

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

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

Выполнил:
Ст. 2 курса гр. АС-53
Демидович А. Г.
Проверила:
Давидюк Ю. И.

Брест, 2020

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

1. Постановка задачи (Вариант 8)

8. ТОВАР

имя – char*

количество – int

стоимость – float

Конструкторы:

- Без параметров
- С 3 параметрами
- Копирования

Функции:

- Вывода
- Аллокации памяти

2. Определение пользовательского класса с комментариями.

```
#pragma once
```

```
class Good {  
private:  
    char* name;  
    int count;  
    float cost;  
    void allocName(const char* _name);  
  
public:  
    Good();  
    Good(const char* _name, int _count, float _cost);  
    Good(const Good&);  
    ~Good();  
    void print();  
};
```

3. Реализация конструкторов и деструктора и других функций

```
#include <iostream>  
#include "good.h"
```

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

```

        allocName(_Good.name);
        count = _Good.count;
        cost = _Good.cost;
        std::cout << "Create class " << name << " with copy constructor." << std::endl;
    }

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

    void Good::print() { // Вывод
        std::cout << "\nName: " << name <<
            "\nCount: " << count <<
            "\nCost: " << cost << "$\n" << std::endl;
    }
}

```

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

`Good* Good1 = new Good("Toy", 200, 10);` - создаем объект класса `Good` с именем `Good1` параметрами «Toy», 200, 10 и получаем на него указатель.

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

```

#include <iostream>
#include "good.h"
int main() {
    Good* Good1 = new Good("Toy", 200, 10);
    Good Good2;
    Good Good3(*Good1);

    Good1->print();
    Good2.print();
    Good3.print();

    delete(Good1);

    return 0;
}

```

```
Create class Toy with 3 constructor parametrs.  
Create class Untitled without constructor parametrs.  
Create class Toy with copy constructor.  
  
Name: Toy  
Count: 200  
Cost: 10$  
  
Name: Untitled  
Count: 0  
Cost: 0$  
  
Name: Toy  
Count: 200  
Cost: 10$  
  
Call Good class "Toy" desctructor.  
Call Good class "Toy" desctructor.  
Call Good class "Untitled" desctructor.
```

6. Вывод:

Получила практические навыки реализации классов на C++.