

## **Отчет по программе, разработанной на языке C.**

( Имитация бестипового решения с удалением АД и с функциональной декомпозицией по модулям)

### **Структура проекта :**

02 как надо\tests --- текстовые файлы с входными данными.

02 как надо\main.c --- вход в приложение

### **Спецификация:**

BC :

Operation System: Arch Linux

Architecture: x64

RAM: 32 Гб

Средства Разработки:

IDE: Ms Visual Studio

Библиотеки:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#include <string.h>
```

Средство сборки: CMake

Флаги:

-random

Предназначен для получения входных данных путем реализации рандомайзера

// Пример : command -random 100 out1.txt out2.txt

При данном вводе будет создано 100 случайно сгенерированных (через рандомайзер, реализованный в программе) объектов. Они будут разделены на типы, затем записаны в файл out1.txt. После чего произойдет селекция по убыванию и результат будет записан в out2.txt.

-tests

Предназначен для получения входных данных через передаваемые текстовые файлы

// Пример :command -tests test02.txt out1.txt out2.txt

При данном вводе будут считаны все входные данные из файла tests.txt

построчно. Строка должна начинаться с цифры 1-3, иначе она будет пропущена.

Каждая строка должна быть разделена определенным символом, зависящим от начальной цифры. И каждая строка должна заканчиваться точкой. Пробелы в строках учитываются.

Пример:

1 It is First string | otvet.

Для строки типа 1 это символ | 2

It is Second string - otvet.

Для строки типа 2 это символ -

3 It is the 3rd string ? otvet.

Для строки типа 3 это символ ?

Далее они будут разделены на типы, затем записаны в файл out1.txt. После чего произойдет селекция по убыванию и результат будет записан в out2.txt.

### ***Характеристики проекта:***

Количество заголовочных файлов: 2

Количество программных объектов: 5

Размер исходных файлов: ~ 15 Кб

Размер исполняемого файла: ~ 95 Кб

Время выполнения программы для различных входных данных  
(для флага -random) :

-random 10	===	0.0007 sec
-random 100	===	0.0015 sec
-random 1000	===	0.0095 sec
-random 10000	===	0.7451 sec

	Время	Размер
<b>процедурный подход</b>	0,0008 - 0,7631	25 / 100
<b>объектно-ориентированный подход</b>	0,0011 - 1,3493	20 / 120
<b>динамическая типизация</b>	-----	-----
<b>уровень системы (набора) команд</b>	0,0007 - 0,7451	15 / 95

Подводя итог, можно сделать следующие выводы сравнительного анализа полученных характеристик с теми, которые были получены для предыдущих программ :

Относительно прошлых реализаций программы, данная версия имеет следующие преимущества : 1) Меньший размер исходных файлов 2) Меньшее время выполнения программы для различных входных данных 3) Меньший размер исполняемого файла

Итог : Данный способ реализации предоставляет более удобное и простое создание объектов программы, за счет имитации бестипового решения с удалением АД и с функциональной декомпозицией по модулям.