

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»
(ФГБОУ ВО «ВГТУ», ВГТУ)

ОТЧЕТ ПО ПРЕДМЕТУ СРЕДЫ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ
ЛАБОРАТОРНАЯ 3.

Обучающийся _____ Самохин Александр Геннадьевич
(ФИО обучающегося)

Группа БИВТ-222

Наименование предприятия ВГТУ

Обучающийся	_____	А.Г. Самохин
	(подпись)	

Руководитель по практической подготовке	_____	А.С. Троценко
	(подпись)	

```

aaaa.h  X  aaaa.cpp
aaaa
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class IVehicle
6  {
7  public:
8      virtual void drive(int kilometres) = 0;
9      virtual void refuel() = 0;
10     virtual void printStatus() = 0;
11 };
12
13
14
15 class AbstractCar : public IVehicle
16 {
17 protected:
18     int mileage; // пробег
19     double fuel; // количество топлива в баке (в литрах)
20     const double tankCapacity; // вместимость бака (в литрах)
21     const double fuelConsumption; // расход топлива (литры на 100 км)
22 public:
23     AbstractCar(double capacity, double consumption) : mileage(0), fuel(0), tankCapacity(capacity), fuelConsumption(consumption) {}
24     void drive(int kilometers)
25     {
26         double fuelNeeded = (static_cast<double>(kilometers) / 100.0) * fuelConsumption;
27         if (fuel >= fuelNeeded)
28         {
29             mileage += kilometers;
30             fuel -= fuelNeeded;
31             cout << "Проехали " << kilometers << " км. Осталось топлива: " << fuel << " литров." << endl;
32         }
33         else
34         {
35             cout << "Недостаточно топлива для поездки на " << kilometers << " км." << endl;
36         }
37     }
38     void refuel()
39     {
40         fuel = tankCapacity;
41         cout << "Бак заправлен до полного. Теперь у вас " << fuel << " литров топлива." << endl;
42     }
43     void printStatus()
44     {
45         cout << "Пробег: " << mileage << " км. Осталось топлива: " << fuel << " литров." << endl;
46     }
47 };
48
49 class Sedan : public AbstractCar
50 {
51 public:
52     Sedan() : AbstractCar(50.0, 6.5) {}
53 };
54
55 class Suv : public AbstractCar
56 {
57 public:
58     Suv() : AbstractCar(70.0, 8.0) {}

```

Рисунок 1.

```

56 {
57 public:
58     Suv() : AbstractCar(70.0, 8.0) {}
59 };
60
61 class Bus : public AbstractCar
62 {
63 public:
64     Bus() : AbstractCar(200.0, 15.0) {}
65 };
66
67 class Bicycle : public IVehicle
68 {
69 protected:
70     int mileage;
71
72 public:
73     void drive(int kilometers) override
74     {
75         mileage += kilometers;
76         cout << "Проехали: " << kilometers << endl;
77     };
78
79     void printStatus() override
80     {
81         cout << "Пробег: " << mileage << " км." << endl;
82     };
83
84     void refuel() override {};
85 };

```

Рисунок 2.

```

#include "aaaa.h"

int main() {
    setlocale(0, "");

    IVehicle* vehicles[4] = { new Sedan(), new Suv(), new Bus(), new Bicycle() };

    for (int i = 0; i < 4; ++i)
    {
        IVehicle* vehicle = vehicles[i];
        vehicle->refuel();
        vehicle->printStatus();
        vehicle->drive(120);
        vehicle->printStatus();
        vehicle->drive(420);
        vehicle->printStatus();
        vehicle->drive(300);
        vehicle->printStatus();
        vehicle->refuel();
        vehicle->drive(300);
        vehicle->printStatus();
        delete vehicle;
        cout << endl;
    }

    return 0;
}

