

# Отчет

## Архитектура вычислительной системы

### Описание задания

#### Вариант 211

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
1. Плоская геометрическая фигура, размещаемые 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 площади фигуры (действительное число)

16. Упорядочить элементы контейнера по убыванию используя сортировку методом деления пополам (Binary Insertion). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

### Описание работы программы

Во входном файле tests.txt расположены все тесты, при выборе опции “считывание из файла” обрабатываются сразу все тесты, и в выходной файл выводится результат: сначала все фигуры контейнера по порядку, а затем отсортированный контейнер. При выборе опции “рандомный ввод” пользователь должен будет ввести seed, и будет сформирован один контейнер с рандомными фигурами, его элементы будут выведены сначала по порядку, а затем в соответствии со значением площади (по убыванию).

Формат одного теста:

<количество элементов в контейнере>

<ключ фигуры – целое число от 1 до 3, 1 – прямоугольник, 2 – треугольник, 3 - круг>

<параметры фигуры>

Если треугольник, то: <ax> <ay> <bx> <by> <cx> <cy> <цвет – целое число от 1 до 7>

Если прямоугольник, то: <left\_up\_x> <left\_up\_y> <right\_down\_x> <right\_down\_y> <цвет>

Если круг, то: <center\_x> <center\_y> <radius> <цвет>

Далее с новой строки ключ следующей фигуры и с новой строки ее параметры, и так далее, пока количество фигур не будет равно введенному в самой первой строке количеству.

### Таблица типов

int	4
double	8

<b>class</b> <b>rectangle</b> left_up_x: int left_up_y: int right_down_x: int right_down_y: int	32  4[0] 4[4] 4[8] 4[12] + 16 из-за наследования от figure
<b>class triangle</b> ax: int ay: int bx: int by: int cx: int cy: int	40 4[0] 4[4] 4[8] 4[12] 4[16] 4[20] + 16 из-за наследования от figure
<b>class circle</b> center_x: int center_y: int radius: int	32 4[0] 4[4] 4[8] + 16 из-за наследования от figure и выравнивание до 32
<b>class figure</b> k: key color: color	16 4[0] 4[4] Виртуальный конструктор - +8
<b>class container</b> len: int container: figure*[max_size] max_size: int	80008 4[0] 8*10000=80000[4] 4[80004]

### Глобальная память

Глобальная память пустая, нет объявленных глобальных переменных.

### Локальная память

<b>main()</b> file_in: ifstream size: int cont: container file_out: ofstream command_number: int seed: int	520[0] 4[520] 80008[524] 512[80532] 4[81044] 4[81048]
------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------

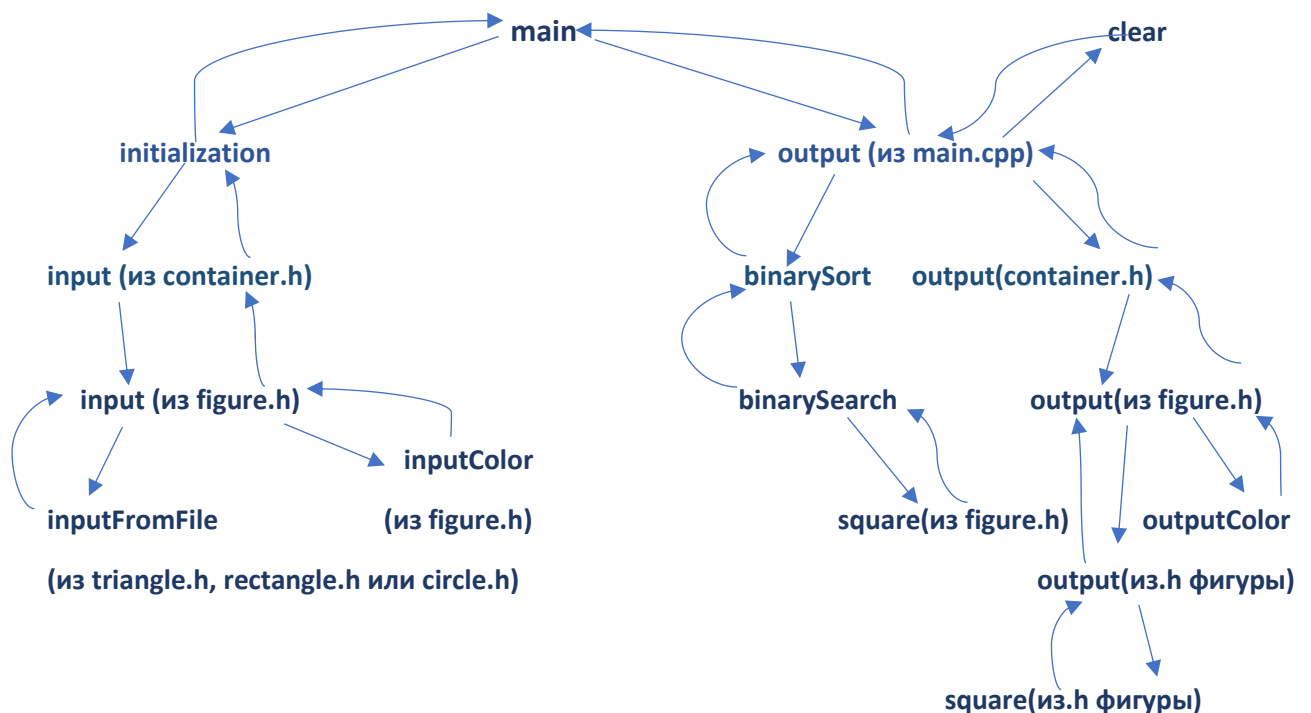
<b>initialization(...)</b> size: int	4[0]
<b>binarySort(..)</b> size: int i: int position: int right_bound: int selected_figure: figure*	4[0] 4[4] 4[8] 4[12] 8[16]
<b>binarySearch(...)</b> fig: figure* left: int right: int middle: int	8[0] 4[8] 4[12] 4[16]

