

ОПИСАНИЕ ЗАДАНИЯ (задание 2, дополнительная функция 20):

Разработать программный продукт с использованием процедурного подхода и статической типизацией. Программа должна содержать следующие структуры:

Обобщенный артефакт, используемый в задании	Базовые альтернативы
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)

Для всех альтернатив общей переменной является **цвет** (перечислимый тип). Он может принимать значения: красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый. Общей функция всех альтернатив выступает вычисление периметра фигуры (действительное число). В качестве дополнительной функции контейнера необходимо удалять из него те фигуры, периметр которых больше чем среднее арифметическое периметров всех фигур контейнера (остальные фигуры передвигать к началу контейнера с сохранением порядка).

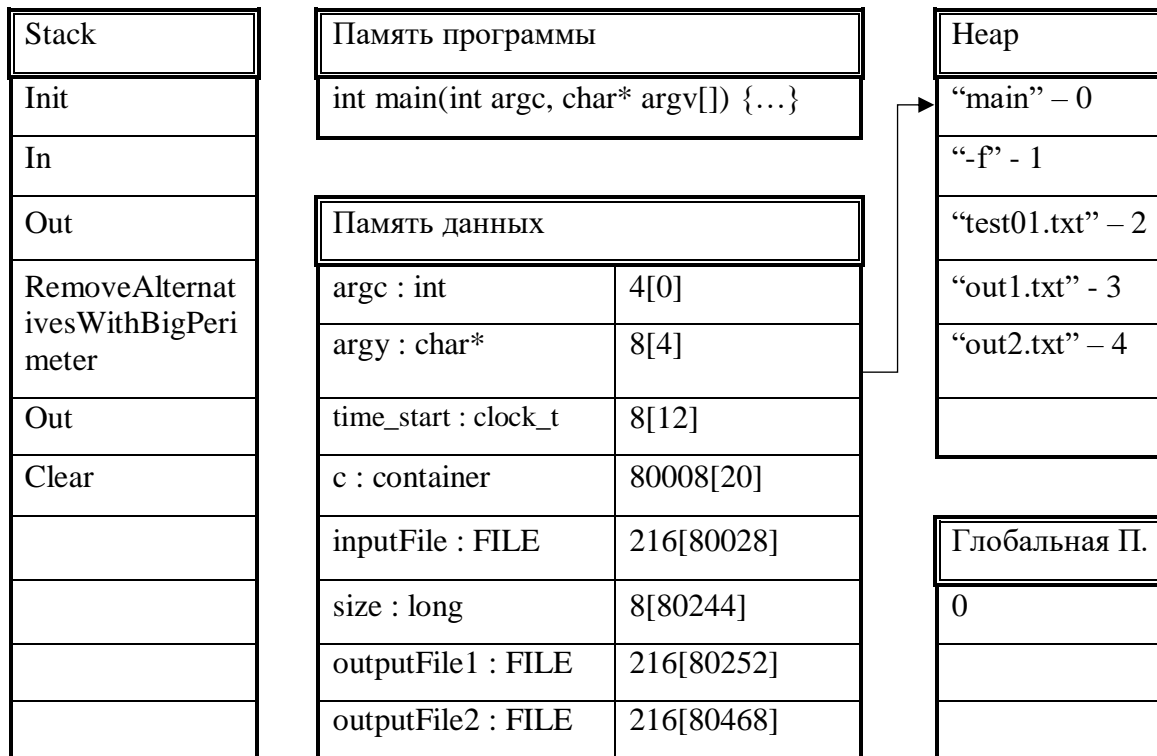
Также нужно: разработать тестовые входные данные и провести тестирование и отладку программы на этих данных (при необходимости, программа должна правильно обрабатывать переполнение по данным); описать структуру используемой ВС с наложением на нее обобщенной схемы разработанной программы; зафиксировать количество заголовочных файлов, программных файлов, общий размер исходных текстов, полученный размер исполняемого кода (если он формируется), время выполнения программы для различных тестовых наборов данных.

СТРУКТУРНАЯ СХЕМА АРХИТЕКТУРЫ ВС С ПРОГРАММОЙ:

Программа разработана в 64 битной системе (дистрибутив Ubuntu на ядре Linux).

