Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Информационная безопасность»

ОТЧЕТ

по лабораторной работе № 4

по учебной дисциплине «Алгоритмические языки»

на тему: «Использование объектов своих классов в последовательных контейнерах библиотеки STL»

Вариант 18

Выполнил:

Студент 1 курса, гр. ИУ8-24

Ожогин Михаил

2024 г.

Цель работы:

Практическое применение контейнеров стандартной библиотеки шаблонов (STL) для хранения объектов пользовательского класса, а также использование методов и алгоритмов STL для работы с этими контейнерами.

Задачи работы:

- Изучить необходимые учебные материалы, посвященные последовательным контейнерам библиотеки шаблонов в языке Си++
- Разработать программу на языке Си++ для решения заданного варианта задания
- Отладить программу
- Представить результаты работы программы
- Подготовить отчет по лабораторной работе

Условие задачи:

1. **Описание класса Vehicle:**
    - Приватные поля класса:
        1. Название модели автомобиля
        2. Массив потребления топлива, состоящий из 3 элементов: расход загородом, в городе, смешанный
        3. Максимальная скорость автомобиля
        4. Мощность автомобиля
        5. Оператор вывода в поток
        6. Оператор получения из потока
        7. Оператор сравнения меньше
    - Публичные поля класса:
        1. Пустой конструктор
        2. Конструктор с заполнением всех полей
        3. Конструктор копирования

2. **Организовать контейнер list объекта своего класса.**
3. **Реализовать сортировку контейнера list с объектами моего класса**
4. **Реализовать перенос из контейнера list в контейнер vector, используя метод std::copy**

Выполнение работы:

Заголовочный файл класса Vehicle – **vehicle.hpp**:

```
#ifndef VEHICLE_HPP_
#define VEHICLE_HPP_

#include <string>
#include <list>
#include <fstream>
#include <iostream>
#include <algorithm>

class Vehicle{
  public:
  Vehicle();
  Vehicle(const std::string&, const float*, const size_t&, const float&);
  Vehicle(const Vehicle&);

  private:
  std::string _model;
  float _fuel_consumption[3];
  size_t _max_speed;
  float _engine_power;

  friend std::ostream& operator<<(std::ostream&, const Vehicle&);
  friend std::istream& operator>>(std::istream&, Vehicle&);
  friend bool operator<(const Vehicle&, const Vehicle&);
};
```

```cpp
Vehicle::Vehicle() : _model(""), _fuel_consumption{0,0,0}, _max_speed(0),
_engine_power(0) {};

Vehicle::Vehicle(const std::string& m, const float* f, const size_t& ma,
const float& e) : _model(m), _fuel_consumption{f[0],f[1],f[2]},
_max_speed(ma), _engine_power(e) {};

Vehicle::Vehicle(const Vehicle& rhs) : _model(rhs._model),
_fuel_consumption{rhs._fuel_consumption[0], rhs._fuel_consumption[1],
rhs._fuel_consumption[2]}, _max_speed(rhs._max_speed),
_engine_power(rhs._engine_power) {};


std::ostream& operator<<(std::ostream& os, const Vehicle& rhs) {

  os << "Vehicle Parameters:\n";

  os << "Model: " << rhs._model << std::endl;

  os << "Fuel consumption at the highway: " << rhs._fuel_consumption[0] <<
", in the city: " << rhs._fuel_consumption[1] << ", mixed: " <<
rhs._fuel_consumption[2] << std::endl;

  os << "Max speed: " << rhs._max_speed << std::endl;

  os << "Engine power: " << rhs._engine_power << std::endl;

  return os;

}


std::istream& operator>>(std::istream& is, Vehicle& rhs) {

  is >> std::ws;

  std::getline(is, rhs._model);

  for (int i = 0; i < 3; ++i) {

    is >> rhs._fuel_consumption[i];

  }

  is >> rhs._max_speed;

  is >> rhs._engine_power;

  return is;

}


template<class T1>

void CustomOutput(T1& obj) {

  std::ofstream output("output.txt", std::ios::app);

  output << obj << std::endl;;

  std::cout << obj << std::endl;

  output.close();
```

```
}

bool operator<(const Vehicle& lhs, const Vehicle& rhs) {
  std::string lowl = lhs._model;
  std::string lowr = rhs._model;
  std::transform(lowl.begin(), lowl.end(), lowl.begin(), [](unsigned char
c){return tolower(c);});
  std::transform(lowr.begin(), lowr.end(), lowr.begin(), [](unsigned char
c){return tolower(c);});
  return lowl < lowr;
}

#endif // VEHICLE_HPP_
```

