

Домашнее задание по предмету «Архитектура вычислительных систем» №1

Выполнила Шелемех Е.В, группа БПИ206

1. Описание полученного задания

Номер варианта – 226 -> условие задачи – 2, номер функции – 17

Условие задачи:

2. Плоская геометрическая фигура, размещаемая в координатной сетке	1. Круг (целочисленные координата центра окружности, радиус) 2. Прямоугольник (целочисленные координаты левого верхнего и правого нижнего углов) 3. Треугольник (целочисленные координаты трех углов)	Цвет фигуры (перечислимый тип) = {красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый}	Вычисление периметра фигуры (действительное число)
--------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------	-------------------------------------------------------

Функция обработки данных в контейнере:

17. Упорядочить элементы контейнера по убыванию используя сортировку с помощью разделения (Quick Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Описание работы программы

Запуск программы осуществляется из командной строки, в которой указываются: имя запускаемой программы; имя файла с исходными данными; имя файла с выходными данными. В файле с исходными данными в первой строке должен находиться тип ввода данных: "random" или "file". Если тип ввода - "random", то во второй строке должно быть положительное число менее 10000, т.е кол-во генерируемых фигур. Если тип ввода - "file", то в последующих строках вводится информация о фигурах в формате:

- тип_фигуры (целое число от 1 до 3 включительно)
- данные о фигуре
 - если треугольник: цвет a_x a_y b_x b_y c_x c_y
 - если прямоугольник: цвет a_x a_y b_x b_y
 - если круг: цвет center_x center_y radius
- может быть пустая строка между вводом разных фигур, но не обязательно

цвет – целое от 0 до 6; a_x, a_y, b_x, b_y, c_x, c_y, center_x, center_y, radius – целые числа;

Результат работы программы, т.е вывод элементов контейнера до сортировки и вывод элементов контейнера после сортировки, помещается в выходной файл, указанный пользователем в качестве аргумента при запуске программы.

2. Архитектура

Используемые структуры, переменные и их типы

int	4
double	8
Struct rectangle	20
a_x: int	4[0]
a_y: int	4[4]
b_x: int	4[8]
b_y: int	4[12]
col: int	4[16]
Struct triangle	28
a_x: int	4[0]
a_y: int	4[4]
b_x: int	4[8]
b_y: int	4[12]
c_x: int	4[16]
c_y: int	4[20]
col: int	4[24]

Struct circle	16
center_x: int	4[0]
center_y: int	4[4]
radius: int	4[8]
col: int	4[12]
Struct shape	32
k: key	4[0]
union	
{ r: rectangle	20[4]
t: triangle	28[4]
c: circle}	16[4]
Struct container	
length: int	4[0]
enum max_len	4[0]
shape*cont[max_len=100	8*10000+4+
00]	4 = 80008

Глобальная память

В глобальной памяти нет переменных, следовательно, она пуста.

Локальная память

1. main()	
input_stream: FILE*	8[0]
cont: container	80008[8]
stream_out: FILE*	8[80016]
number_of_figures: int	4[80024]
input_type: char*	4[80028]
str: char[250]	250[80032]
2. Perimeter(...)	
<i>из triangle.cpp</i>	
triang: triangle	28[0]
ab: double	8[28]
ac: double	8[36]
bc: double	8[44]

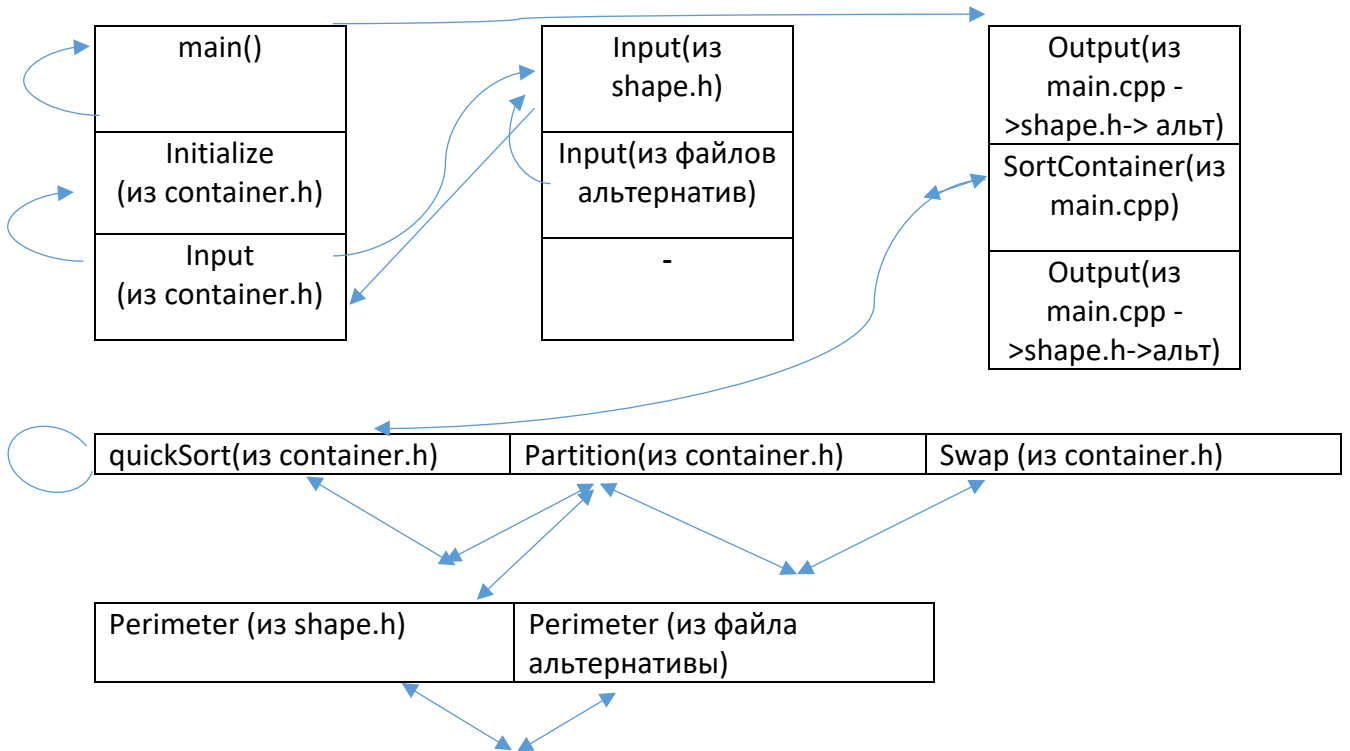
3. swap(...)	
<i>из container.cpp</i>	
a: shape*	4[0]
b: shape*	4[4]
t: shape	32[8]
4. partition(...)	
<i>из container.cpp</i>	
cont: container	80008[0]
low: int	4[80008]
high: int	4[80012]
basis: shape	32[80016]
basis_number: double	8[80048]
i: int	4[80056]