Таблица типов				
Название	Размер		Название	Размер
int	4		struct circle	12
bool	1		center : point	8[4]
double	8		r : int	4[0]
long	8		struct rectangle	16
FILE	216		point1 : point	8[0]
char*	8		point2 : point	8[8]
enum color { }	4		struct triangle	24
enum key { }	4		a : point	8[0]
struct container	80008		b : point	8[8]
len : int	4[0]		c : point	8[16]
cont : *shape[max_len]	80000[4]		clock_t	8
struct point	8			
x : int	[0]			
y : int	[4]			
struct shape	32			
col : enum color	4[0]			
k : enum key	4[4]			
union {				
r : rectangle	16[8]			
t : triangle	24[8]			
c : circle	12[8]			
}				

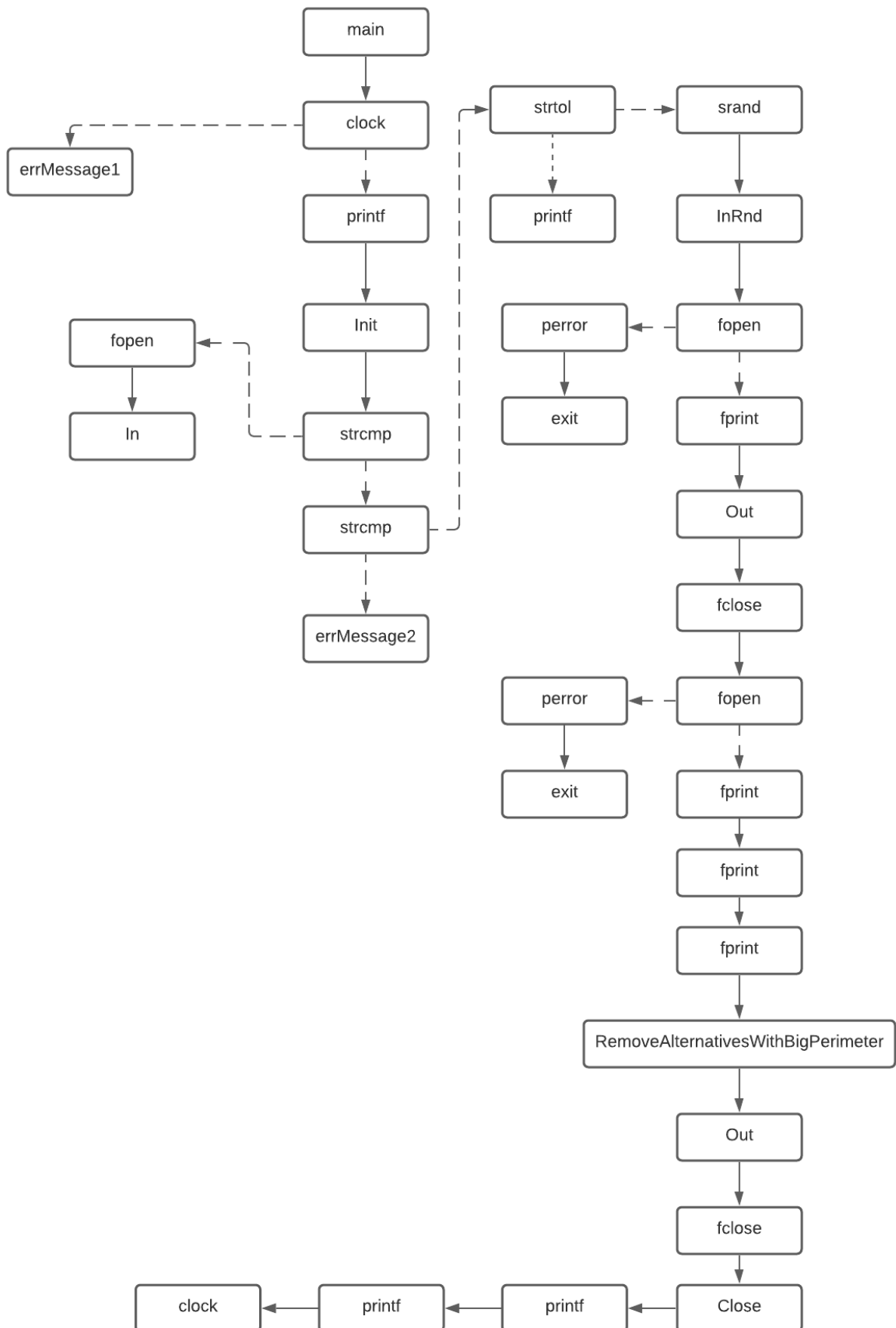
Описание работы функции main в рамках архитектуры:



