



Обработка формата PNG

Общие требования - представлены [тут](#)

Программа должна иметь следующие функции по обработке изображений:

1. Разделяет изображение на $N \times M$ частей, а линии требуемой толщины должны рисоваться между ними. Расстояния между блоками - `thickenss` (то есть изображение будет расширено на количество линий по оси * толщину). При определении участка, в который будет использоваться крайний пиксель округление выполнять в соответствии с математическими правилами округления. Флаг для выполнения данной операции: `-split`. Реализация: провести линии заданной толщины. функциональность определяется:
 1. Количество частей по "оси" Y. Флаг `-number_x`. На вход принимает число больше 1
 2. Количество частей по "оси" X. Флаг `-number_y`. На вход принимает число больше 1
 3. Толщина линии. Флаг `-thickness`. На вход принимает число больше 0
 4. Цвет линии. Флаг `-color` (цвет задаётся строкой `rrr.ggg.bbb`, где `rrr/ggg/bbb` - числа, задающие цветовую компоненту. пример `-color 255.0.0` задаёт красный цвет)
2. Рисование прямоугольника. Флаг для выполнения данной операции: `-rect`. Он определяется:
 1. Координатами левого верхнего угла. Флаг `-left_up`, значение задаётся в формате `left.up`, где `left` - координата по x, `up` - координата по y
 2. Координатами правого нижнего угла. Флаг `-right_down`, значение задаётся в формате `right.down`, где `right` - координата по x, `down` - координата по y
 3. Толщиной линий. Флаг `-thickness`. На вход принимает число больше 0
 4. Цветом линий. Флаг `-color` (цвет задаётся строкой `rrr.ggg.bbb`, где `rrr/ggg/bbb` - числа, задающие цветовую компоненту. пример `-color 255.0.0` задаёт красный цвет)
 5. Прямоугольник может быть залит или нет. Флаг `-fill`. Работает как бинарное значение: флага нет - `false`, флаг есть - `true`.
 6. цветом которым он залит, если пользователем выбран залитый. Флаг `-fill_color` (работает аналогично флагу `-color`)
3. Сделать рамку в виде узора. Флаг для выполнения данной операции: `-ornament`. Рамка определяется:
 1. Узором. Флаг `-pattern`. Обязательные значения: `rectangle` и `circle`, `semicircles`. Также можно добавить свои узоры (красивый узор можно получить используя фракталы). Подробнее здесь: https://se.moevm.pro/doku.php/courses/programming:cw_spring_ornament
 2. Цветом. Флаг `-color` (цвет задаётся строкой `rrr.ggg.bbb`, где `rrr/ggg/bbb` - числа, задающие цветовую компоненту. пример `-color 255.0.0` задаёт красный цвет)
 3. Шириной. Флаг `-thickness`. На вход принимает число больше 0
 4. Количеством. Флаг `-count`. На вход принимает число больше 0
 5. При необходимости можно добавить дополнительные флаги для необозначенных узоров
4. Поворот изображения (части) на 90/180/270 градусов. Флаг для выполнения данной операции: `-rotate`. Функциональность определяется

1. Координатами левого верхнего угла области. Флаг `—left_up`, значение задаётся в формате `left.up`, где `left` – координата по `x`, `up` – координата по `y`
2. Координатами правого нижнего угла области. Флаг `—right_down`, значение задаётся в формате `right.down`, где `right` – координата по `x`, `down` – координата по `y`
3. Углом поворота. Флаг `—angle`, возможные значения: 90, 180, 270

From:

<https://se.moevm.pro/> - **МОЭВМ Вики** [se.moevm.pro]

Permanent link:

https://se.moevm.pro/doku.php/courses:programming:cw_vars:spring:5:5.11

Last update:

