормализовать значения цветовых компонентов пикселей после применения преобразований!

В архиве размещены несколько тестовых изображений, часто используемых в анализе алгоритмов обработки изображений. Историю одного из самых знаменитых, можно почитать [здесь](#).

#### План работы

1. Подготовьте каталог task6 в каталоге inf.
2. В каталог inf/task6 распакуйте архив с кодом программы, приложенный к этому заданию
3. Изучите поведение программы на примере приложенных bmp-файлов.
4. Реализуйте функцию фильтра в соответствии с индивидуальным заданием из следующих ниже. Номер варианта необходимо **заранее** получить у преподавателя вашей группы.

1. Фильтр «Размывание» (Blur / Box Blur)

Значение каждого пикселя заменяется на 1 / 25 суммы произведений цветов соседних точек в пределах матрицы 5 x 5 на весовые коэффициенты в соответствии с матрицей:

1,	1,	1,	1,	1
1,	1,	1,	1,	1
1,	1,	1,	1,	1
1,	1,	1,	1,	1
1,	1,	1,	1,	1

Этот свёрточный фильтр создает эффект размытости изображения.

2. Фильтр «Размывание по Гауссу» (Gaussian Blur)

## 6) Изменения

### Вариант 2

1. Сформулировал задачу
2. Добавил описание выбранной платформы
3. Увеличил размер и жирность шрифта текста для лучшей читаемости с экрана
4. Уменьшил расстояния между блоками на иллюстрациях, для улучшения читаемости

### Вариант 3

1. Заменял фрагменты кода схемами
2. Расписал описание и анализ модуля
3. Выделил заголовки жёлтым цветом
4. Добавил нумерацию на слайды
5. Добавил слайд про материалы

### Вариант 4

1. Поставил практическую задачу
2. Изменил вывод