Примечание:

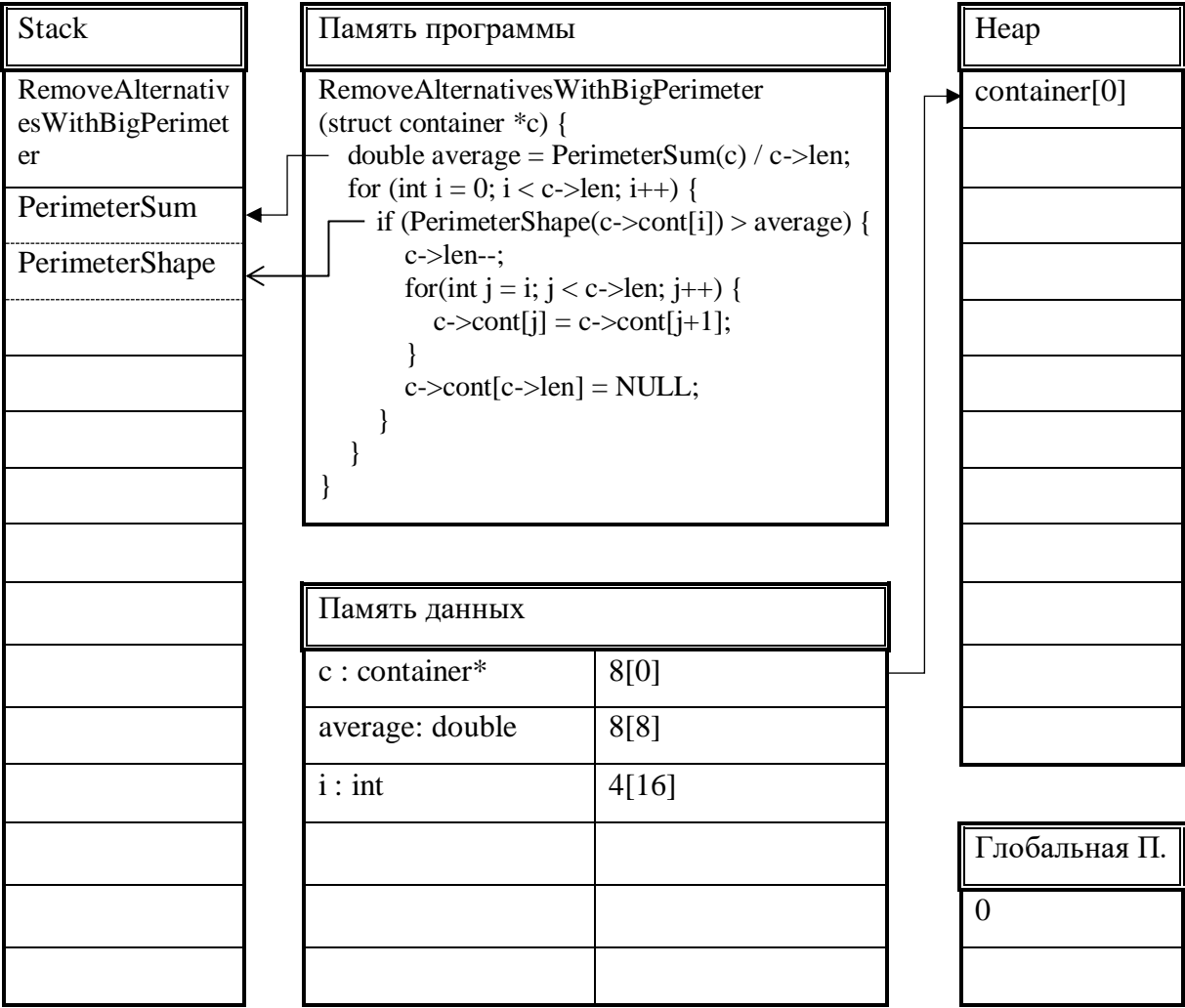
1. В таблице стек представлен для частного случая (загрузка данных из файла) и содержит только созданные в процессе написания программы функции). Все вариации возможного стека вызовов(с глубиной в 1 шаг) изображены на следующей странице в виде блок схемы.
2. Под {...} подразумевается код функции main (он написан в исходном файле main.c).

