

Министерство Образования Республики Беларусь
УО Брестский Государственный Технический Университет
Кафедра интеллектуальных информационных технологий

Лабораторная работа №5.
"Классы и объекты C++"

Выполнил:
Студент 2-го курса
Группы АС-53
Вожейко М.В.
Проверил:
Давидюк Ю. И.

Брест, 2020

1. Цель. Получить практические навыки реализации классов на C++.

2. Постановка задачи (Вариант 6)

Автомобиль

имя – char*

мощность – int

стоимость – float

Конструкторы:

- Без параметров
- С 3 параметрами
- Копирования

Функции:

- Возврата
- Вывода
- Аллокации памяти

3. Определение пользовательского класса с комментариями.

```
class Car {  
private:  
    char* name; // Марка  
    int power = 0; // Мощность  
    float cost = 0.0f; // Стоимость  
    void allocateChars(const char* _name);  
  
public:  
    Car();  
    Car(const char* _name, int _power, float _disp);  
    Car(const Car&);  
    ~Car();  
    inline char* getName() const { return name; }  
    inline int getPower() const { return power; }  
    inline float getCost() const { return cost; }  
    void print();  
};
```

4. Реализация конструкторов и деструктора и других функций

```
void Car::allocateChars(const char* _name) { // Выделяем память  
    name = new char[strlen(_name)+1];  
    strcpy(name, _name);  
}  
  
Car::Car() { // Конструктор без параметров  
    allocateChars("Unnamed");  
    std::cout << "Create class Car without constructor params." << std::endl;  
}  
  
Car::Car(const char* _name, int _power, float _cost) { // Конструктор с 3 параметрами  
    allocateChars(_name);  
    power = _power;  
    cost = _cost;  
    std::cout << "Create class Car with 3 constructor params." << std::endl;  
}  
  
Car::Car(const Car& _Car) { // Конструктор копирования
```

```

        allocateChars(_Car.name);
        power = _Car.power;
        cost = _Car.cost;
        std::cout << "Create class Car with copy constructor." << std::endl;
    }

    Car::~Car() { // Деструктор
        std::cout << "Call Car class \"" << name << "\" desctructor." << std::endl;
        delete[] name;
    }

    void Car::print() { // Вывод
        std::cout << "\nName: " << name << "\nPower: " << power << "HP"
            << "\nCost: " << cost << "$\n" << std::endl;
    }
}

```

5. Фрагмент программы, показывающий использование указателя на объект с объяснением.

`Car* Car1 = new Car("Audi A4 B7", 142, 60000.0);` – создаем объект класса с именем `Car1` параметрами «Audi A4 B7», 142, 60000.0 и получаем на него указатель.

6. Листинг основной программы, в котором должно быть указано, в каком месте и какой конструктор или деструктор вызываются.

```

int main() {
    Car* Car1 = new Car("Audi A4 B7", 142, 60000.0);
    Car Car2;
    Car Car3(*Car1);

    Car1->print();
    Car2.print();
    Car3.print();

    return 0;
}

```

```

Create class Car with 3 constructor params.
Create class Car without constructor params.
Create class Car with copy constructor.

Name: Audi A4 B7
Power: 142HP
Cost: 60000$

Name: Unnamed
Power: 0HP
Cost: 0$

Name: Audi A4 B7
Power: 142HP
Cost: 60000$

Call Car class "Audi A4 B7" desctructor.
Call Car class "Audi A4 B7" desctructor.
Call Car class "Unnamed" desctructor.

```

6. Вывод:

Получил практические навыки реализации классов на C++.