```

Рисунок 3.

```
Консоль отладки Microsoft V X + v

Бак заправлен до полного. Теперь у вас 50 литров топлива.
Пробег: 0 км. Осталось топлива: 50 литров.
Проехали 120 км. Осталось топлива: 42.2 литров.
Пробег: 120 км. Осталось топлива: 42.2 литров.
Проехали 420 км. Осталось топлива: 14.9 литров.
Пробег: 540 км. Осталось топлива: 14.9 литров.
Недостаточно топлива для поездки на 300 км.
Пробег: 540 км. Осталось топлива: 14.9 литров.
Бак заправлен до полного. Теперь у вас 50 литров топлива.
Проехали 300 км. Осталось топлива: 30.5 литров.
Пробег: 840 км. Осталось топлива: 30.5 литров.

Бак заправлен до полного. Теперь у вас 70 литров топлива.
Пробег: 0 км. Осталось топлива: 70 литров.
Проехали 120 км. Осталось топлива: 60.4 литров.
Пробег: 120 км. Осталось топлива: 60.4 литров.
Проехали 420 км. Осталось топлива: 26.8 литров.
Пробег: 540 км. Осталось топлива: 26.8 литров.
Проехали 300 км. Осталось топлива: 2.8 литров.
Пробег: 840 км. Осталось топлива: 2.8 литров.
Бак заправлен до полного. Теперь у вас 70 литров топлива.
Проехали 300 км. Осталось топлива: 46 литров.
Пробег: 1140 км. Осталось топлива: 46 литров.

Бак заправлен до полного. Теперь у вас 200 литров топлива.
Пробег: 0 км. Осталось топлива: 200 литров.
Проехали 120 км. Осталось топлива: 182 литров.
Пробег: 120 км. Осталось топлива: 182 литров.
Проехали 420 км. Осталось топлива: 119 литров.
Пробег: 540 км. Осталось топлива: 119 литров.
Проехали 300 км. Осталось топлива: 74 литров.
Пробег: 840 км. Осталось топлива: 74 литров.
Бак заправлен до полного. Теперь у вас 200 литров топлива.
Проехали 300 км. Осталось топлива: 155 литров.
Пробег: 1140 км. Осталось топлива: 155 литров.

Пробег: 0 км.
Проехали: 120
Пробег: 120 км.
Проехали: 420
Пробег: 540 км.
Проехали: 300
Пробег: 840 км.
Проехали: 300
Пробег: 1140 км.

C:\Users\sanya\OneDrive\Рабочий стол\c++\aaaa\x64\Debug\aaaa.exe (процесс 28616) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 4.

Пояснение задания:

Этот код на C++ представляет собой пример использования полиморфизма и абстрактных классов для создания и управления различными видами транспортных средств (автомобили, автобус и велосипед) с общим интерфейсом.

Вот основные компоненты этой программы:

`IVehicle` - это абстрактный класс, представляющий интерфейс для всех видов транспортных средств. Он имеет три чисто виртуальных метода: `drive`, `refuel` и `printStatus`, которые должны быть реализованы в производных классах.

`AbstractCar` - это абстрактный класс, который наследуется от `IVehicle` и представляет общие характеристики для всех автомобилей. Этот класс содержит информацию о пробеге, уровне топлива в баке, вместимости бака и расходе топлива. Он также реализует методы интерфейса `IVehicle` в соответствии с характеристиками автомобилей.

`Sedan`, `Suv` и `Bus` - это конкретные классы, представляющие разные типы автомобилей. Они наследуются от `AbstractCar` и определяют конкретные параметры вместимости бака и расхода топлива для каждого типа автомобиля.

`Bicycle` - это класс, представляющий велосипед, который также наследуется от `IVehicle`. Велосипед не имеет бака для топлива, поэтому метод `refuel` оставлен пустым, а методы `drive` и `printStatus` реализованы соответственно для велосипеда.

В функции `main`, создаются объекты разных типов транспортных средств, включая `Sedan`, `Suv`, `Bus` и `Bicycle`. Затем происходит итерация по массиву `vehicles`, и для каждого транспортного средства выполняются следующие действия:

Вызывается метод `refuel` для заправки транспортного средства.

Выводится статус транспортного средства с помощью метода `printStatus`.

Вызывается метод `drive` для преодоления определенного расстояния.

Снова выводится статус транспортного средства.

Затем транспортное средство заправляется повторно, и статус выводится снова.

Затем объект удаляется.

Этот код демонстрирует использование полиморфизма, так как все объекты в массиве `vehicles` имеют общий интерфейс `IVehicle`, и методы вызываются в зависимости от конкретного типа объекта.