Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

за 1 семестр

По дисциплине: «Языки программирования»

Тема: «Классы и объекты в С++»

Выполнила:

Студент 2 курса

Группы ПО-7(1)

Крук Д.Н.

Проверил:

Дряпко М. В.

2021

**Лабораторная работа №1**

**Классы и объекты в С++**

**Цель работы:** получить практические навыки реализации классов на С++. Написать программу, в которой создаются и разрушаются объекты, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.

**Вариант 8**

**Задание:**

1. Определить пользовательский класс в соответствии с вариантом задания.

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

Цена – float

1. **Определение пользовательского класса с комментариями:**

class Product

{

private:

    char \*name;

    int number;

    float price;

public:

    Product(); //Конструктор без параметров

    Product(char \*, int, float); //Конструктор с параметрами

    Product(Product &); //Конструктор копии

    ~Product(); //Деструктор класса

    char \*GatName(); //Функция получения имени товара

    int GetNumber(); //Функция получения количества товара

    float GetPrice(); //Функция получения цены товара

    void Show(); //Функция для просмотра полей экземпляра

    void SetName(char \*); //Функция для установки значения имени товара

    void SetNumber(int); //Функция для установки значения количества товаров

    void SetPrice(float); //Функция для установки значения цены товаров

    void Set(char\*, int, float); //Функция для установки значений полей товара

};

1. **Реализация конструкторов и деструкторов:**

Product::Product()

{

    name = "-";

    number = 0;

    price = 0;

    cout << "Конструктор без параметров вызван для объкта" << endl;

}

Product::Product(char \*NAME, int NUMBER, float PRICE)

{

    name = NAME;

    number = NUMBER;

    price = PRICE;

    cout << "Конструктор с параметрами вызван для объкта" << endl;

}

Product::Product(Product &Copy)

{

    name = Copy.name;

    number = Copy.number;

    price = Copy.price;

    cout << "Конструктор копированния вызван для объкта" << endl;

}

Product::~Product()

{

    delete name;

    delete &number;

    delete &price;

    cout << "Деструктор для объекта был вызван";

}

1. **Фрагмент программы, показывающий использование указателя на объект и указателя на функцию с объяснением:**

Product \*p = &List1[0]; //Объявление указателя на экземпляр класса

p->Show(); //Использование указателя на экземпляр класса

...

void (Product::\*f)(char\*, int, float) = &Product::Set; //Объявление указателя на

//функцию

(List2[1].\*f)("Конфеты", 100, 0.20); //Использование указателя на функцию

**Модуль class.h**

#ifndef CLASS\_H

#define CLASS\_H

#include <iostream>

#include <Windows.h>

using namespace std;

class Product {

private:

    char \*name;

    int number;

    float price;

public:

    Product(); //Конструктор без параметров

    Product(char \*, const int, const float); //Конструктор с параметрами

    Product(const Product &); //Конструктор копии

    ~Product(); //Деструктор класса

    char \*GatName(); //Функция получения имени товара

    int GetNumber(); //Функция получения количества товара

    float GetPrice(); //Функция получения цены товара

    void Show(); //Функция для просмотра полей экземпляра

    void SetName(char \*); //Функция для установки значения имени товара

    void SetNumber(int); //Функция для установки значения количества товаров

    void SetPrice(float); //Функция для установки значения цены товаров

    void Set(char \*, int, float); //Функция для установки значений полей экземпляра

};

#endif

**Модуль realization.cpp**

#include "class.h"

Product::Product() {

    name = "-";

    number = 0;

    price = 0;

    cout << "Конструктор без параметров вызван для объкта " << this << endl;

}

Product::Product(char \*NAME, const int NUMBER, const float PRICE) {

    name = NAME;

    number = NUMBER;

    price = PRICE;

    cout << "Конструктор с параметрами вызван для объкта " << this << endl;

}

Product::Product(const Product &Copy) {

    name = Copy.name;

    number = Copy.number;

    price = Copy.price;

    cout << "Конструктор копированния вызван для объкта " << this << endl;

}

Product::~Product() {

    delete name;

    delete &number;

    delete &price;

    cout << "Деструктор был вызван для объекта " << this << endl;

}

char \*Product::GatName() {

    return name;

}

int Product::GetNumber() {

    return number;

}

float Product::GetPrice() {

    return price;

}

void Product::Show() {

    cout << "Название: " << name << endl

         << "Количество: " << number << endl

         << "Цена: " << price << " руб." << endl << endl;

}

void Product::SetName(char \*new\_name) {

    name = new\_name;

}

void Product::SetNumber(int new\_number) {

    number = new\_number;

}

void Product::SetPrice(float new\_price) {

    price = new\_price;

}

void Product::Set(char \*new\_name, int new\_number, float new\_price) {

    name = new\_name;

    number = new\_number;

    price = new\_price;

}

**Модуль demonstration.cpp**

 #include "class.h"

Product Example(Product &a) {

    Product temp(a);

    temp.SetName("Чёрный хлеб");

    return temp;

}

int main() {

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    Product List1[] = {Product("Хлеб", 10, 0.79), //Выделения массива

                       Product("Батон", 5, 0.89)}; //в статической памяти

    Product \*p = &List1[0]; //Объявление указателя на экземпляр класса

    p->Show(); //Использование указателя на экземпляр класса

    List1[1].Show();

    Product \*List2 = new Product[2]; //Выделения массива в динамической памяти

    void (Product::\*f)(char \*, int, float) = &Product::Set; //Объявление указателя на функцию

    (List2[0].\*f)("Конфеты", 100, 0.20); //Использование указателя на функцию

    (List2[1].\*f)("Печенья", 124, 1.34); //Использование указателя на функцию

    List2[0].Show();

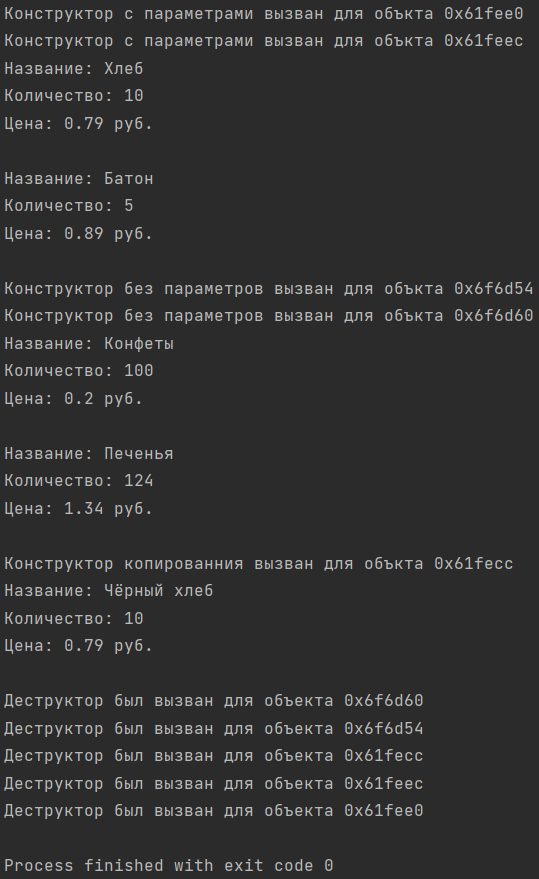
    List2[1].Show();

    Product Copy = Example(List1[0]);

    Copy.Show();

    delete[] List2;

}

**Пример выполнения программы:**

**Вывод:** получила практические навыки реализации классов на С++. Написала программу, в которой создаются и разрушаются объекты, определенного пользователем класса. Выполнила исследование вызовов конструкторов и деструкторов.