

Пояснительная записка к домашнему заданию 1

Тимохина Софья Константиновна, БПИ207

Описание задания

Вариант 267, задание 1, функция 20.

Программа должна содержать следующие структуры:

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)
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)

Общая для всех альтернатив переменная - цвет фигуры, перечислимый тип = {красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый}.

Общая для всех альтернатив функция: вычисление площади фигуры (действительное число).

Дополнительная функция: удаление из контейнера тех элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, то есть функция вычисления площади, больше, чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы передвинуть к началу контейнера с сохранением порядка.

Также нужно:

1. Провести отладку и тестирование разработанной программы на заранее подготовленных тестовых наборах данных. Количество тестовых наборов данных – не менее пяти. Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000. При необходимости программа должна правильно обрабатывать переполнение по данным. Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные формируемые генератором случайных наборов должны поддерживать

допустимые значения. Управление вводом данных задается из командной строки.

2. Описать структуру используемой ВС с наложением на нее обобщённой схемы разработанной программы.
3. Зафиксировать для отчета основные характеристики программы, такие как: число интерфейсных модулей (заголовочных файлов) и модулей реализации (фалов с определением программных объектов), общий размер исходных текстов, полученный размер исполняемого кода (если он формируется), время выполнения программы для различных тестовых наборов данных.

Структурная схема архитектуры ВС

Таблица типов

Название	Размер		Название	Размер
int	4		struct container { int len shape cont[max_len] }	32004 4[0] 32000[4]
double	8		struct circle { int radius point center }	12 4[0] 8[4]
long	8		struct triangle { point p1 point p2 point p3 }	24 8[0] 8[8] 8[16]
ifstream			struct rectangle { point p1 point p2 }	16 8[0] 8[0]
ofstream			enum color {}	4
string	8		enum key {}	4
clock_t	8			
struct shape { color col key k union { rectangle r, triangle t, circle c } }	32 4[0] 4[4] 16[8] 24[8] 12[8]			
struct point { int x int y }	8 4[0] 4[4]			

Описание работы функции main (main.cpp)

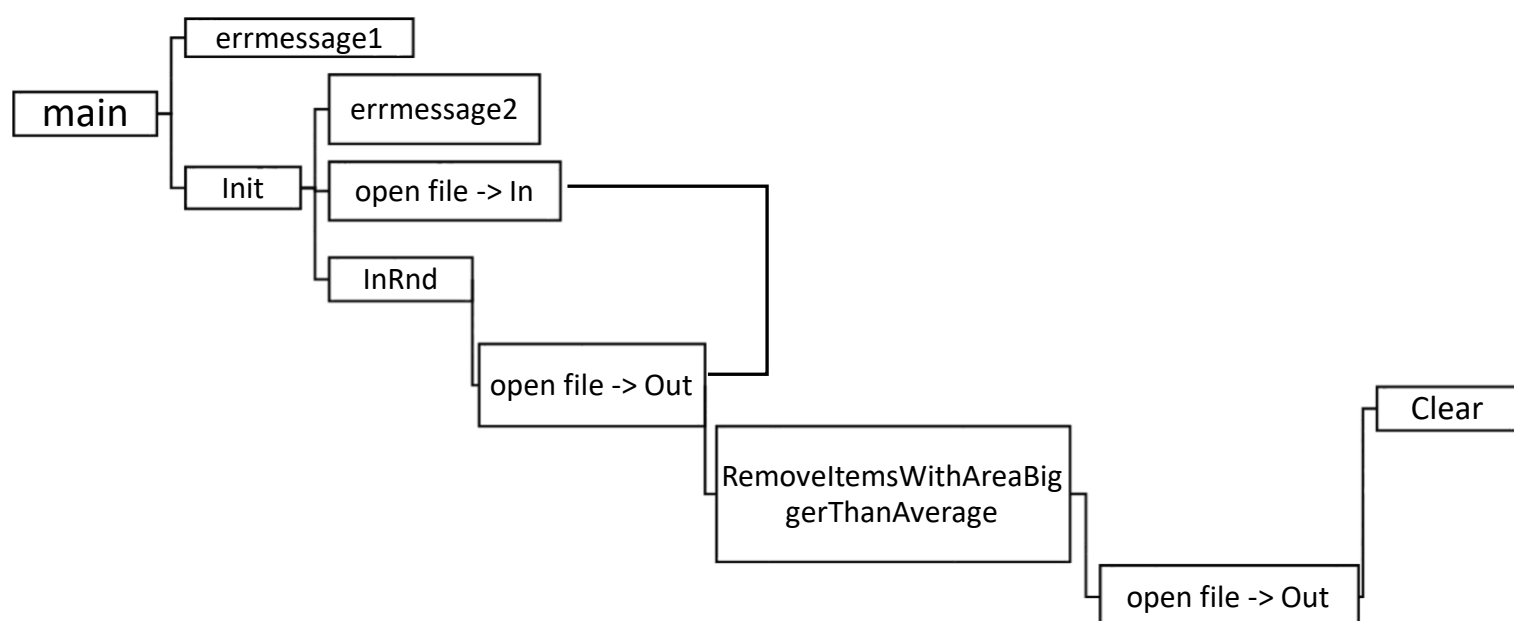
Память программы
int main(int argc, char* argv[])

