

Пояснительная записка к  
Домашнему заданию №1 по курсу  
Архитектура вычислительных систем

Шагаров Дмитрий Александрович  
БПИ202

Сентябрь-Октябрь  
2021

# Описание полученного задания

номер варианта	номер задания	номер функции
70	14	5

## Разработка сущностей

1. **Грузовик** (грузоподъемность кг – целое, емкость топливного бака в литрах (целое), расход топлива на 100 км в литрах (действительное))
2. **Автобус** (пассажировместимость – короткое целое, емкость топливного бака в литрах (целое), расход топлива на 100 км в литрах (действительное))
3. **Легковой автомобиль** (максимальная скорость – короткое целое, емкость топливного бака в литрах (целое), расход топлива на 100 км в литрах (действительное))

Каждая сущность имеет функцию нахождения максимального расстояния, которое может пройти автомобиль в км (действительное число).

Все введенные сущности размещаются в разработанном контейнере, после чего к контейнеру применяется сортировка Шелла (ключ - значение функции максимального расстояния). Элементы контейнера до и после сортировки выводится в форматизируемый поток.

## Формат ввода

Программа расположена по пути AoCS/bin/AoCS.

В программе предусмотрено два способа ввода данных при запуске из командной строки:

### 1. Использование генераторов случайных наборов данных -

команда

```
./AoCS -n number_of_vehicles out_path sorted_out_path
```

### 2. Ввод из заранее подготовленных тестовых файлов -

команда

```
./AoCS -f in_path out_path sorted_out_path
```

Второй способ предусматривает наличие файла in с описанием сущностей в формате: "одна строка = одно ТС.

Первый параметр - число от 1 до 3, где 1 - легковое авто, 2 - автобус, 3 - грузовик.

Второй параметр - уникальное для каждого ТС свойство 80 - 320 целое для легкового авто, 10 - 70 целое для автобуса, 1000 - 4000 целое для грузовика.

Третий параметр - объем топливного бака 50 - 150 целое для легкового авто, 80 - 250 целое для автобуса, 200 - 800 целое для грузовика.

Последний параметр - расход 6 - 25 вещественное для легкового авто, 12 - 30 вещественное для автобуса, 20 - 45 вещественное для грузовика.

Параметры указываются через один пробел, для вещественных разделитель - точка. Окончание файла - пустая строка.

Пример такого файла - `in_example.txt` находится в папке с проектом. Для корректной работы программы необходимо наличие файла, передаваемого как `in_path`.

# Структурная схема программного продукта с использованием процедурного подхода и статической типизацией

**Таблица типов**

short, int, double	2 байта, 4 байта, 8 байт
<u>struct Car</u> int tank_volume short max_speed double consumption	<u>14 байт</u> 4 байта[0] 2 байта[4] 8 байт[6]
<u>struct Bus</u> int tank_volume short max_passengers double consumption	<u>14 байт</u> 4 байта[0] 2 байта[4] 8 байт[6]
<u>struct Truck</u> int max_weight, tank_volume double consumption	<u>16 байт</u> 8 байт[0, 4] 8 байт[8]
<u>struct Vehicle</u> enum key key k Car car Bus bus Truck truck	<u>52 байта</u> 4 байта[0] 4 байта[4] 14 байт[8] 14 байт[22] 16 байт[36]
<u>struct Container</u> enum max_len int len Vehicle cont	<u>520060 байт</u> 4 байта[0] 4 байта[4] 520052 байта[8] (= 52 * 10001)

## Память программы

main(int argc, char *argv[]) int argc char *argv Container c int size	4 байта[0] 8 байт[4] 520052 байта[8] 4 байта[520064]
void StartMessage	
void ErrorMessage1	
void ErrorMessage2	
void Init(Container &cont)	
void In(Container &cont, FILE *input)	
void InRnd(Container &cont, int size)	
void ShellSortByMaxDistance(Container &c) int d, i, j Vehicle *temp	12 байт[0, 4, 8] 52 байта[12]
double MaxDistance(Vehicle &vehicle)	
void Out(Container &cont, FILE *output) int i	4 байта[0]
void Clear(Container &cont) int i	4 байта[0]

## Стек вызовов

Возможны следующие варианты ( | означает что вызывается одна из соответствующих функций, ? ?  $\rightarrow$  - что будет, если программа вызовет данную функцию)

main
?ErrorMessage1? $\rightarrow$ main
StartMessage
StartMessage
Init
Init
In (error $\rightarrow$ main)   InRnd   ?ErrorMessage2? $\rightarrow$ main
In   InRnd
Out
Out
ShellSortByMaxDistance
ShellSortByMaxDistance
Out
Out
Clear
Clear
main

# Основные характеристики программы

интерфейсных модулей	модулей реализации
6	6 ( + main.cpp для тестирования)

общий размер исходных текстов	общий размер результатов тестов
17,37 Кб	8,27 Мб

## Результаты тестов

Тесты расположены в папке tests и были сгенерированы с использованием написанного случайного генератора тестов. Результаты расположены в папке test\_results.

Время работы программы в тестах (в секундах)		
	тесты из файлов	случайная генерация в программе
50 элементов	0,001125	0,000540
500 элементов	0,004507	0,004260
2500 элементов	0,048916	0,049225
6000 элементов	0,226239	0,318252
10000 элементов	0,662193	0,649833

Как можно видеть, особо существенной разницы в зависимости от типа ввода нет. Разброс скорее обусловлен степенью начальной сортировки контейнера.

## Заключение

На данный момент процедурный подход со статической типизацией - первое задание на курсе, поэтому сравнивать его не с кем. Однако, в прошлом году был изучен язык С#, и во время написания кода программы очень хотелось писать в его стиле с применением ООП и, в частности, наследования.