



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т (В А Р И А Н Т 1 2)

по лабораторной работе № 1

Название: **Простые объекты**

Дисциплина: Объектно-ориентированное программирование

Студент

ИУ6-23Б

(Группа)

10.05.2023

(Подпись, дата)

В.К. Залыгин

(И.О. Фамилия)

Преподаватель

А.М. Минитаева

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель работы

Научиться работать с объектной моделью языка C++ на простом уровне.

Задание

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Протестировать все методы класса. Поля класса должны быть скрытыми (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – внедорожник. Поля: марка, объем топливного бака, расход топлива на 100 км пути. Методы: процедура инициализации, процедура вывода информации об объекте на экран, функция вычисления запаса хода.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Проект программы

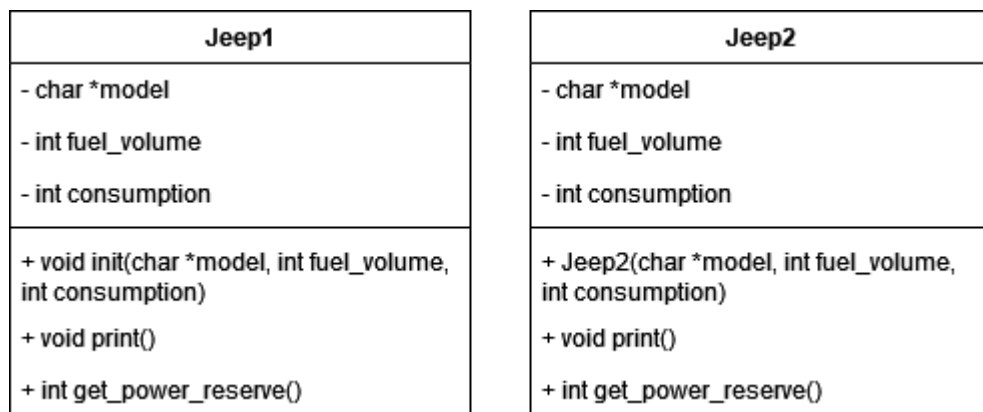


Рисунок 1 - диаграмма классов

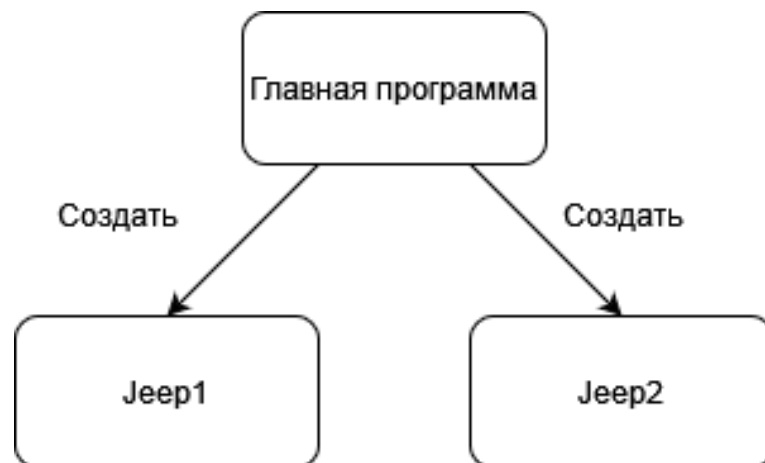


Рисунок 2 - объектная декомпозиция

Текст программы

```
You, 3 days ago | 1 author (You)
1  #include "jeep1.h"
2  #include "jeep2.h"
3  #include <cstring>
4  #include <iostream>
5
6  using namespace std;
7
8  int main() {
9      Jeep1 j1;
10     j1.init("ford focus", 30, 10);
11     j1.print();
12     cout << "power reverse is " << j1.get_power_reserve() << endl;
13
14     Jeep2 j2("mazda cx5", 55, 15);
15     j2.print();
16     cout << "power reserve is " << j2.get_power_reserve() << endl;
17
18     return 0;
19 }
```

Рисунок 3 - код файла main.cpp

```
1  class Jeep1 {
2  private:
3      char *model;
4      int fuel_volume;
5      int consumption;
6
7  public:
8      void init(char *model, int fuel_volume, int consumption);
9      void print();
10     int get_power_reserve();
11 };
```

Рисунок 4 - код файла jeep1.h

```
1  #include <iostream>
2  #include "jeep1.h"
3
4  using namespace std;
5
6  void Jeep1::init(char *model, int fuel_volume, int consumption) {
7      this->model = model;
8      this->fuel_volume = fuel_volume;
9      this->consumption = consumption;
10 }
11
12 void Jeep1::print() {
13     cout << "Jeep; model: " << model << "; fuel_volume: " << fuel_volume
14     | | << "; consumption: " << consumption << ";" << endl;
15 }
16
17 int Jeep1::get_power_reserve() { return (fuel_volume * 100) / consumption; }
```

Рисунок 5 - код файла jeep1.cpp

```

1  class Jeep2 {
2  private:
3      char *model;
4      int fuel_volume;
5      int consumption;
6
7  public:
8      Jeep2(char *model, int fuel_volume, int consumption);
9      void print();
10     int get_power_reserve();
11 };

```

Рисунок 6 - код файла jeep2.h

```

1  #include "jeep2.h"
2  #include <iostream>
3
4  using namespace std;
5
6  Jeep2::Jeep2(char *model, int fuel_volume, int consumption)
7      : model(model), fuel_volume(fuel_volume), consumption(consumption) {}
8
9  void Jeep2::print() {
10     cout << "Jeep; model: " << model << "; fuel_volume: " << fuel_volume
11     << "; consumption: " << consumption << ";" << endl;
12 }
13
14 int Jeep2::get_power_reserve() { return (fuel_volume * 100) / consumption; }

```

Рисунок 7 - код файла jeep2.cpp

Тестовые данные

```

$ ./main
Jeep; model: ford focus; fuel_volume: 30; consumption: 10;
power reserve is 300
Jeep; model: mazda cx5; fuel_volume: 55; consumption: 15;
power reserve is 366

```

Рисунок 8 - результат работы

Вывод

Были изучены основы работы с объектной моделью C++.