

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 2.

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)	

Воронеж 2023

```

lab 2.h* X lab 2.cpp
lab 2
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5
6 class Vehicle
7 {
8 protected:
9     int mileage; // пробег
10    double fuel; // количество топлива в баке (в литрах)
11    const double tankCapacity; // вместимость бака (в литрах)
12    const double fuelConsumption; // расход топлива (литры на 100 км)
13 public:
14    Vehicle(double capacity, double consumption) : mileage(0), fuel(0), tankCapacity(capacity), fuelConsumption(consumption) {}
15    void drive(int kilometers)
16    {
17        double fuelNeeded = (static_cast<double>(kilometers) / 100.0) * fuelConsumption;
18        if (fuel >= fuelNeeded)
19        {
20            mileage += kilometers;
21            fuel -= fuelNeeded;
22            cout << "Проехали " << kilometers << " км. Осталось топлива: " << fuel << " литров." << endl;
23        }
24        else
25        {
26            cout << "Недостаточно топлива для поездки на " << kilometers << " км." << endl;
27        }
28    }
29    void refuel()
30    {
31        fuel = tankCapacity;
32        cout << "Бак заправлен до полного. Теперь у вас " << fuel << " литров топлива." << endl;
33    }
34    void printStatus()
35    {
36        cout << "Пробег: " << mileage << " км. Осталось топлива: " << fuel << " литров." << endl;
37    }
38 };
39
40 class Sedan : public Vehicle
41 {
42 public:
43     Sedan() : Vehicle(50.0, 6.5) {}
44 };
45
46 class Suv : public Vehicle
47 {
48 public:
49     Suv() : Vehicle(70.0, 8.0) {}
50 };
51
52 class Bus : public Vehicle
53 {
54 public:
55     Bus() : Vehicle(200.0, 15.0) {}
56 };

```

Рисунок 1.

```
#include "lab 2.h"

int main() {

    setlocale(0, "");

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

    for (int i = 0; i < 4; ++i)
    {
        Vehicle* 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 2.

Пояснение задания:

Этот код на C++ описывает классы для различных видов автомобилей и демонстрирует их использование в функции `main()`. Давайте разберемся с каждой его частью:

1. В начале программы включены необходимые заголовки, такие как `<iostream>` и `<string>`, а также используется директива `using namespace std;`, чтобы облегчить доступ к элементам стандартной библиотеки;

2. Затем определен базовый класс `Vehicle`, который представляет собой абстрактное транспортное средство. Он имеет следующие `protected` (защищенные) члены данных:

`mileage` (пробег) - хранит текущий пробег автомобиля.

`fuel` (количество топлива в баке) - хранит количество топлива в баке в литрах.

`tankCapacity` (емкость бака) - хранит емкость бака в литрах

(константное значение);

`fuelConsumption` (расход топлива) - хранит расход топлива в литрах на 100 км

(константное значение).

В конструкторе класса `Vehicle` происходит инициализация этих членов данных на основе переданных значений емкости бака и расхода топлива.

3. У класса `Vehicle` есть несколько методов:

`drive(int kilometers)` - метод, который позволяет совершить поездку на заданное количество километров. Внутри метода производится расчет необходимого количества топлива для поездки, и если в баке достаточно топлива, пробег увеличивается, а количество топлива уменьшается. Иначе выводится сообщение о недостатке топлива;

`refuel()` - метод, который позволяет заправить бак до полного.

`printStatus()` - метод, который выводит информацию о текущем пробеге и количестве топлива.

4. Затем в коде определены три производных класса: `Sedan`, `Suv` и `Bus`, которые наследуются от базового класса `Vehicle`. Каждый из этих классов представляет

определенный тип автомобиля и имеет свои собственные параметры вместимости бака и расхода топлива;

5. В функции `main()`, создаются объекты каждого из классов (`Sedan`, `Suv`, `Bus`) и помещаются в массив указателей на базовый класс `Vehicle`. Затем происходит итерация по массиву, для каждого объекта выполняются следующие действия: Вызывается метод `refuel()` для заправки бака.

Вызывается метод `printStatus()` для вывода информации о текущем состоянии автомобиля.

Вызывается метод `drive()` для совершения нескольких поездок разной длины.

Снова вызывается `printStatus()` для вывода информации после каждой поездки.

Вызывается метод `refuel()` еще раз, чтобы заправить бак до полного.

Затем объект удаляется с помощью `delete` для освобождения памяти.

В результате выполнения этой программы вы получите вывод, который покажет состояние каждого автомобиля после различных действий (заправка, поездки и т.д.).