

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

В папке input находятся файлы с тестами (input1.txt input2.txt, ...). В командной строке необходимо ввести путь до входного файла и до выходного, разделенные пробелом, например, если папка input находится в папке проекта, входные данные будут выглядеть следующим образом: input\input1.txt output.txt. При выборе опции "считывание из файла" обрабатывается тест из входного файла, и в выходной файл выводится результат: сначала все фигуры контейнера по порядку, а затем отсортированный контейнер. При выборе опции "рандомный ввод" 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:

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

<параметры фигуры>

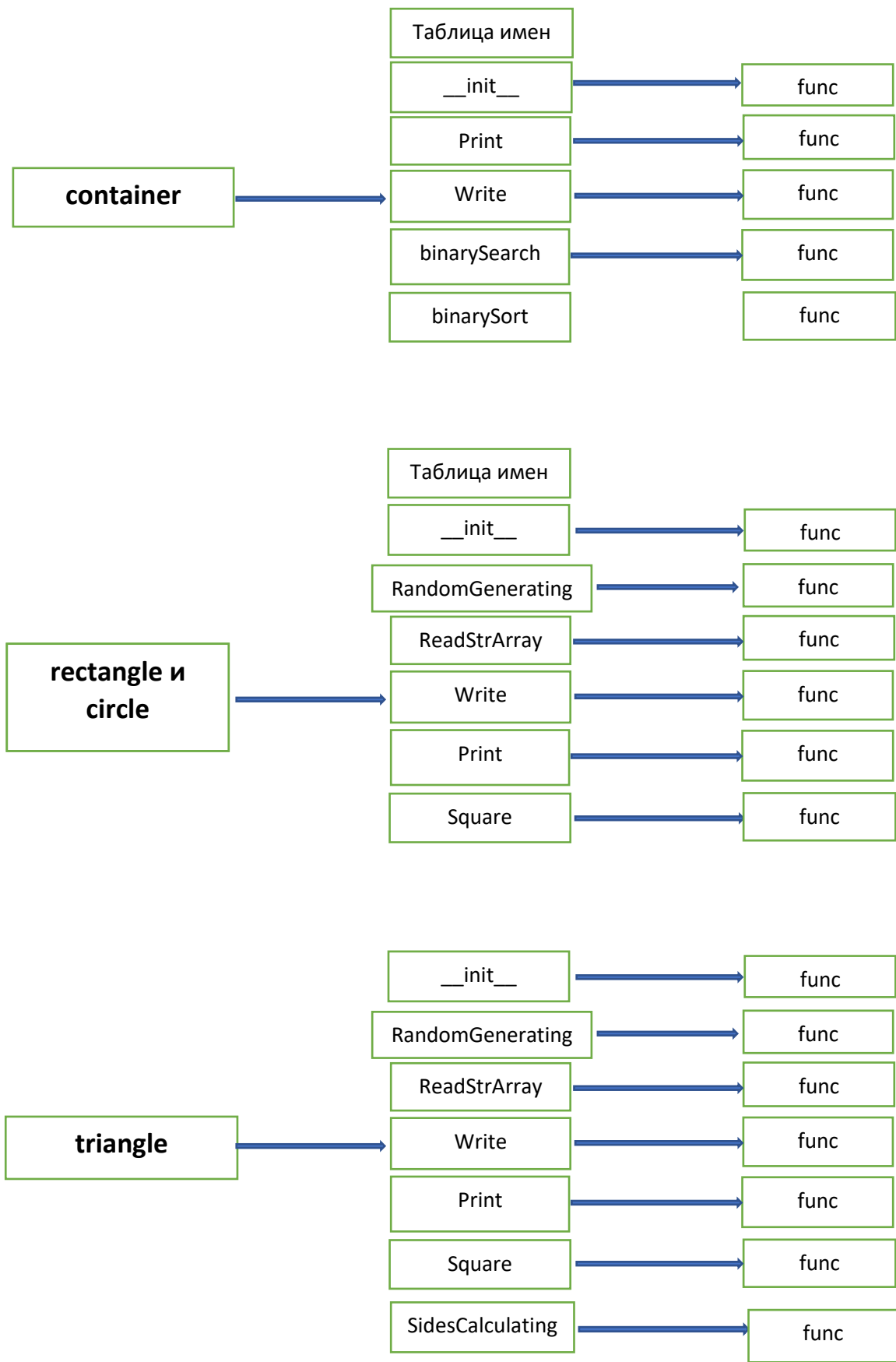
Если треугольник, то: <цвет – целое число от 1 до 7> <ax> <ay> <bx> <by> <cx> <cy>