Куча

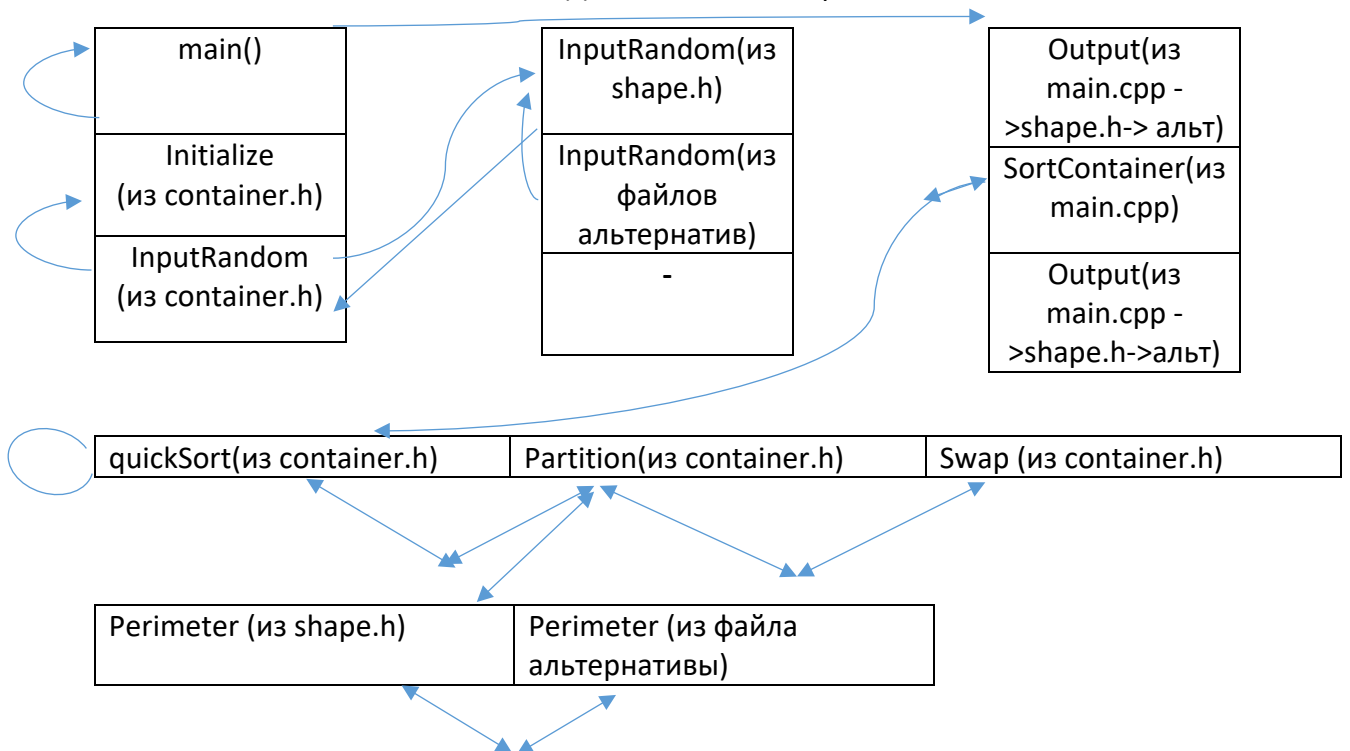
В куче хранятся элементы массива container: shape*[max_size] – указатели типа shape, так как они были созданы динамически с помощью new. Каждый элемент занимает 8 байт.

Стек вызовов

1. Данные вводятся из файла



2. Данные вводятся случайно



3. Характеристики

Количество заголовочных файлов = 6

Количество модулей реализации = 6

Общий размер исходных текстов программы = 19.6 Кб

Размер исполняемого файла = 148 Кб

Время выполнения программы(на приложенных тестовых наборах):

Входной файл	Тип ввода данных	Описание тестового набора	Выходной файл	Время выполнения
tests\test1.txt	Из файла	8 элементов	tests\output_test1.txt	9мс
tests\test2.txt	Из файла	11 элементов	tests\output_test2.txt	5мс
tests\test3.txt	рандом	66 ранд. элементов	tests\output_test3.txt	7мс
tests\test4.txt	рандом	Некорректное число элементов	tests\output_test4.txt	3мс
tests\test5.txt	Некорректный тип ввода	-	tests\output_test5.txt	1мс
tests\test6.txt	рандом	1000 ранд. элементов	tests\output_test6.txt	15мс