

Министерство науки и высшего образования РФ
Пензенский государственный университет
Кафедра “Вычислительная техника”

Отчёт

по лабораторной работе №3
по курсу “Объектно-ориентированное программирование”
на тему “Изучение механизмов наследования в языке C++”
Вариант 1

Выполнил студент гр. 22ВВВ3:
Кулахметов С.И.

Приняли:
к.т.н., доцент Евсеева Ю.И.
к.т.н., доцент Гудков А.А.

Пенза 2024

Цель работы

Изучить механизмы наследования в языке C++.

Лабораторное задание

Создать абстрактный класс **Vehicle** (транспортное средство). На его основе реализовать классы **Plane** (самолет), **Car** (автомобиль) и **Ship** (корабль). Классы должны иметь возможность задавать и получать координаты и параметры средств передвижения (цена, скорость, год выпуска) с помощью свойств. Для самолета должна быть определена высота, для автомобиля и корабля – количество пассажиров, для корабля – порт приписки. Динамические характеристики задать с помощью методов.

Пояснительный текст к программе

Программа разделена на 9 файлов, каждый из которых включает в себя различные модули лабораторной работы: 1 файл `main.cpp` — содержит главную часть программы и функцию `main()`; 2 файл `Vehicle.h` — содержит описание базового класса **Vehicle**, от которого наследуются все остальные классы в лабораторной работе; 3 файл `Vehicle.cpp` — содержит реализацию методов класса **Vehicle**; 4 файл `Car.h` - содержит описание дочернего класса **Car**; 5 файл `Car.cpp` — содержит реализацию собственных методов класса **Car**; 6 файл `Plane.h` — содержит описание дочернего класса **Plane**; 7 файл `Plane.cpp` — содержит реализацию собственных методов класса **Plane**; 8 файл `Ship.h` — содержит описание дочернего класса **Ship**; 9 файл `Ship.cpp` — содержит реализацию собственных методов класса **Ship**.

Результаты работы программы

В начале работы программы необходимо заполнить все поля объектов (рис. 1).

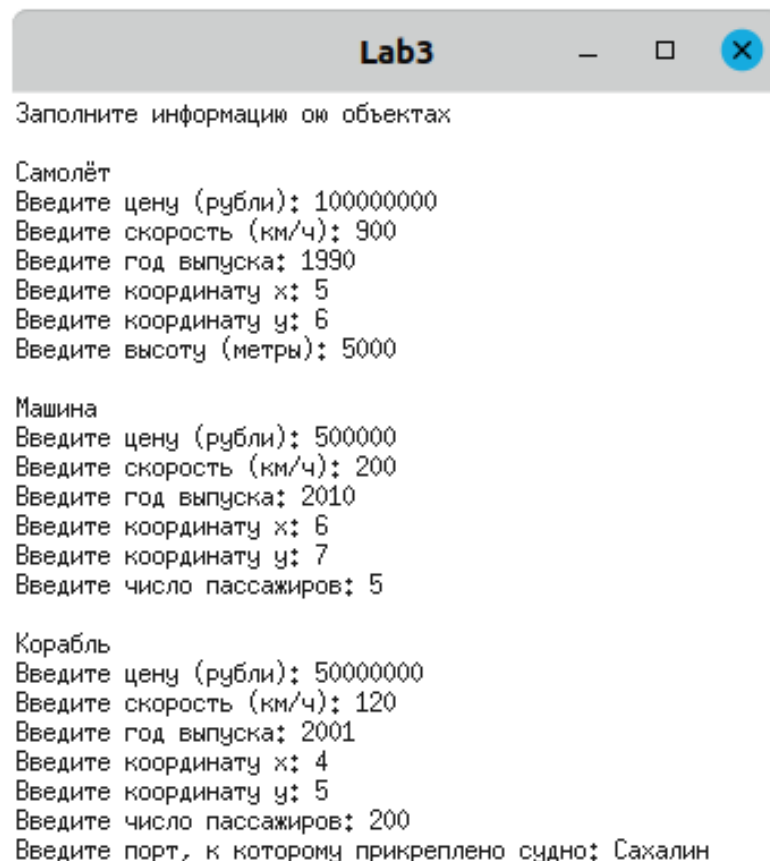


Рисунок 1 — Заполнение полей объектов

После чего выводится вся информация из полей объектов (рис. 2).

```
Самолёт
100000000 900 1990 5 6 5000
Машина
500000 200 2010 6 7 5
Корабль
50000000 120 2001 4 5 200 Сахалин
Process returned 0 (0x0)   execution time : 90.617 s
Press ENTER to continue.
```

