Nicoleta Radu

Gr.2,Sgr.4, Proiect c++

# Header “Vehicle.h”

#pragma once

#include <iostream>

#include <vector>

#include <string>

class Vehicle

{

public:

// Variables

int seats;

float horsePower;

std::string registrationNumber;

std::string nature;

std::string fuel;

std::string brand;

std::string brandClass;

private:

// Private Variables

void initVariables();

public:

// Constructor & Destructor

Vehicle();

~Vehicle();

};

class Mercedes: Vehicle

{

private:

// Variables

std::vector<Vehicle> mercedes;

int parkSize;

// Private Functions

void initVariables(int);

public:

// Variables

// Functions

void addMercedes();

void deleteMercedes(int);

void replaceMercedes(int);

void setNumberOfSeats(int,int);

void setHorsePower(int, float);

void setRegistrationNumber(int, std::string);

void setNature(int, std::string);

void setBrand(int, std::string);

void setBrandClass(int, std::string);

void printCar(int);

// Constructors & Destructors

Mercedes(int);

~Mercedes();

};

# CPP “Vehicle.cpp”

#include "Vehicle.h"

void Vehicle::initVariables()

{

seats = 4;

horsePower = 120.0;

registrationNumber = "0000 CITY 0";

nature = "Berlin";

brandClass = "S";

brand = "Mercedes";

}

void Mercedes::initVariables(int numberOfCars)

{

mercedes.resize(numberOfCars);

}

// Functions

void Mercedes::addMercedes()

{

Vehicle carToPush;

mercedes.push\_back(carToPush);

}

void Mercedes::deleteMercedes(int index)

{

mercedes.erase(mercedes.begin() + index);

}

void Mercedes::replaceMercedes(int index)

{

Vehicle carToPush;

mercedes.insert(mercedes.begin() + index, carToPush);

}

void Mercedes::setNumberOfSeats(int mercedesIndex, int mercedesSeats)

{

mercedes.at(mercedesIndex).seats = mercedesSeats;

}

void Mercedes::setHorsePower(int mercedesIndex, float mercedesHorsePower)

{

mercedes.at(mercedesIndex).horsePower = mercedesHorsePower;

}

void Mercedes::setRegistrationNumber(int mercedesIndex, std::string mercedesRegistrationNum)

{

mercedes.at(mercedesIndex).registrationNumber = mercedesRegistrationNum;

}

void Mercedes::setNature(int mercedesIndex, std::string mercedesNature)

{

mercedes.at(mercedesIndex).nature = mercedesNature;

}

void Mercedes::setBrand(int mercedesIndex, std::string mercedesBrand)

{

mercedes.at(mercedesIndex).brand = mercedesBrand;

}

void Mercedes::setBrandClass(int mercedesIndex, std::string mercedesClass)

{

mercedes.at(mercedesIndex).brandClass = mercedesClass;

}

void Mercedes::printCar(int mercedesIndex)

{

std::cout << "Car brand: " << mercedes.at(mercedesIndex).brand << std::endl;

std::cout << "Car brand class: " << mercedes.at(mercedesIndex).brandClass << std::endl;

std::cout << "Car horse power: " << mercedes.at(mercedesIndex).horsePower << std::endl;

std::cout << "Car Number of seats: " << mercedes.at(mercedesIndex).seats << std::endl;

std::cout << "Car nature: " << mercedes.at(mercedesIndex).nature << std::endl;

std::cout << "Car registration number: " << mercedes.at(mercedesIndex).registrationNumber << std::endl;

}

// Constructors & Destructors

Vehicle::Vehicle()

{

initVariables();

}

Vehicle::~Vehicle()

{

}

Mercedes::Mercedes(int numberOfCars): Vehicle()

{

initVariables(numberOfCars);

}

Mercedes::~Mercedes()

{

}

# Main “Parc Automobile”

// Parc Automobile.cpp : This file contains the 'main' function. Program execution begins and ends there.

//

#include "Vehicle.h"

#include <iostream>

#include <vector>

using namespace std;

int main()

{

Mercedes myMercedesPark(10);

myMercedesPark.addMercedes();

myMercedesPark.setBrand(0, "Lighting");

myMercedesPark.setBrandClass(0, "S");

myMercedesPark.setHorsePower(0, 2000);

myMercedesPark.setNature(0, "Berlin");

myMercedesPark.setNumberOfSeats(0, 5);

myMercedesPark.setRegistrationNumber(0, "44432 NY 7");

myMercedesPark.printCar(0);

}

