

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  
Вожейко М.В.  
Проверила:  
Давидюк Ю. И.

Брест, 2020

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

## 1. Постановка задачи (Вариант 6)

### 6. СЛУЖАЩИЙ

имя – char\*

возраст – int

рабочий стаж – int

Конструкторы:

- Без параметров
- С 3 параметрами
- Копирования

Функции:

- Вывода
- Аллокации памяти

## 2. Определение пользовательского класса с комментариями.

```
#pragma once
```

```
class employer {
private:
    char* name;
    int age;
    int stage;
    void allocName(const char* _name);

public:
    employer();
    employer(const char* _name, int _age, int _stage);
    employer(const employer&);
    ~employer();
    void print();
};
```

### 1. };Реализация конструкторов и деструктора и других функций

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <iostream>
```

```
#include "employer.h"
```

```
void employer::allocName(const char* _name) { // Выделяем память
    name = new char[strlen(_name) + 1];
    strcpy(name, _name);
}
```

```
employer::employer() { // Конструктор без параметров
    allocName("Untitled");
    std::cout << "Create class " << name << " without constructor params." << std::endl;
}
```

```
employer::employer(const char* _name, int _age, int _stage) { // Конструктор с 3 параметрами
    allocName(_name);
    age = _age;
    stage = _stage;
    std::cout << "Create class " << name << " with 3 constructor params." << std::endl;
}
```

```

employer::employer(const employer& _empl) { // Конструктор копирования
    allocName(_empl.name);
    age = _empl.age;
    stage = _empl.stage;
    std::cout << "Create class " << name << " with copy constructor." << std::endl;
}

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

void employer::print() { // Вывод
    std::cout << "\nName: " << name <<
        "\nAge: " << age <<
        "\nStage: " << stage << "\n" << std::endl;
}

```

**1. } }Фрагмент программы, показывающий использование указателя на объект с объяснением.**

`employer* employer1 = new employer("Matt", 32, 5);` - создаем объект класса `employer` с именем `employer` параметрами «Matt», 32, 5 и получаем на него указатель.

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

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include "employer.h"
int main() {
    employer* employer1 = new employer("Matt", 32, 5);
    employer employer2;
    employer employer3(*employer1);

    employer1->print();
    employer2.print();
    employer3.print();

    delete(employer1);

    return 0;
}

```

**1. Вывод:**

Получила практические навыки реализации классов на C++.