|  |  |
| --- | --- |
|  | **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, искусственный интеллект и системы управления” (ИУ) |

|  |  |  |
| --- | --- | --- |
| КАФЕДРА | “Информационная безопасность” (ИУ8) | |
|  | |

Лабораторная работа № 3

ПО КУРСУ

« Алгоритмические языки »

на тему «Изучение возможностей наследования классов»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-24 |  |  |  | А.А. Иванова |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | Д.В. Барыкин |
|  |  |  |  |  | (И.О. Фамилия) |

**Цель работы:**

Овладение навыками разработки программ на языке Си++, использующих возможности наследования классов для решения различных задач.

***Вариант 6:***

Создать базовый класс «автомобиль». Элементы класса: поле, содержащее наименование модели автомобиля; поле, содержащее значение максимальной скорости (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров автомобиля. Создать производный класс «грузовой автомобиль». Элементы класса: дополнительно поле, содержащее грузоподъемность автомобиля в тоннах; конструктор для инициализации полей; переопределенная функция печати параметров автомобиля (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

*Текст программы:*

#include <iostream>

#include <cstring>

using namespace std;

class Car {

protected:

string model;

int maxSpeed;

public:

Car(const string& model, int maxSpeed) : model(model), maxSpeed(maxSpeed) {}

virtual void printInfo() {

cout << "Model: " << model <<endl;

cout << "Max Speed: " << maxSpeed << " km/h" <<endl;

}

};

class Truck : public Car {

private:

double cargoCapacity;

public:

Truck(const string& model, int maxSpeed, double cargoCapacity) : Car(model, maxSpeed), cargoCapacity(cargoCapacity) {}

void printInfo() override {

Car::printInfo();

cout << "Cargo Capacity: " << cargoCapacity << " tons" << endl;

}

};

int main() {

Car car("Toyota", 200);

Truck truck("Volvo", 150, 5.0);

// Статический полиморфизм

car.printInfo();

cout <<endl;

truck.printInfo();

cout << "\n\n";

// Динамический полиморфизм

Car\* carPtr = &car;

carPtr->printInfo();

cout << endl;

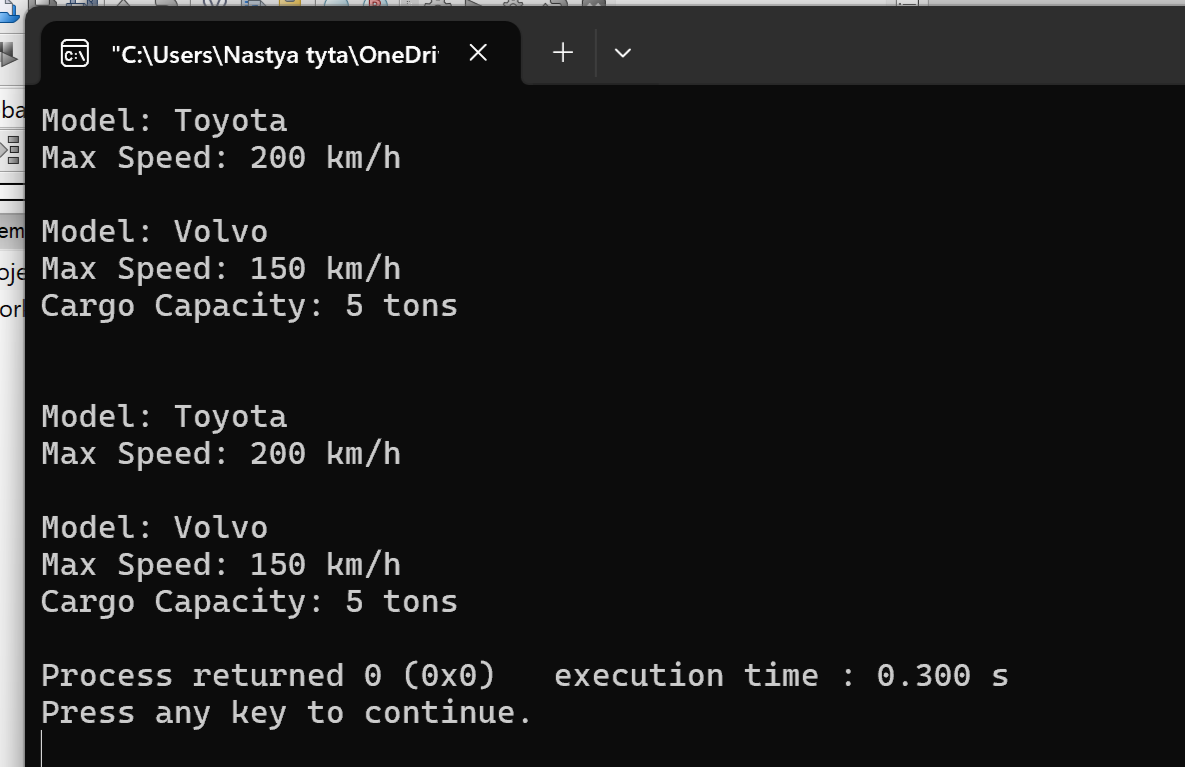
Car\* truckPtr = &truck;

truckPtr->printInfo();

return 0;

}

**Результат выполнения программы:**

****

**Вывод:**

Выполнив лабораторную работу №3, я узнала тему «Изучение возможностей наследования классов», необходимые учебные материалы, посвященные наследованию классов в языке Си++, разработала и отладила программу на языке Си++ для решения заданного варианта задания. Проанализировав результаты выполнения программы, подготовлен отчёт по лабораторной работе.