

МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«Брестский государственный технический университет»
Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №3
По дисциплине «Объектно-ориентированное программирование и
проектирование»
За 4 семестр
Тема: «Классы и объекты в C++»

Выполнила:
студентка 2 курса
группы АС-56
Карпенко М.В.

Проверил:
Давидюк Ю.И.

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

Постановка задачи. Вариант 13

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение):

13. КВИТАНЦИЯ

номер – int

дата – int

сумма – float

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 экземпляр класса.

6. 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).

7. Показать в программе использование указателя на объект.

Код программы:

RECEIPT.h

```
#pragma once
class RECEIPT
{
    int number;
    int data;
    float sum;
public:
    RECEIPT();
    RECEIPT(int number, int data, float sum);
    RECEIPT(const RECEIPT&);
    ~RECEIPT();

    void SetNumber(int number);
    void SetData(int data);
    void SetSum(float sum);

    int GetNumber();
    int GetData();
    float GetSum();

    void Show();
};
```

RECEIPT.cpp

```
#include "RECEIPT.h"
#include <iostream>
using namespace std;

RECEIPT::RECEIPT()
{
    cout << "Вызван конструктор по умолчанию" << endl;
    number = 0;
    data = 0;
    sum = 0;
}

RECEIPT::RECEIPT(int number, int data, float sum)
{
    cout << "Вызван конструктор с параметрами" << endl;
    this->number = number;
    this->data = data;
    this->sum = sum;
}

RECEIPT::RECEIPT(const RECEIPT& obj)
{
    cout << "Вызван конструктор копирования" << endl;
    number = obj.number;
    data = obj.data;
    sum = obj.sum;
}

RECEIPT::~RECEIPT()
{
    cout << "Вызван деструктор" << endl;
}

void RECEIPT::SetNumber(int numb)
{
    number = numb;
}

void RECEIPT::SetData(int dat)
{
    data = dat;
}

void RECEIPT::SetSum(float su)
{
    sum = su;
}

int RECEIPT::GetNumber()
{
    return number;
}

int RECEIPT::GetData()
{
    return data;
}

float RECEIPT::GetSum()
{
    return sum;
}

void RECEIPT::Show()
{
    cout << number << endl;
```

```

        cout << data << endl;
        cout << sum << endl;
    }

```

third_lab.cpp

```

#include <iostream>
#include "RECEIPT.h"

using namespace std;

void View(RECEIPT one_place)
{
    one_place.Show();
};

int main()
{
    setlocale(0, "");

    RECEIPT one;
    RECEIPT two(158,22,50.05);
    RECEIPT three(two);

    View(one);
    cout << endl;

    one.SetNumber(159);
    one.SetData(23);
    one.SetSum(75.99);

    int a = two.GetNumber();
    int b = two.GetData();
    int c = two.GetSum();

    cout << a << endl << b << endl << c << endl << endl;

    RECEIPT* p = new RECEIPT(11,22,33);
    cout << p->GetNumber() << endl << endl;

    one.Show();
    cout << endl;
}

```

Результат работы программы:

```
Вызван конструктор по умолчанию
Вызван конструктор с параметрами
Вызван конструктор копирования
Вызван конструктор копирования
0
0
0
Вызван деструктор

158
22
50

Вызван конструктор с параметрами
11

159
23
75.99

Вызван деструктор
Вызван деструктор
Вызван деструктор
```

Вывод: Я научилась определять пользовательский класс, реализовывать конструкторы и деструктор. Определила указатель на объект класса (В функции main создала указатель при помощи конструктора с параметрами, при помощи метода GetNumber вывела один из атрибутов).