

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский государственный технический университет”  
Кафедра ИИТ

Лабораторная работа № 4  
Объектное-ориентированное программирование и проектирование  
**" КЛАССЫ И ОБЪЕКТЫ В C++ "**

Выполнил:  
Студент 2-го курса  
Группы АС-53  
Бранчук Д.В.  
Проверила:  
Давидюк Ю.И.

Брест 2020

**Цель.** Получить практические навыки реализации классов в C++

**Постановка задачи:**

1. Определить пользовательский класс АДРЕС с полями  
имя-`char*` улица – `char*` номер дома – `int`

Написать конструктор без аргументов, с аргументами, конструктор копирования, определить в классе деструктор, определить геттеры и сеттеры, определить указатель на экземпляр класса.

```
#pragma once

using namespace std;

class Address
{
private:
    string name;//поле для имени
    string street;//поле для улицы
    int number;//поле для номера дома
public:
    //всё что ниже - прототипы чего-либо
    Address(string name, string street, int number);//конструктора с параметрами
    Address();//конструктора без параметров
    Address(const Address &item);//конструктора копирования
    ~Address();//деструктора

    void SetName(string val);//сеттера имени
    string GetName();//геттера имени

    void SetStreet(string val);//сеттера улицы
    string GetStreet();//геттера улицы
    void SetNumber(int val);//сеттера номера дома
    int GetNumber();//геттера номера дома
};
```

2.

```
#include "pch.h"
#include <string>
#include <iostream>
#include "Address.h"
```

```
using namespace std;
```

```
Address::Address()//конструктор без параметров
```

```
{
    cout << "Constructor bez parametrov bil vizvan. Object: " << this <<
endl; //сообщение о вызове конструктора
    name = "null";
    street = "null";
    number = 0;
}
```

```
Address::Address(string name, string street, int number)//конструктор с параметрами
```

```

{
    cout << "Constructor bez parametrov bil vizvan. Object: " << this << endl;
    this->name = name;
    this->street = street;
    this->number = number;
}
Address::Address(const Address &address)//конструктор копирования
{
    cout << "Constructor copirovaniya bil vizvan. Object: " << this << endl;
    this->name = address.name;
    this->street = address.street;
    this->number = address.number;
}
Address::~Address()//деструктор
{
    cout << "Destructor bil vizvan. Object: " << this << endl;
}
void Address::SetName(string val)//сеттер имени
{
    name = val;
}
string Address::GetName()//геттер имени
{
    return name;
}

void Address::SetStreet(string val)
{
    street = val;
}
string Address::GetStreet()
{
    return street;
}

void Address::SetNumber(int val)
{
    number = val;
}
int Address::GetNumber()
{
    return number;
}

```

3.

```
Address::Address()//конструктор без параметров
{
    cout << "Constructor bez parametrov bil vizvan. Object: " << this << endl;//сообщение о вызове конструктора
    name = "null";
    street = "null";
    number = 0;
}

Address::Address(string name, string street, int number)//конструктор с параметрами
{
    cout << "Constructor bez parametrov bil vizvan. Object: " << this << endl;
    this->name = name;
    this->street = street;
    this->number = number;
}

Address::Address(const Address &address)//конструктор копирования
{
    cout << "Constructor copirovaniya bil vizvan. Object: " << this << endl;
    this->name = address.name;
    this->street = address.street;
    this->number = address.number;
}

Address::~Address()//деструктор
{
    cout << "Destructor bil vizvan. Object: " << this << endl;
}
```

4.

```
ptr = &address2;
cout << ptr->GetName() << endl;
ptr = &address3;
cout << ptr->GetName() << endl;
ptr->SetName("changed");
cout << address3.GetName() << endl;
```

5.

```
int main()
{
    Address address; //вызов конструктора без параметров
    Address address2("Dimasik", "Sakharova", 47);//вызов конструктора с параметрами
    Address address3(address);//вызов конструктора копирования
    Address *ptr =new Address;//вызов конструктора без параметров
    address2.SetName("Ananasik");
    ptr = &address2;
    cout << ptr->GetName() << endl;
    ptr = &address3;
    cout << ptr->GetName() << endl;
    ptr->SetName("changed");
    cout << address3.GetName() << endl;
    return 0;
    //Вызов деструктора
    //Вызов деструктора
    //Вызов деструктора
    //Вызов деструктора
}
```

**Вывод:** Мы изучили классы и способы их обработки в языке C++