Если прямоугольник, то: <цвет> <left_up_x> <left_up_y> <right_down_x> <right_down_y>

Если круг, то: <цвет> <center_x> <center_y> <radius>

Далее с новой строки ключ следующей фигуры и с новой строки ее параметры, и так далее.

Отображение содержимого классов **Container**, **Rectangle**, **Triangle**, **Circle**

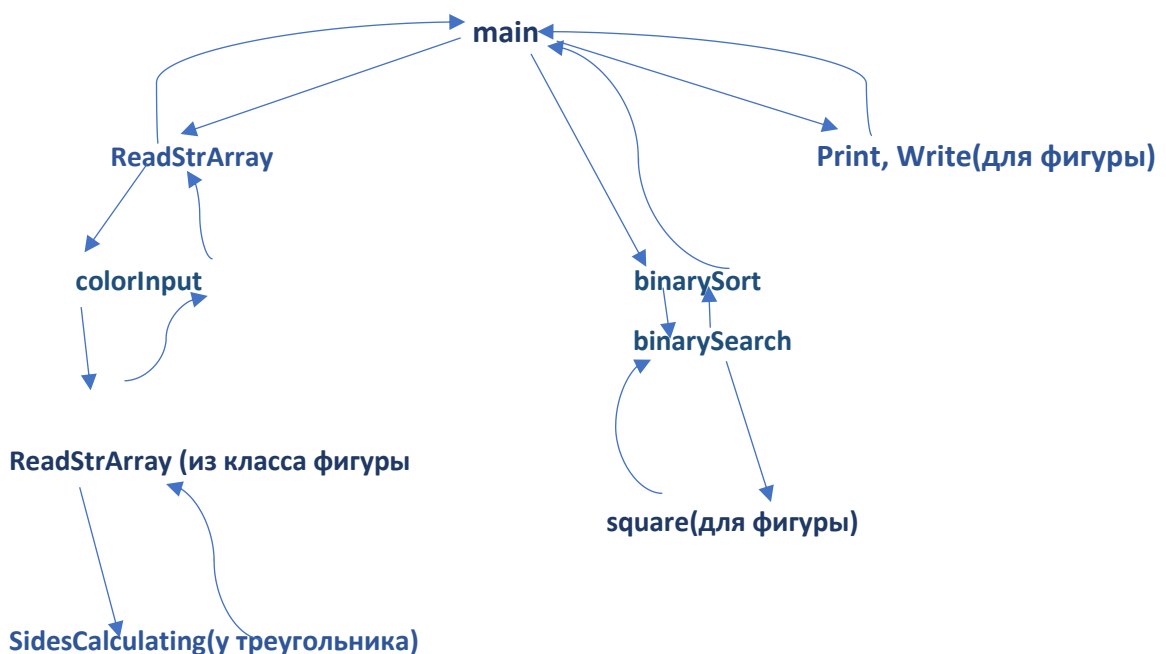


Отображение некоторых методов классов на память

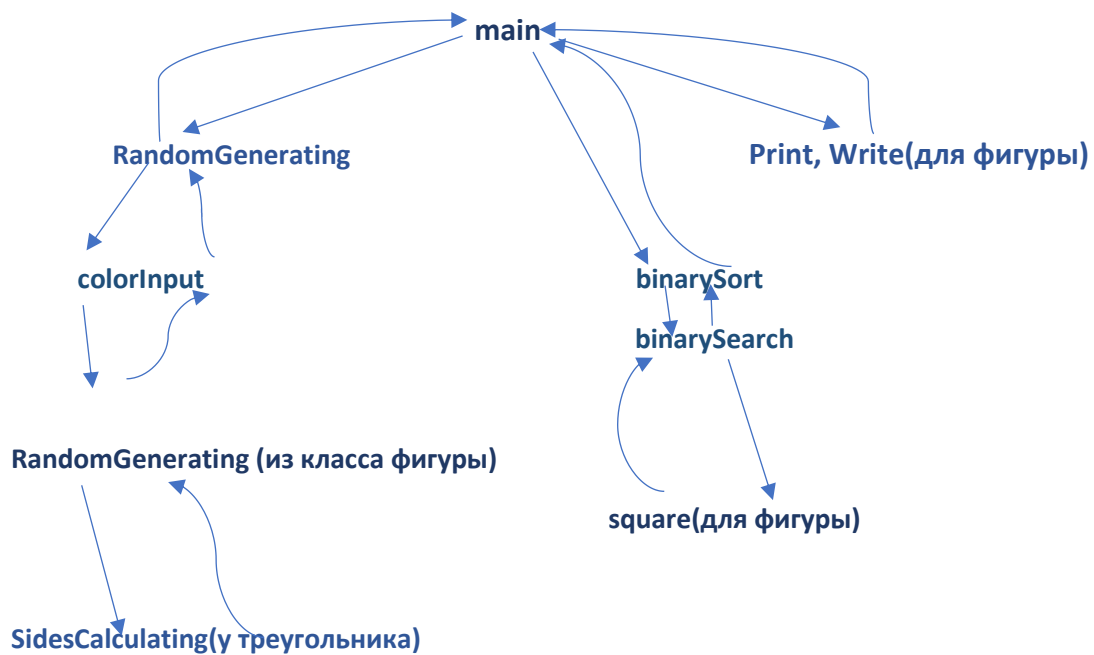
main.py ifile str strArray container ofile command_number figNum	file – Filename list – [...] list – [...] container -container.py file – Filename int - <number> int - <number>
container.binarySearch fig left right middle	shape – shape.py int - <number> int - <number> int - <number>
container.binarySort size i position right_bound selected_figure	Int - <number> Int- <number> Int - <number> Int - <number> Shape – shape.py
container.__init__ store	list – [...]

Stack вызовов функций

1) При вводе из файла



2) При случайном вводе



Характеристики программы

Число заголовочных файлов – 0

Число файлов реализации – 9

