**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Языки программирования

|  |
| --- |
| Контейнеры и алгоритмы STL |

Руководитель А.В. Проскурин

подпись, дата инициалы, фамилия

Обучающийся БПИ21-02 В. М. Самохвалов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2022 г.

# ЦЕЛЬ РАБОТЫ

Получение практических навыков разработки и отладки программ по обработке строк c использованием стандартной библиотеки шаблонов (STL).

# порядок выполнения работы

1. Ознакомиться с общей постановкой задачи.
2. Ознакомится с вариантом задания – соответствует вашему номеру в списке группы (при нехватке заданий вариант задания вычисляется как номер\_в\_списке\_группы - количество\_заданий).
3. Разработать классы согласно варианту задания.
4. Написать и отладить программу на подготовленных наборах тестовых данных.
5. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:

•  титульный лист;

•  цель лабораторной работы;

•  постановку задачи;

•  схему наследования классов (UML диаграмма классов);

•  текст программы с комментариями;

•  демонстрацию работы программы (Снимки экрана при выполнении действий программы с описанием).

•  краткие ответы на контрольные вопросы;

•  выводы по лабораторной работе.

1. Защитить лабораторную работу перед преподавателем.

# постановка задачи

Вариант №21.  
Формулировка задания: На основе каталога автосалона требуется определить пять самых быстрых автомобилей с автоматической коробкой передач и пять с механической. Вывод данных на экран осуществлять с группировкой по коробке передач и отображать в алфавитном порядке по формату:

Марка автомобиля Модель (Скорость)

Формат входных данных: Марка автомобиля, Модель, Скорость автомобиля и тип коробки передач.

# ХОД РАБОТЫ

CMakeLists.txt

cmake\_minimum\_required(*VERSION* 3.24)  
project(stl)  
  
set(CMAKE\_CXX\_STANDARD 23)  
  
add\_executable(stl main.cpp Car.cpp Car.h)

main.cpp

*//Подключаем зависимости*#include <iostream>  
#include <vector>  
#include <fstream>  
#include "Car.h"  
  
*//<editor-fold desc="Comparators">  
//Компараторы для сортировки  
//Компаратор для сортировки по скорости  
bool* cmpMaxSpeed(*const* Car& a, *const* Car& b) {  
 *return* a.getMaxSpeed() > b.getMaxSpeed();  
}  
  
*//Компаратор для сортировки по производителю  
bool* cmpManufacturer(Car a, Car b) {  
 *return* a.getManufacturer() < b.getManufacturer();  
}  
*//</editor-fold>  
//<editor-fold desc="Functions">  
//Функция сортировки  
void* sorting(std::vector<Car> &cars) {  
 std::vector<Car> A, M;  
 *//Фильтруем по типу коробки передач  
 for* (*const auto*& car: cars)  
 car.getTransmission() == 'A' ? A.push\_back(car) : M.push\_back(car);  
 *//Сортируем по скорости* std::sort(A.begin(), A.end(), cmpMaxSpeed);  
 std::sort(M.begin(), M.end(), cmpMaxSpeed);  
 *//Оставляем первые 5 элементов* A = std::vector<Car>(A.begin(), A.begin() + 5);  
 M = std::vector<Car>(M.begin(), M.begin() + 5);  
 *//Сортируем по производителю* std::sort(A.begin(), A.end(), cmpManufacturer);  
 std::sort(M.begin(), M.end(), cmpManufacturer);  
 *//Объединяем в один вектор* cars.clear();  
 cars.insert(cars.end(), A.begin(), A.end());  
 cars.insert(cars.end(), M.begin(), M.end());  
}  
*//Функция парсинга файла  
void* parse(*const* std::string &path, std::vector<Car> &cars) {  
 std::ifstream fin(path);  
 *//Проверяем, открылся ли файл  
 if* (!fin.is\_open()) {  
 std::cerr << path << " is not found!" << std::endl;  
 *return*;  
 }  
 *//Парсим файл* std::string manufacturer;  
 std::string model;  
 *double* maxSpeed;  
 *char* transmission;  
 *while* (fin >> manufacturer >> model >> maxSpeed >> transmission) {  
 cars.emplace\_back(manufacturer, model, maxSpeed, transmission);  
 }  
 *//Закрываем файл* fin.close();  
}  
*//</editor-fold>  
int* main() {  
 *//Создаем вектор автомобилей* std::vector<Car> cars;  
 *//Парсим файл* parse("/Users/vadimsam/CLionProjects/stl/data.txt", cars);  
 *//Выводим данные о всех автомобилях* std::cout << "====================" << std::endl << "All cars:" << std::endl << "====================" << std::endl;  
 *for* (*const auto*& car: cars)  
 std::cout << car << std::endl;  
 *//Сортируем* sorting(cars);  
 *//Выводим данные о топ 5 автомобилях по скорости в каждой категории* std::cout << "====================" << std::endl;  
 std::cout << "Top 5 cars:" << std::endl;  
 std::cout << "====================" << std::endl;  
 std::cout << "Automatic:" << std::endl;  
 *for* (*int* i = 0; i < 5; ++i)  
 std::cout << "\t" << i + 1 << ". " << cars[i] << std::endl;  
 std::cout << "Manual:" << std::endl;  
 *for* (*int* i = 5; i < 10; ++i)  
 std::cout << "\t" << i - 4 << ". " << cars[i] << std::endl;  
 *return* 0;  
}

