

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

**ОТЧЁТ**  
по лабораторной работе №1  
**«Классы и объекты в C++»**

Выполнил:

Студент 2 курса  
группы ПО-9  
Харитонович Захар Сергеевич

Проверил:

Козик И. Д.

Брест 2023

## Вариант 8

### Ход работы

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение).
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных.
5. Определить указатель на компоненту-функцию.
6. Определить указатель на экземпляр класса.
7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).
8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

### ТОВАР

имя – char\*

количество – int

стоимость – float

### Текст программы

#### *Product.h*

```
#ifndef LAB1_PRODUCT_H
#define LAB1_PRODUCT_H
#include <string.h>
#include <iostream>
class Product {
public:
    Product(); // конструктор без параметров
    Product(const char *, const int, const float); // конструктор с
    параметрами
    Product(const Product &); // конструктор копирования
    ~Product(); // деструктор
    char *getName();
    void setName(char *name);
    int getAmount();
    void setAmount(int amount);
    float getPrice();
    void setPrice(float price);
    void display(); // вывод содержимого класса
private:
    char * name = new char[256]; // имя
    int amount; // количество
    float price; // цена
};
#endif
```

#### *Product.cpp*

```
#include "Product.h"

Product::Product(): amount(0), price(0) {
    strcpy(name, "default");
    std::cout << "Object " << this << " created." << std::endl;
}

Product::Product(const char * name, const int amount, const float price):
amount(amount), price(price) {
```

```

        strcpy(this->name, name);
        std::cout << "Object " << this << " created with parameters." <<
std::endl;
    }
    Product::Product(const Product & source): amount(source.amount),
price(source.price){
        strcpy(this->name, source.name);
        std::cout << "Object " << this << " created with copying." << std::endl;
    }
    Product::~~Product() {
        delete[] name;
        std::cout << "Object " << this << " deleted." << std::endl;
    }
    void Product::display() {
        std::cout << "Name: " << name << "; Amount: " << amount << "; Price: " <<
price << ";" << std::endl << std::endl;
    }
    char *Product::getName() { return name; }
    void Product::setName(char *name) { Product::name = name; }
    int Product::getAmount() { return amount; }
    void Product::setAmount(int amount) { Product::amount = amount; }
    float Product::getPrice() { return price; }
    void Product::setPrice(float price) { Product::price = price; }

```

### **main.cpp**

```

#include <iostream>
#include "Product.h"
int main() {
    Product def; // конструктор по умолчанию
    def.display();
    def.setName("Bread"); // методы модификации полей
    def.setAmount(4);
    def.setPrice(1.5);
    def.display();
    Product pr2("Milk", 23, 2.2); // конструктор с параметрами
    pr2.display();
    Product pr2Clone(pr2); // конструктор копирования
    pr2Clone.display();
    Product defClone = def;
    defClone.display();
    Product * def_ptr = &def; // указатель на экземпляр класса
    def_ptr->setName("Yogurt");
    void (Product::*pf)(); // указатель на метод
    pf = &Product::display;
    (def.*pf)();
    return 0;
}

```

### **Результат работы:**

```

Object 0xa81d9ffc20 created.
Name: default; Amount: 0; Price: 0;
Name: Bread; Amount: 4; Price: 1.5;
Object 0xa81d9ffc10 created with parameters.
Name: Milk; Amount: 23; Price: 2.2;
Object 0xa81d9ffc00 created with copying.
Name: Milk; Amount: 23; Price: 2.2;
Object 0xa81d9ffbf0 created with copying.
Name: Bread; Amount: 4; Price: 1.5;
Name: Yogurt; Amount: 4; Price: 1.5;
Object 0xa81d9ffbf0 deleted.
Object 0xa81d9ffc00 deleted.
Object 0xa81d9ffc10 deleted.

```