## Heap

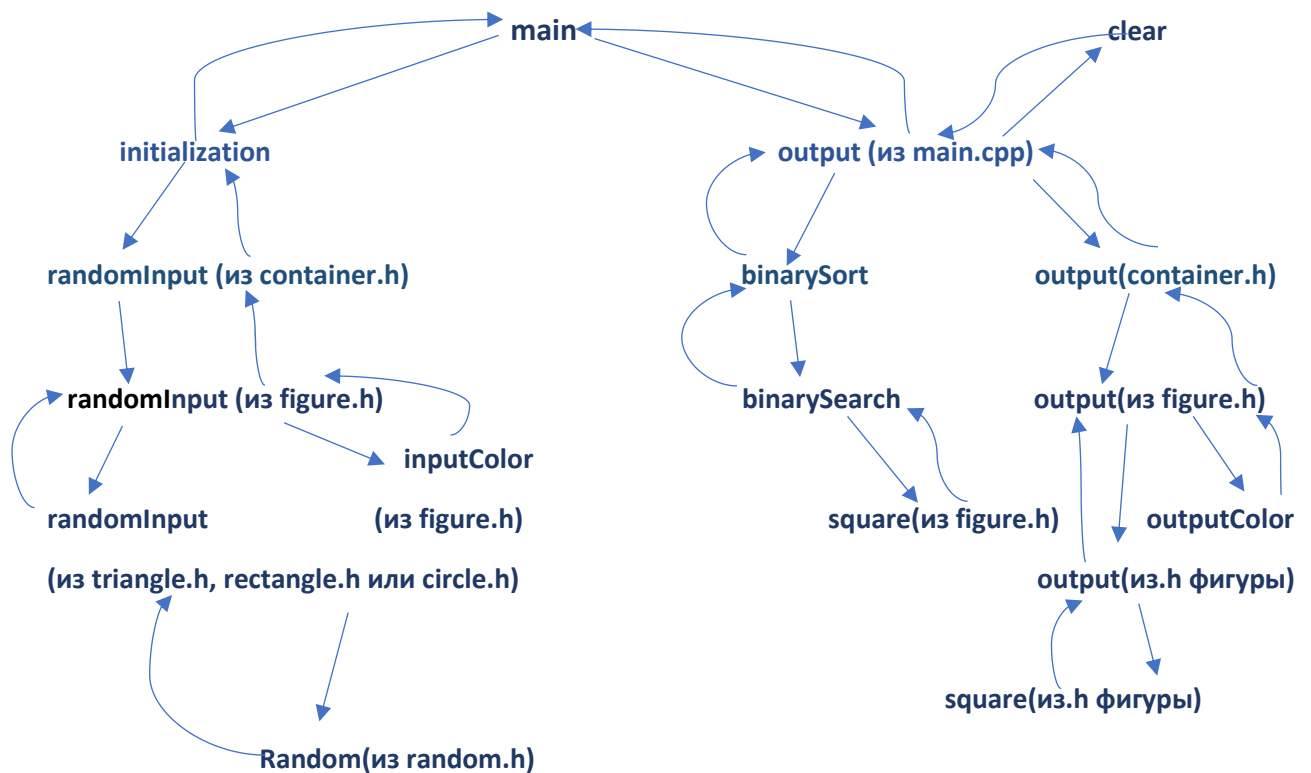
В Heap хранятся элементы массива `cont.container: figure*[max_size]` – указатели типа `figure`, в которые помещаются указатели на объекты производных классов класса `figure` (`rectangle`, `triangle`, `circle`). Под каждый такой элемент выделяется 8 байт динамической памяти.

## Stack вызовов функций

1) При вводе из файла



2) При случайном вводе



## Характеристики программы

Число заголовочных файлов – 6

Число файлов реализации – 6

Размер исполняемого файла – 616 КБ

Размер исходных текстов программы – 27.1 КБ

Время:

Случайное заполнение:

Seed = 5 - 15625 ms

Seed = 10 – 78125 ms

Seed = 100 – 31250 ms

Seed = 1000 – 46875 ms

Seed = 10000 – 15625 ms

Тесты из файла tests.txt:

Время выполнения всех тестов из приложенного тестового файла – 15625 ms

## **Сравнительный анализ**

**Сходства с процедурным подходом:** одинаковый стек вызовов функций, пустая глобальная память, одинаковый размер класса container

**Отличия от предыдущей реализации:** количество заголовочных файлов (убрала структуру point), из-за появления виртуальных деструкторов и наследования изменились размеры типов figure, rectangle, triangle, circle. В контейнере теперь хранятся указатели на типы rectangle, triangle, circle, приведенные к типу figure\*. Появился полиморфизм и переопределение методов, появилась инкапсуляция.