Car.h

#ifndef **STL\_CAR\_H**#define **STL\_CAR\_H***//Подключаем библиотеку для работы со строками*#include <string>  
  
*//Класс Car  
class* Car {  
*private*:  
 *//<editor-fold desc="Fields">  
 //Поля класса* std::string manufacturer; *//Производитель* std::string model; *//Модель  
 char* transmission; *//Тип коробки передач  
 double* maxSpeed; *//Максимальная скорость  
 //</editor-fold>  
public*:  
 *//Конструктор класса* Car(std::string manufacturer, std::string model, *double* maxSpeed, *char* transmission);  
*//<editor-fold desc="Getters and Setters">  
//Геттеры и сеттеры* std::string getManufacturer();  
  
 std::string getModel();  
  
 *double* getMaxSpeed() *const*;  
  
 *char* getTransmission() *const*;  
  
 *void* setManufacturer(std::string manufacturer);  
  
 *void* setModel(std::string model);  
  
 *void* setMaxSpeed(*double* maxSpeed);  
  
 *void* setTransmission(*char* transmission);  
  
*//</editor-fold>  
//<editor-fold desc="I/O">  
 friend* std::ostream &*operator*<<(std::ostream &os, *const* Car &car);  
  
 *friend* std::istream &*operator*>>(std::istream &is, Car &car);  
*//</editor-fold>*};  
  
  
#endif *//STL\_CAR\_H*

Car.cpp

#include "Car.h"  
#include <iostream>  
  
Car::Car(std::string manufacturer, std::string model, *double* maxSpeed, *char* transmission) {  
 *this*->manufacturer = manufacturer;  
 *this*->model = model;  
 *this*->maxSpeed = maxSpeed;  
 *this*->transmission = transmission;  
}  
  
*//<editor-fold desc="Getters and Setters">*std::string Car::getManufacturer() {  
 *return* manufacturer;  
}  
  
std::string Car::getModel() {  
 *return* model;  
}  
  
*double* Car::getMaxSpeed() *const* {  
 *return* maxSpeed;  
}  
  
*char* Car::getTransmission() *const* {  
 *return* transmission;  
}  
  
*void* Car::setManufacturer(std::string \_manufacturer) {  
 *this*->manufacturer = std::move(\_manufacturer);  
}  
  
*void* Car::setModel(std::string \_model) {  
 *this*->model = std::move(\_model);  
}  
  
*void* Car::setMaxSpeed(*double* \_maxSpeed) {  
 *this*->maxSpeed = \_maxSpeed;  
}  
  
*void* Car::setTransmission(*char* \_transmission) {  
 *this*->transmission = \_transmission;  
}  
*//</editor-fold>  
//<editor-fold desc="I/O">*std::ostream &*operator*<<(std::ostream &os, *const* Car &car) {  
 os << car.manufacturer << " " << car.model << " " << car.maxSpeed << " " << car.transmission;  
 *return* os;  
}  
  
std::istream &*operator*>>(std::istream &is, Car &car) {  
 is >> car.manufacturer >> car.model >> car.maxSpeed >> car.transmission;  
 *return* is;  
}  
*//</editor-fold>*

Data.txt

DeLorean DMC-12 177 A  
Skoda Kodiaq 216 M  
Rolls-Royce Phantom 250 A  
Bugatti Chiron 261 A  
Ford GT 216 M  
Tesla Model-S 250 A  
Skoda Octavia 216 M  
Lamborghini Aventador 217 M  
Aston-Martin DB11 200 M  
Honda S2000 240 A

 Вывод

/Users/vadimsam/CLionProjects/stl/cmake-build-debug/stl

====================

All cars:

====================

DeLorean DMC-12 177 A

Skoda Kodiaq 216 M

Rolls-Royce Phantom 250 A

Bugatti Chiron 261 A

Ford GT 216 M

Tesla Model-S 250 A

Skoda Octavia 216 M

Lamborghini Aventador 217 M

Aston-Martin DB11 200 M

Honda S2000 240 A

====================

Top 5 cars:

====================

Automatic:

1. Bugatti Chiron 261 A

2. DeLorean DMC-12 177 A

3. Honda S2000 240 A

4. Rolls-Royce Phantom 250 A

5. Tesla Model-S 250 A

Manual:

1. Aston-Martin DB11 200 M

2. Ford GT 216 M

3. Lamborghini Aventador 217 M

4. Skoda Kodiaq 216 M

5. Skoda Octavia 216 M

Process finished with exit code 0

# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Охарактеризуйте класс vector, как происходит обращение к элементам?

C-подобный динамический массив произвольного доступа с автоматическим изменением размера при добавлении/удалении элемента. Элементы хранятся в непрерывной области памяти.

1. Как происходит добавление/удаление элементов в классе vector?
   1. v.push\_back(elem) - вставка в конец
   2. v.insert(it±offset, elem) – вставка единственного элемента
   3. v.insert(it±offset, n, elem) – вставка n одинаковых элементов
   4. v.insert(vit±voffset,vecit±vecoffsetbegin,vecit±vecoffsetend) – вставка одого контейнера в другой
2. v.pop\_back() – удаление последнего элемента с сохранением емкости
3. v.erase(it±voffset) – удаление одного элемента в заданной позиции
4. v.erase(it±voffsetbegin,it±voffsetend) – удаление нескольких элементов
5. v.clear() - удаление всего содержимого

# ВЫВОДЫ

Получены практические навыки разработки и отладки программ по обработке строк c использованием стандартной библиотеки шаблонов (STL).