Файл реализации – **main.cpp**:

```
#include "../include/vehicle.hpp"
#include <vector>

int main() {
  std::ofstream clear("output.txt");
  clear.close();
  std::ifstream input("input.txt");

  std::list<Vehicle> task_list;

  for (size_t i = 0; i < 5; ++i) {
    Vehicle intolist;
    input >> intolist;
    task_list.push_back(intolist);
  }

  CustomOutput("List before Sorting: ");
  for (const auto& el : task_list) {
    CustomOutput(el);
  }
```

```
  CustomOutput("List after Sorting: ");
  task_list.sort();
  for (auto el : task_list) {
    CustomOutput(el);
  }

  CustomOutput("Vector copied from List: ");
  std::vector<Vehicle> result;
  std::copy(task_list.begin(), task_list.end(),
std::back_inserter(result));
  for (const auto& el : result) {
    CustomOutput(el);
  }
}
```

Файл со входными данными – **input.txt:**

Lada Vesta Sedan

6.3 10.1 7.7

181

122

Lada Niva Legend

8.3 12.1 9.9

142

83

Lada Niva Travel

8.5 13.4 10.2

140

80

Lada Largus Cross

6.3 10.2 7.8

170

106

Lada Vesta SW Sportline

5.8 9.9 7.4

186

118

Lada Granta Cross

5.2 8.7 6.5

178

106

Файл с выходными данными – output.txt:

List before Sorting:

Vehicle Parameters:

Model: Lada Vesta Sedan

Fuel consumption at the highway: 6.3, in the city: 10.1, mixed: 7.7

Max speed: 181

Engine power: 122

Vehicle Parameters:

Model: Lada Niva Legend

Fuel consumption at the highway: 8.3, in the city: 12.1, mixed: 9.9

Max speed: 142

Engine power: 83

Vehicle Parameters:

Model: Lada Niva Travel

Fuel consumption at the highway: 8.5, in the city: 13.4, mixed: 10.2

Max speed: 140

Engine power: 80

Vehicle Parameters:

Model: Lada Largus Cross

Fuel consumption at the highway: 6.3, in the city: 10.2, mixed: 7.8

Max speed: 170

Engine power: 106

Vehicle Parameters:

Model: Lada Vesta SW Sportline

Fuel consumption at the highway: 5.8, in the city: 9.9, mixed: 7.4

Max speed: 186

Engine power: 118

List after Sorting:

Vehicle Parameters:

Model: Lada Largus Cross

Fuel consumption at the highway: 6.3, in the city: 10.2, mixed: 7.8

Max speed: 170

Engine power: 106

Vehicle Parameters:

Model: Lada Niva Legend

Fuel consumption at the highway: 8.3, in the city: 12.1, mixed: 9.9

Max speed: 142

Engine power: 83

Vehicle Parameters:

Model: Lada Niva Travel

Fuel consumption at the highway: 8.5, in the city: 13.4, mixed: 10.2

Max speed: 140

Engine power: 80

Vehicle Parameters:

Model: Lada Vesta Sedan

Fuel consumption at the highway: 6.3, in the city: 10.1, mixed: 7.7

Max speed: 181

Engine power: 122

Vehicle Parameters:

Model: Lada Vesta SW Sportline

Fuel consumption at the highway: 5.8, in the city: 9.9, mixed: 7.4

Max speed: 186

Engine power: 118

Vector copied from List:

Vehicle Parameters:

Model: Lada Largus Cross

Fuel consumption at the highway: 6.3, in the city: 10.2, mixed: 7.8

Max speed: 170

Engine power: 106

Vehicle Parameters:

Model: Lada Niva Legend

Fuel consumption at the highway: 8.3, in the city: 12.1, mixed: 9.9

Max speed: 142

Engine power: 83

Vehicle Parameters:

Model: Lada Niva Travel

Fuel consumption at the highway: 8.5, in the city: 13.4, mixed: 10.2

Max speed: 140

Engine power: 80

Vehicle Parameters:

Model: Lada Vesta Sedan

Fuel consumption at the highway: 6.3, in the city: 10.1, mixed: 7.7

Max speed: 181

Engine power: 122

Vehicle Parameters:

Model: Lada Vesta SW Sportline

Fuel consumption at the highway: 5.8, in the city: 9.9, mixed: 7.4

Max speed: 186

Engine power: 118

Вывод:

В ходе выполнения лабораторной работы было успешно реализовано использование объектов собственных классов в последовательных контейнерах библиотеки STL с применением различных методов и алгоритмов для работы с ними. Это позволило познакомиться с применением стандартной библиотеки шаблонов в практических задачах программирования.