

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 \_\_\_\_\_ Тихоненко Дмитрий Анатольевич  
(ФИО обучающегося)

Группа БИВТ-222

Наименование предприятия ВГТУ

Обучающийся	_____	Д.А. Тихоненко
	(подпись)	

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

```

Users > macbookpro13retina2015 > Desktop > C laba3.h > ...
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     }

```

Строка 86, столбец 1 Пробелов: 4 UTF-8 LF {}

```

Users > macbookpro13retina2015 > Desktop > C laba3.h > ...
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) {}
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;

```

Строка 86, столбец 1 Пробелов: 4 UTF-8 LF {}

```

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     };
86

```

```

Users > macbookpro13retina2015 > Desktop > C++ lab3.cpp > main()
1  #include "lab3.h"
2  int main() {
3      setlocale(0, "");
4      IVehicle* vehicles[4] = { new Sedan(), new Suv(), new Bus(), new Bicycle() };
5      for (int i = 0; i < 4; ++i) {
6          IVehicle* vehicle = vehicles[i];
7          vehicle->refuel();
8          vehicle->printStatus();
9          vehicle->drive(120);
10         vehicle->printStatus();
11         vehicle->drive(420);
12         vehicle->printStatus();
13         vehicle->drive(300);
14         vehicle->printStatus();
15         vehicle->refuel();
16         vehicle->drive(300);
17         vehicle->printStatus();
18         delete vehicle;
19         cout << endl;
20     }
21     return 0;}

```

### Пояснение задачи:

Этот код представляет собой более абстрактную реализацию транспортных средств с использованием интерфейса и абстрактного базового класса.

Интерфейс IVehicle:

Определяет интерфейс для транспортных средств с чисто виртуальными функциями drive, refuel и printStatus.

AbstractCar:

Абстрактный базовый класс, реализующий интерфейс IVehicle.

Содержит защищенные члены данных: mileage (пробег), fuel (количество топлива), tankCapacity (емкость бака) и fuelConsumption (расход топлива).

Реализует методы интерфейса, такие как drive, refuel и printStatus, общие для всех транспортных средств.

Конструктор этого класса инициализирует параметры топливного бака и расхода топлива.

Классы Sedan, Suv, Bus:

Эти классы являются конкретными реализациями транспортных средств на основе AbstractCar.

Каждый класс инициализирует параметры базового класса своими уникальными значениями в конструкторе.

Класс Bicycle:

Реализует интерфейс IVehicle.

Представляет велосипед, который не использует топливо.

Реализует методы drive, refuel (пустая реализация, так как велосипед не заправляется) и printStatus.

Функция main:

Создает массив указателей на объекты интерфейса IVehicle, включая объект велосипеда.

В цикле выполняются различные действия для каждого транспортного средства, такие как заправка, поездки и вывод статуса.

Затем освобождает выделенную память с использованием оператора delete.

Этот код обеспечивает более гибкую структуру для различных типов транспортных средств, позволяя добавлять новые классы без изменения кода в функции main, благодаря использованию интерфейса и абстрактного базового класса.