Размер исходных текстов программы – 14.5 КБ

Случайное заполнение:

97 элементов – 1.90363 s

88 элементов – 1.88191 s

39 элементов 13.1419 s

14 элементов 2.17994 s

5754 элементов – 3.18884 s

1129 элементов – 2.32473 s

Тесты из папки input:

Test 1	1.88728 s
Test 2	1.99841 s
Test 3	1.75441 s
Test 4	1.79149 s
Test 5	1.85115 s
Test 6	2.32263 s
Test 7	1.7617 s

Test 8	2.03015 s
Test 9	1.77128 s
Test 10	1.86829 s

ООП:

Test 1	0.015625 s
Test 2	0.015625 s
Test 3	0 s
Test 4	0.015625 s
Test 5	0.031250 s
Test 6	0.015625 s
Test 7	0.031250 s
Test 8	0.015625 s

4771 элементов – 0.03125 s
7801 элементов – 0.078125 s
8973 элементов – 0.125 s
137 элементов – 0.015625 s

Процедурное:

7801 элементов – 0.093750 s
3473 элементов – 0.015625 s
4196 элементов – 0.031250 s
137 элементов – 0.0000001 s
6430 элементов – 0.046875 s

Сравнение

1. Программа на питоне работает намного медленнее, чем в предыдущих реализациях на С и С++
2. Непривычно не указывать типы переменных, тем не менее, код стал выглядеть намного проще
3. Отсутствие фигурных скобок тоже непривычно
4. Размер исходных текстов программы стал меньше