

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №4
по ООПиП
«Классы и объекты в С++»

Выполнил:
Студент 2 курса
Группы АС-53
Брашевец Д. Г.
Проверил:
Давидюк Ю.И..

Брест 2020

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

Вариант №4

- 1) Новая последовательность сначала должна содержать все элементы исходной с четными индексами, затем - все остальные (с сохранением исходного относительного порядка).

2) Текст программы

```
#define _CRT_SECURE_NO_WARNINGS
#include <cstring>
#include <iostream>

using namespace std;

class Job {
private:
    char* name;
    char* boss;
    int workers;
    void allocateChars(const char* _name);
    void allocateChars_boss(const char* _boss);

public:
    Job();
    Job(const char* _name, const char* _boss, int _workers);
    Job(const Job&);
    ~Job();
    void print();
};

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

void Job::allocateChars_boss(const char* _boss) { // Выделяем память
    boss = new char[strlen(_boss) + 1];
    strcpy(boss, _boss);
}

Job::Job() { // Конструктор без параметров
    allocateChars("Unnamed");
    allocateChars_boss("Unnamed");
    cout << "Create class Job without constructor params." << endl;
}

Job::Job(const char* _name, const char* _boss, int _workers) { // Конструктор с 3 параметрами
    allocateChars(_name);
    allocateChars_boss(_boss);
    workers = _workers;
    cout << "Create class Job with 3 constructor params." << endl;
}

Job::Job(const Job& _Job) { // Конструктор копирования
    allocateChars(_Job.name);
    allocateChars_boss(_Job.boss);
    workers = _Job.workers;
    cout << "Create class Job with copy constructor." << endl;
}

Job::~Job() { // Деструктор
    cout << "Call Job class \'" << name << "\' destructor." << endl;
}
```

```

        cout << "Call Job class \"" << boss << "\" destructor." << endl;
        delete[] name;
        delete[] boss;
    }

    void Job::print() { // Вывод
        cout << "\nName: " << name << "\nBoss: " << boss << "\nWorkers: ";
        if (workers) cout << 25; else cout << 10;
        cout << "\n" << endl;
    }

    int main() {
        Job* Job1 = new Job("Dmitriy", "Brashik", 0);
        Job Job2;
        Job Job3(*Job1);

        Job1->print();
        Job2.print();
        Job3.print();

        delete(Job1);

        return 0;
    }

```

Результат выполнения программы

```

Create class Job with 3 constructor params.
Create class Job without constructor params.
Create class Job with copy constructor.

Name: Dmitriy
Boss: Brashik
Workers: 10

Name: Unnamed
Boss: Unnamed
Workers: 25

Name: Dmitriy
Boss: Brashik
Workers: 10

Call Job class "Dmitriy" destructor.
Call Job class "Brashik" destructor.
Call Job class "Dmitriy" destructor.
Call Job class "Brashik" destructor.
Call Job class "Unnamed" destructor.
Call Job class "Unnamed" destructor.

```