Блок схема возможного стека, в результате работы функции main (с глубиной в 1 шаг):



*Пунктирными стрелками обозначена вариативность стека.

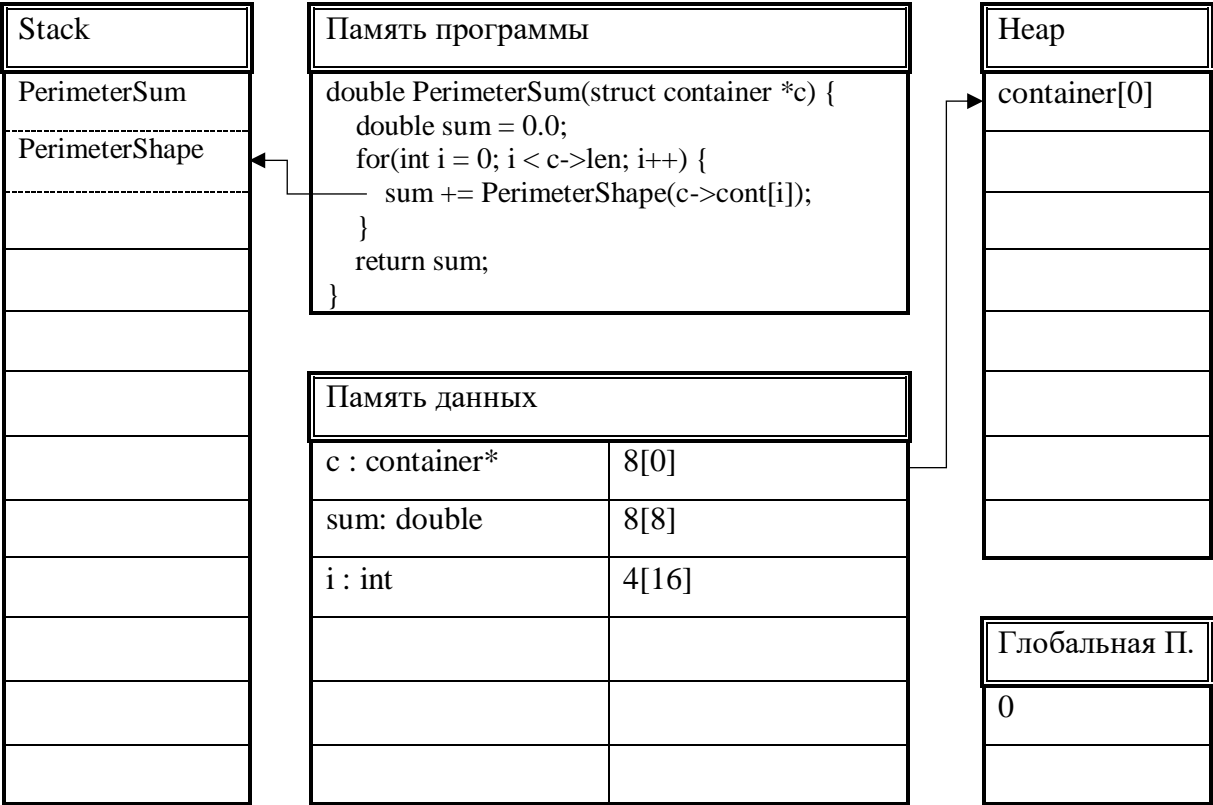
Описание работы для функции RemoveAlternativesWithBigPerimeter в рамках архитектуры:



