Отчет.

Алиева Рената Эдуардовна ФКН ПИ ВШЭ БПИ202.

Вариант 42. Артефакт номер 14. Функция номер 3.

**Описание полученного задания:**

Данная программа описывает вычисление максимального расстояния, которое может пройти автомобиль(условие задачи 14) а также выполняет сортировку контейнера пузырьком (Bubble Sort) .В качестве ключей для сортировки и других действий используются результаты функции, вычисляющей максимальное расстояние(обработка данных в контейнере под номером 3). Программа выполнена в стиле ООП. В программе реализованы такие классы, как автомобильный транспорт(car), грузовик(truck) – содержит грузоподъемность; емкость топливного бака; расход топлива, автобус(bus) – содержит пасажировмещаемость; емкость топливного бака; расход топлива, легковой автомобиль(automobile) – содержит максимальную скорость;емкость топливного бака; расход топлива и контейнер(container).

**Работа программы:**

Программа ожидает одну из команд:

* -f infile outfile01 outfile02 ,где infile – имя файла в котором храняться входные данные, outfile01 – имя файла в котором будут выходные данные,outfle02 – имя файла в котором будут выходные дананные, после выполнения программой функцией(В данном случае Bubble Sort).
* -n number outfile01 outfile02 – похожая команда, в которой number – кол-во артефактов, которые необходимо сгенерировать с помощью функции rnd(). Остальные параметры такие же, как и в первой команде.

Ввод в программу в файле реализован следующим образом:

* Первый параметр – целое число от 1 до 3 обозначающее тип машины: 1 – грузовик,2 – автобус,3 - автомобиль.Второй параметр – индивидуальный параметр(описан выше для каждого типа машины),емкость топливного бака, расход топлива.
* **Число интерфейсных модулей:** 6
* **Число модулей реализации не включая main():** 5
* **Размер файлов кода(без Cmake):** 23000 Б
* **Размер скомпилированного кода:** 103000 Б

В директории input\_tests находятся тесты вида test01.txt

В директории output\_tests находятся тесты вида rnd\_test\_out.txt(выходной файл с сгенерированными рандомно параметрами) , rnd\_test\_sorted.txt(выходной файл с сортированными сгенерированными рандомно параметрами) ,а также test01\_out.txt(выходной файл после считывания из файлов папки input\_tests) и test01\_sorted.txt(сортированный вывод).

**Структурная схема:**

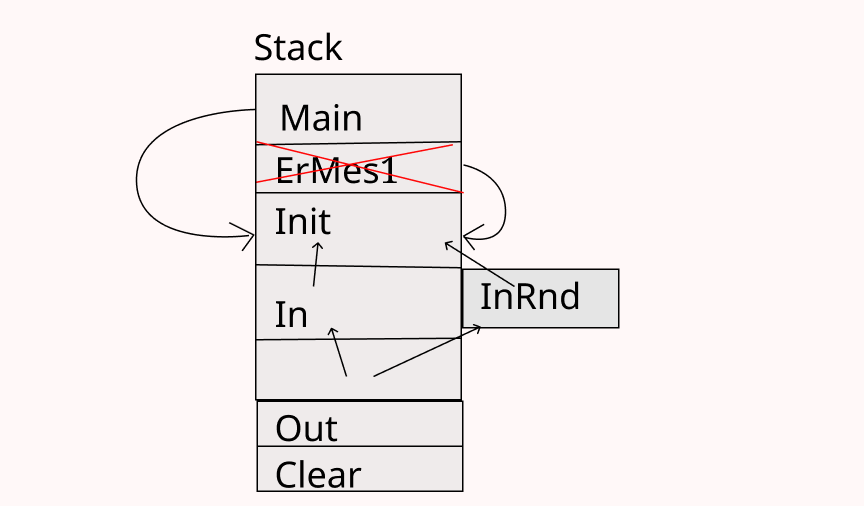
**Таблица типов.**

|  |  |  |
| --- | --- | --- |
| Тип | Имя | Размер |
| int | argc | 4 |
| double |  | 8 |
| **Class car:**  int  double  Random  Random  Random  Random  Random  Random | fuel\_capasity  fuel\_consumption  rnd3  rnd20  rnd50  rnd200  rnd5000  rnd30 | 12  4[0]  8[4]  0[12]  0[12]  0[12]  0[12]  0[12]  0[12] |
| **Class bus:**  short | passenger\_capacity | 2  2[0] |
| **Class automobile:**  short | max\_speed | 2  2[0] |
| **Class truck:**  short | lifting\_capacity | 2  2[0] |
| **Class container:**  int  car\*\*  int | len  storage  size | 20  4[0]  12[4]  4[16] |

**Глобальная память –** нет элементов.

**Память программы**

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Размер |
| Main(): arc | int | 4[0] |
| Main(): char\* | argv | 8[4] |
| Main(): container | c | 20\*10001[12] |
| Main():int | size | 4[200032] |
| car\* InRnd():  car\*  k | c  int | 12[0]  4[12] |
| Container Sort():  Size  Car\*  I  j | Int  temp  int  int | 4[0]  12[4]  4[16]  4[20] |
| Container In():  ifst | Ifstream& | По ссылке |

**Стек(Main())**

**Куча**

|  |
| --- |
| **Ifst** |
| **Ofst1** |
| **Ofst2** |

**Метрики.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест** | **Размер** | **Время выполнения(ООП)** | **Время выполнения(Процедурный стиль)** |
| Test01.txt | 8 | real 0m0,006s  user 0m0,001s  sys 0m0,000s | real 0m0,012s  user 0m0,000s  sys 0m0,001s |
| Test02.txt | 12 | real 0m0,010s  user 0m0,000s  sys 0m0,003s | real 0m0,016s  user 0m0,000s  sys 0m0,004s |
| Test03.txt | 1000 | real 0m0,012s  user 0m0,007s  sys 0m0,000s | real 0m0,030s  user 0m0,005s  sys 0m0,008s |
| Test04.txt | 5000 | real 0m0,027s  user 0m0,010s  sys 0m0,007s | real 0m0,031s  user 0m0,010s  sys 0m0,012s |
| Test05.txt | 9999 | real 0m0,024s  user 0m0,020s  sys 0m0,011s | real 0m0,031s  user 0m0,015s  sys 0m0,027s |

**Вывод:**

Программа, написанная на ООП позволяет написать код более эффективно и удобно, и лучше читаемо. Скорость работы программы, написанной в объектно-ориентированном стиле выше ,чем написанной в процедурном стиле. А также удобен в модульной разработке.