

Описание полученного задания

Реализация контейнера, содержащего плоские геометрические фигуры, размещаемые в координатной сетке.

Типы фигур:

- **Круг** - имеет целочисленные координаты центра окружности, радиус, цвет, функцию вычисления площади.
- **Прямоугольник** - имеет целочисленные координаты левого верхнего угла, правого нижнего угла, цвет, функцию вычисления площади.
- **Треугольник** - имеет целочисленные координаты трех вершин, цвет, функцию вычисления площади.

Дополнительная функция - сортировка контейнера алгоритмом Шелла.

Параметры ввода:

- **Случайная генерация фигур:** `./hw1 -n number output_1_path.txt output_2_path.txt`, где параметр `number` отвечает за количество генерируемых фигур
- **Считывание параметров фигур из файла:** `./hw1 -f input_path.txt output_1_path.txt output_2_path.txt`

Формат описания фигур в файле: считывание строк идет попарно, каждые 2 строки описывают одну фигуру. На первой строке подается индекс фигуры, на второй - ее параметры.

Параметры фигур:

- **Прямоугольник** - индекс фигуры = 1, параметры передаются строкой вида $x_1\ y_1\ x_2\ y_2\ color_index$, где x_1, y_1 - координаты левой верхней вершины.
- **Треугольник** - индекс фигуры = 2, параметры передаются строкой вида $x_1\ y_1\ x_2\ y_2\ x_3\ y_3\ color_index$, где x_i, y_i - координаты i -ой вершины.
- **Круг** - индекс фигуры = 3, параметры передаются строкой вида $x_1\ y_1\ radius\ color_index$

Пример описания круга в файле:

```
3
-66 81 46 4
```

Параметр `color_index` должен принимать значение от 0 до 6, в противном случае цвет фигуры будет none

Структурная схема ВС с размещенной на ней программой

Таблица типов:

int	4 байта
double	8 байт
struct circle	20 байт
int x_1, x_2	8 байт[0, 4]
int radius	4 байта[8]
enum color	4 байта[12]
int color_index	4 байта[16]
struct rectangle	24 байта
int x_1, y_1, x_2, y_2	16 байт[0, 4, 8, 12]
enum color	4 байта[16]
int color_index	4 байта[20]
struct triangle	32 байта
int $x_1, y_1, x_3, y_3, x_3, y_3$	24 байта[0, 4, 8, 12, 16, 20]
enum color	4 байта[24]
int color_index	4 байта[28]
struct shape	40 байт
enum key	4 байта[0]
key k	4 байта[4]
circle c	20 байт[8]
rectangle r	24 байта[8]
triangle t	32 байта[8]
struct container	400048 байт
enum max_len	4 байта[0]
int len	4 байта[4]
shape cont	40 * 10001 = 400040 байт[8]

Память программы:

main()	
int argc	4 байта[0]
char** argv	8 байт[4]
container c	400048 байт[12]
int size	4 байта[400060]
Clear()	
int i	4 байта[0]
Out()	
int i	4 байта[0]
Area(triangle)	
double a, b, c, p	32 байта[0]
ShellSort()	
int i, d, j	12 байт[0]

stack:

main
Init
Init - end
In/InRnd
In/InRnd - end
Out
Out - end
ShellSort
ShellSort - end
Out
Out - end
main - end

Тестовые прогоны

Файловый ввод:

10000 элементов: 2.88854 сек.

5000 элементов: 0.729822 сек.

1000 элементов: 0.034942 сек.

500 элементов: 0.009173 сек.

100 элементов: 0.001 сек.

Генерация случайных данных:

10000 элементов: 2.537785 сек.

5000 элементов: 0.722036 сек.

1000 элементов: 0.032829 сек.

500 элементов: 0.009331 сек.

100 элементов: 0.001 сек.

Что интересно, под системой Windows программа работает почти в 2 раза медленнее.

Так же программа поддреживает обработку идеологически некорректных данных из файла(нулевой радиус, левый верхний угол ниже правого и тд)