Stack
Init
InRnd
Out
RemoveItemsWithAreaBigger ThanAverage
Out
Clear

Память данных	Размер (байты)
int argc	4[0]
char* argv[]	8[4]
container c	32004[12]
clock_t startTime	8[32016]
int size	4[32024]
ifstream ifst	
ofstream ofst1	
ofstream ofst2	

Heap	
Ввод из файла	Генерация
main	
-f	-n
in.txt	int size
out1.txt	
out2.txtx	

Блок-схема возможного стека в результате работы функции main (main.cpp)



Описание работы функции

RemoveItemsWithAreaBiggerThanAverage

Функция
<pre>void RemoveItemsWithAreaBiggerThanAverage(container &c) { int currentLen = 0; double averageArea = AverageArea(c); for (int i = 0; i < c.len; i++) { if (Area(*(c.cont[i])) <= averageArea) { c.cont[currentLen] = c.cont[i]; ++currentLen; } else { delete c.cont[i]; } } c.len = currentLen; }</pre>

Stack
RemoveItemsWithAreaBiggerThanAverage
AverageArea
Area

Память данных	Размер (байты)
container& c	8[0]
double averageArea	8[8]
int currentLen	4[16]
int i	4[20]

Heap
container

Описание работы функции AverageArea

Функция

```
double AverageArea(container &c) {  
    double sum = 0;  
    for (int i = 0; i < c.len; i++) {  
        sum += Area(*(c.cont[i]));  
    }  
    return sum / c.len;  
}
```

Stack

Area

Память данных

Размер
(байты)

container& c

8[0]

double sum

8[8]

int i

4[16]

Heap

container

Описание работы функции InRnd (shape.cpp)

Функция

```
shape *InRnd() {
    shape *sp;
    int k = rand() % 2 + 1;
    int color = rand() % 7 + 1;
    switch(k) {
        case 1:
            sp = new shape;
            sp->k = shape::RECTANGLE;
            InRnd(sp->r);
            SetColor(sp, color);
            return sp;
        case 2:
            sp = new shape;
            sp->k = shape::TRIANGLE;
            InRnd(sp->t);
            SetColor(sp, color);
            return sp;
        case 3:
            sp = new shape();
            sp->k = shape::CIRCLE;
            InRnd(sp->c);
            SetColor(sp, color);
            return sp;
        default:
            return 0;
    }
}
```

Стек	
rand	
SetColor	
InRnd	

Память данных	Размер (байты)
shape* sp	8[0]
int k	8[8]
int color	4[16]

Куча
shape [0]

Основные характеристики программы

Число заголовочных файлов: 8

Число исходных файлов: 7

Общий размер кода: 667 строк (23 КБ)

Время выполнения программы для различных тестов:

Номер теста	Время (с)
1	0.000691
2	0.000793
3	0.000596
4	0.000736
5	0.000772