Рисунок 2 — Вывод информации из полей объектов

Наследование в языке C++ можно рассмотреть на примере классов **Vehicle** (рис. 3) и **Car** (рис. 4). После того, как мы указали, что класс **Car** наследуется от **Vehicle** при помощи знака « : », можно задать одно поле нового класса, отвечающее за число пассажиров. Как только мы начинаем взаимодействовать с объектом класса **Car**, мы видим, что в нём доступны те же методы и поля, которые изначально были определены только в классе **Vehicle** (рис. 5, 6).

```

class Vehicle
{
    protected:
        int price = 0;
        int speed = 0;
        int year_of_issue = 0;
        int x_coord = 0;
        int y_coord = 0;

    public:
        void SetPrise(int price);
        void SetSpeed(int speed);
        void SetYear(int year);
        void SetX(int x_coord);
        void SetY(int y_coord);

        int GetPrice() {return price;}
        int GetSpeed() {return speed;}
        int GetYear() {return year_of_issue;}
        int GetX() {return x_coord;}
        int GetY() {return y_coord;}
};

```

Рисунок 3 — Код базового класса Vehicle

```

class Car : public Vehicle //Объявление и наследование от Vehicle
{
    private:
        int number = 0;

    public:
        void SetNumber(int number);

        int GetNumbet() {return number;}
};

```

Рисунок 4 — Код дочернего класса Car

```

car.SetPrise(price);
car.SetSpeed(speed);
car.SetYear(year);
car.SetX(x);
car.SetY(y);
car.SetNumber(number);

```

Рисунок 5 — Доступ к методам, унаследованным от Vehicle

```

cout << endl << "Машина" << endl;
cout << car.GetPrice() << " " << car.GetSpeed() << " " << car.GetYear() \
<< " " << car.GetX() << " " << car.GetY() << " " << car.GetNumbet();

```

Рисунок 6 — Доступ к полям, унаследованным от Vehicle

Вывод

В ходе выполнения данной лабораторной работы был изучен механизм наследования в языке программирования C++.

Ссылка на *GitHub* репозиторий с лабораторной работой

<https://github.com/KulakhmetovS/OOP>

Приложение А

Листинг программы

Файл main.cpp

```
#include <iostream>
#include "Car.h"
#include "Plane.h"
#include "Ship.h"

using namespace std;

int main()
{
    int price, speed, year, x, y, number, high;
    string port;

    Car car;
    Plane plane;
    Ship ship;

    cout << "Заполните информацию ою объектах" << endl;

    cout << endl << "Самолёт" << endl;
    cout << "Введите цену (рубли): ";
    cin >> price;
    cout << "Введите скорость (км/ч): ";
    cin >> speed;
    cout << "Введите год выпуска: ";
    cin >> year;
    cout << "Введите координату x: ";
    cin >> x;
    cout << "Введите координату y: ";
    cin >> y;
    cout << "Введите высоту (метры): ";
    cin >> high;

    plane.SetPrise(price);
    plane.SetSpeed(speed);
    plane.SetYear(year);
    plane.SetX(x);
    plane.SetY(y);
    plane.SetHigh(high);

    cout << endl << "Машина" << endl;
    cout << "Введите цену (рубли): ";
    cin >> price;
    cout << "Введите скорость (км/ч): ";
    cin >> speed;
    cout << "Введите год выпуска: ";
    cin >> year;
    cout << "Введите координату x: ";
    cin >> x;
    cout << "Введите координату y: ";
    cin >> y;
    cout << "Введите число пассажиров: ";
    cin >> number;
```

```

car.SetPrise(price);
car.SetSpeed(speed);
car.SetYear(year);
car.SetX(x);
car.SetY(y);
car.SetNumber(number);

