

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ»)  
Факультет информационных технологий компьютерной безопасности  
Кафедра автоматизированных вычислительных систем

**ОТЧЕТ**

По лабораторной работе № 3 по дисциплине  
«Среды визуального программирования»

Выполнил

Сафарзода Ф.Ф

---

инициалы, фамилия

Руководитель

Сокольников В.В

---

инициалы, фамилия

«4» октября 2023 г.

Воронеж 2023

## Задача

### СВП. ЛАБОРАТОРНАЯ РАБОТА 3

**Тема:** Интерфейсы и абстрактные классы в C++.

**Задание:** Моделирование процесса движения транспортного средства. Консольное приложение.

Шаги:

1. Взять за основу результаты лабораторной работы 2.
2. Создать класс-интерфейс **IVehicle** (Транспортное средство) с пустыми *виртуальными* методами:
  - `virtual void drive(int kilometres) = 0;`
  - `virtual void refuel() = 0`
  - `virtual void printStatus() = 0.`
3. Переименовать класс **Vehicle** в **AbstractCar** и унаследовать этот класс от класса-интерфейса **IVehicle**. Содержание класса **AbstractCar** должно остаться тем же, что и было в классе **Vehicle**.

INOBITES



Рис.1 - Описание задачи №1

### СВП. ЛАБОРАТОРНАЯ РАБОТА 3

4. Изменить базовый класс для классов **Sedan**, **Suv** и **Bus** с **Vehicle** на **AbstractCar**.

5. Создать класс **Bicycle** (Велосипед), унаследованный от класса-интерфейса **IVehicle**. Реализовать в классе все методы интерфейса. Так как у велосипеда топливного бака нет, то реализация метода **refuel()** будет пустой, т.е. ничего не выполнять. При реализации метода **drive()** не нужно учитывать расход топлива - считаем, что велосипед может ехать бесконечно долго. В методе **printStatus()** выводим только пробег, т.к. никакого расхода топлива нет.

Таким образом, нам удастся внедрить в существующую модель транспортное средство, которое не расходует топливо. Абстракция "велосипед" в этом плане существенно отличается от абстракции "автомобиль". Применяя класс-интерфейс без реализации методов и множественное наследование мы получаем нужный результат с **минимальными изменениями кода** программы из лабораторной работы 2.

INOBITES



Рис.2 - Описание задачи №2

## СВП. ЛАБОРАТОРНАЯ РАБОТА 3

6. Функция `main()` в итоге может быть примерно такой:

```
int main()
{
    IVehicle* vehicles[4] = {new Sedan, new Suv, new Bus, new Bicycle}; // Создаем объекты классов для моделирования
    for (int i = 0; i < 4; ++i)
    {
        IVehicle* vehicle = vehicles[i];
        vehicle->refuel(); // Заправляем полный бак
        vehicle->printStats(); // Выводим пробег и оставшееся топливо
        vehicle->drive(120); // Едем 120 км
        vehicle->printStats();
        vehicle->drive(420);
        vehicle->printStats();
        vehicle->drive(300);
        vehicle->printStats();
        vehicle->refuel(); // Дозаправляемся до полного бака
        vehicle->drive(300);
        vehicle->printStats();
        delete vehicle;
        cout << endl;
    }
    return 0;
}
```

INOBITES



Рис.3 - Описание задачи №3

Решение:

```
#include <iostream>
```

```
using namespace std;
```

```
class IMechine
{
```

```
    public:
```

```
        virtual void drive(int driven)=0;
```

```
        virtual void reBenzin()=0;
```

```
        virtual void printStatus()=0;
```

```
};
```

```
class AbstractCar:public IMechine
```

```
{
```

```
    protected:
```

```
        int Benzmax=0,probeg=0, Benzin=0,BenzinTrata=0;
```

```
    public:
```

```
    AbstractCar()
```

```
    {
```

```
    }
```

```

void reBenzin()
{
    Benzin=Benzmax;
}
void drive(int driven)
{
    probeg+=driven;
    Benzin-=driven*BenzinTrata;
}
void printStatus()
{
    cout<<"Benzin= "<<Benzin<<endl;
    cout<<"Drive= "<<probeg<<endl;
}
};

class Bicycle:public IMechine
{
    private:
        int probeg;
public:
    Bicycle() : probeg(0) {}

    void drive(int driven) {
        probeg += driven;
    }

    void reBenzin() {}

    void printStatus() {
        cout << "Probeg: " << probeg << endl;
    }
};

class Sedan:public AbstractCar
{
    public:
        Sedan()
        {
            Benzmax=70;
            BenzinTrata=2;
        }
};

class Suv:public AbstractCar

```

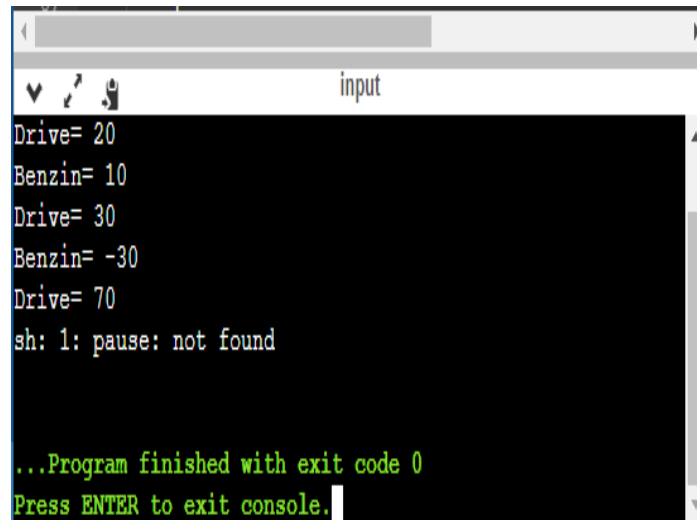
```

{
    public:
        Suv()
        {
            Benzmax=100;
            BenzinTrata=3;
        }
};
class Bus:public AbstractCar
{
    public:
        Bus()
        {
            Benzmax=300;
            BenzinTrata=5;
        }
};
int main()
{
    Bicycle bicycle;
    Sedan sedan;
    Suv suv;
    Bus bus;
    bicycle.drive(10);
    bicycle.printStatus();
    sedan.reBenzin();
    sedan.drive(20);
    sedan.printStatus();
    suv.reBenzin();
    suv.drive(30);
    suv.printStatus();
    sedan.reBenzin();
    sedan.drive(50);
    sedan.printStatus();
    system("pause");

    return 0;
}

```

Результат:

A screenshot of a terminal window titled 'input'. The window has a standard Linux-style title bar with minimize, maximize, and close buttons. The terminal content is as follows:

```
Drive= 20
Benzin= 10
Drive= 30
Benzin= -30
Drive= 70
sh: 1: pause: not found

...Program finished with exit code 0
Press ENTER to exit console.
```

The text is white on a black background. The final two lines are in green. A white cursor is positioned at the end of the last line.

Рис.4 - Результат кода