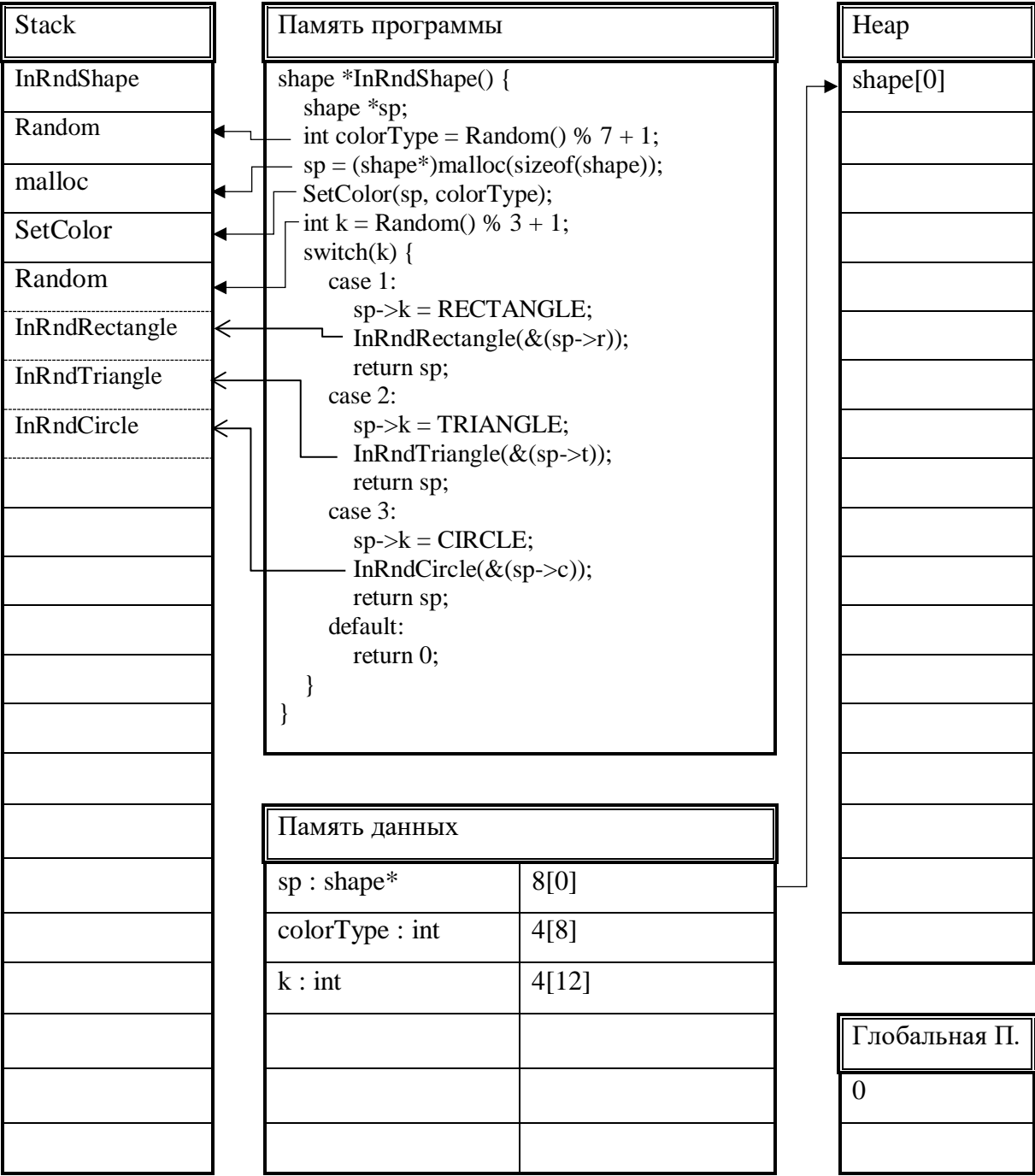
Примечание:

В данной (и последующих) таблицах стек изображен полностью (вариативная часть отображается выделением функции пунктирными линиями).

Описание работы для функции PerimeterSum в рамках архитектуры:



Описание работы для функции InRndShape в рамках архитектуры:



ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

- Число заголовочных файлов – 8
- Число модулей реализации – 8
- Общий размер исходных текстов – 470 строк кода (14.8 кб).
- Размер исполняемого кода – 41 кб.
- Время выполнения программы для различных тестовых прогонов:

Номер теста	Время выполнения в секундах
Тест1	0.000349
Тест2	0.000385
Тест3	0.000358
Тест4	0.000407
Тест5	0.000367