

## Пояснительная записка

### Описание полученного задания

**Вариант: 282**

**Начальное условие задач: 2.**

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
2. 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 периметра фигуры (действительное число)

**Функция обработки данных в контейнере: 21.**

Удалить из контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив меньше, чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы передвинуть к началу контейнера с сохранением порядка.

### Требуемые метрики, определяющие характеристики программы, для различных тестовых прогонов.

Программа содержит 6 интерфейсных модулей (заголовочных файлов):

- Headers/random.h
- Headers/triangle.h
- Headers/rectangle.h
- Headers/shape.h
- Headers/container.h
- Headers/circle.h

А также 6 модулей реализации (файлов с определением программных объектов):

- Sources/triangle.cpp
- Sources/rectangle.cpp
- Sources/shape.cpp
- Sources/container.cpp

- Sources/main.cpp
- Sources/circle.cpp

Общий размер исходных текстов: 18,5 КБ.

Полученный размер исполняемого кода: 49,5 КБ.

Время выполнения программы для различных тестовых наборов данных:

Количество фигур	Время работы программы при вводе данных из файла	Время работы программы при случайной генерации данных
1 фигура	1 ms	1 ms
3 фигуры	2 ms	3 ms
8 фигур	4 ms	4 ms
12 фигур	4 ms	4 ms
14 фигур	5 ms	4 ms
100 фигур	11 ms	8 ms
1000 фигур	38 ms	32 ms
10000 фигур	195 ms	202 ms

Программа была протестирована на 12 тестах. Все тесты расположены в папке проекта «...\CSA-HW1\cmake-build-debug\tests» и имеют название «test<i>.txt», где i – числа от 0 до 11. Причём некорректный ввод проверяют файлы «test1.txt» (некорректно заданные фигуры), «test7.txt» (пустой контейнер), «test8.txt» (переполнение контейнера). При некорректном цвете фигуры, её координатах программа генерирует их случайным образом. Тесты «test9.txt», «test10.txt», «test11.txt» написаны для получения времени работы программы на 10000, 100 и 1000 фигурах соответственно.

Результаты прохождения i-го теста записываются в два файла: «output<i>.txt» и «output\_deleted<i>.txt». В первом из них содержится информация о контейнере, во втором информация о контейнере после применения функции, данной в задании. Файлов «output7.txt» и «output\_deleted7.txt», «output8.txt» и «output\_deleted8.txt» нет, так как на соответствующих им тестах программа в виду некорректных данных заканчивает работу. Также проект содержит два файла: «outputR.txt» и «output\_deletedR.txt», которые были получены в ходе тестирования программы на случайно сгенерированном контейнере. Все файлы с результатами работы находятся в папке проекта «...\CSA-HW1\cmake-build-debug\outputs».

### Формат аргументов командной строки и тестов

Для тестирования программы на тестах необходимо ввести следующую команду (i – число от 1 до 11):

```
-f tests/test<i>.txt outputs/output<i>.txt outputs/output_deleted<i>.txt
```

Для тестирования программы на случайно сгенерированных тестах необходимо ввести следующую команду:

```
-n <number> outputs/outputR.txt outputs/output_deletedR.txt
```

Также в обоих случаях допустимо указание других файлов вывода.

Фигуры в тестах написаны в следующем формате:

<figure\_number> <color\_number> <coordinates>

Номер фигуры – цифра от 1 до 3. Цифра 1 соответствует кругу, 2 – прямоугольнику, 3 – треугольнику. Номер цвета – цифра от 0 до 6, где каждая цифра соответствует цвету радуги по порядку. Координаты – числа, количество которых для каждой фигуры своё. У круга 3 числа в следующем порядке: x-координата центра, y-координата центра и радиус. У прямоугольника 4 числа в порядке x-координата левого верхнего угла, y-координата левого верхнего угла, x-координата правого нижнего угла, y-координата правого нижнего угла. У треугольника 6 чисел, аналогично по порядку три (x, y) координаты вершин.

### Структурная схема изучаемой архитектуры ВС с размещенной на ней разработанной программы

В стек (stack) помещаются вызовы методов, примитивные типы. Небольшой фрагмент стека представлен на рисунке.

В кучу (heap) помещаются динамические массивы (\*cont, \*argv), экземпляры структур (экземпляры структур shape, circle, rectangle, triangle). Фрагмент кучи представлен на рисунке.

В памяти программы описана функция main(argc, argv) и ещё четыре функции, представленные на рисунке.