```

```

cout << endl << "Корабль" << endl;
cout << "Введите цену (рубли): ";
cin >> price;
cout << "Введите скорость (км/ч): ";
cin >> speed;
cout << "Введите год выпуска: ";
cin >> year;
cout << "Введите координату x: ";
cin >> x;
cout << "Введите координату y: ";
cin >> y;
cout << "Введите число пассажиров: ";
cin >> number;
cout << "Введите порт, к которому прикреплено судно: ";
cin >> port;

```

```

ship.SetPrise(price);
ship.SetSpeed(speed);
ship.SetYear(year);
ship.SetX(x);
ship.SetY(y);
ship.SetNumber(number);
ship.SetPort(port);

```

```

cout << endl << "Самолёт" << endl;
cout << plane.GetPrice() << " " << plane.GetSpeed() << " " << plane.GetYear() \
<< " " << plane.GetX() << " " << plane.GetY() << " " << plane.GetHigh();

```

```

cout << endl << "Машина" << endl;
cout << car.GetPrice() << " " << car.GetSpeed() << " " << car.GetYear() \
<< " " << car.GetX() << " " << car.GetY() << " " << car.GetNumbet();

```

```

cout << endl << "Корабль" << endl;
cout << ship.GetPrice() << " " << ship.GetSpeed() << " " << ship.GetYear() \
<< " " << ship.GetX() << " " << ship.GetY() << " " << ship.GetNumbet() << " "
<< ship.GetPort();

return 0;
}

```

Файл Vehicle.h

```

#ifndef VEHICLE_H
#define VEHICLE_H

```

```

class Vehicle
{
protected:
    int price = 0;

```

```

    int speed = 0;
    int year_of_issue = 0;
    int x_coord = 0;
    int y_coord = 0;

public:
    void SetPrice(int price);
    void SetSpeed(int speed);
    void SetYear(int year);
    void SetX(int x_coord);
    void SetY(int y_coord);

    int GetPrice() {return price;}
    int GetSpeed() {return speed;}
    int GetYear() {return year_of_issue;}
    int GetX() {return x_coord;}
    int GetY() {return y_coord;}
};

#endif // VEHICLE_H

```

Файл Vehicle.cpp

```

#include "Vehicle.h"

void Vehicle::SetPrice(int price)
{
    if (price >= 0) this->price = price;
}

void Vehicle::SetSpeed(int speed)
{
    if (speed >= 0) this->speed = speed;
}

void Vehicle::SetYear(int year)
{
    if ((year > 1900) && (year < 2025)) year_of_issue = year;
}

void Vehicle::SetX(int x_coord)
{
    this->x_coord = x_coord;
}

void Vehicle::SetY(int y_coord)
{
    this->y_coord = y_coord;
}

```

Файл Car.h

```

#ifndef CAR_H
#define CAR_H
#include "Vehicle.h"

class Car : public Vehicle //Объявление и наследование от Vehicle
{
private:

```

```

        int number = 0;

    public:
        void SetNumber(int number);

        int GetNumbet() {return number;}
};

#endif // CAR_H

```

Файл Car.cpp

```

#include "Car.h"

void Car::SetNumber(int number)
{
    if (number > 0) this->number = number;
}

```

Файл Plane.h

```

#ifndef PLANE_H
#define PLANE_H
#include "Vehicle.h"

class Plane : public Vehicle
{
    private:
        int high = 0;

    public:
        void SetHigh(int high);

        int GetHigh() {return high;}
};

#endif // PLANE_H

```

Файл Plane.cpp

```

#include "Plane.h"

void Plane::SetHigh(int high)
{
    if (high >= 0) this->high = high;
}

```

Файл Ship.h

```

#ifndef SHIP_H
#define SHIP_H
#include "Car.h"
#include <iostream>

using namespace std;

class Ship : public Car
{

```



```
private:
    string port = "Порт не определён";

public:
    void SetPort(string port);

    string GetPort() {return port;}
};

#endif // SHIP_H
```

Файл Ship.cpp

```
#include "Ship.h"

void Ship::SetPort(string port)
{
    this->port = port;
}
```