

Министерство науки и высшего образования РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курский государственный университет»

Кафедра программного  
обеспечения и администрирования  
информационных систем

Направление подготовки  
математическое обеспечение и  
администрирование  
информационных систем

Форма обучения очная

**Отчет**  
**по лабораторной работе №1**  
**«Объекты и классы»**

Выполнил:

студент группы 213

Тихонов Е.Е.

Проверил:

старший преподаватель кафедры ПОиАИС

Ураева Е.Е.

Курск, 2021

**Цель работы:** изучить особенности написания программ на языке C++ с использованием объектов и классов.

### ***Задание***

**Задача 1.** Разработать класс “лабораторная работа” и программу, в которой создаются и разрушаются объекты данного класса (при использовании всех типов конструкторов). В программе предусмотреть использование массива объектов класса. Продемонстрировать в программе примеры явного и неявного вызова деструкторов: для статических и динамических объектов класса.

Требования к классу:

- наличие как минимум трех полей класса разных типов данных, один из которых должен быть перечисляемым;
- наличие закрытой (private) и общедоступной (public) частей;
- класс должен иметь три типа конструкторов, определенных программистом: конструктор без параметров, конструктор с параметрами и конструктор копии;
- хотя бы в одном из конструкторов необходимо использовать список инициализаторов;
- хотя бы в одном из конструкторов необходимо использовать указатель this в явном виде;
- вызов каждого из конструкторов должен сопровождаться выдачей соответствующего сообщения;
- класс должен иметь деструктор.

### ***Разработка алгоритма***

#### ***Задача 1***

Входные данные: нет.

Выходные данные: логи создания, копирования и уничтожения объектов в консоли.

### ***Текст программы***

*Текст программы для решения задачи 1*

```
#include <iostream>
#include <string>

using namespace std;

enum Level { easy, normal, hard };
string LvlToStr(Level lvl)
{
    switch(lvl)
    {
        case 0: return "easy";
        case 1: return "normal";
        case 2: return "hard";
    }
}

class Task
{
    private:
        int Num;
        string Text;

    public:
        Task(int Num, string Text)
        {
            this->Num = Num;
            this->Text = string(Text);
        }
};
```

```

        cout << "Create Task obj " + to_string(this->Num) << endl;
    }

    Task(const Task& CopyObj)
    {
        this->Num = CopyObj.Num;
        this->Text = string(CopyObj.Text);
        cout << "Copy Task obj " + to_string(this->Num) << endl;
    }

    ~Task()
    {
        cout << "Destruct Task obj " + to_string(this->Num) << endl;
    }

    public:
    void Print()
    {
        cout << " " + to_string(this->Num);
        cout << ". " + this->Text << endl;
    }
};

class LabWork
{
    private:
    int Num;
    string Name;
    Level Lvl;

    Task** Tasks;
    int TaskCount;

```

```

    public:
        LabWork(int Num, string Name, Level Lvl, Task** Tasks, int
TaskCount)
    {
        cout << "Create LabWork obj with params" << endl;

        this->Num = Num;
        this->Name = string(Name);
        this->Lvl = Lvl;
        CopyTasks(Tasks, TaskCount);
    }

        LabWork(): Num(1), Name("Some lab name"), Lvl(easy),
Tasks(nullptr), TaskCount(0)
    {
        cout << "Create default LabWork obj" << endl;
    }

        LabWork(const LabWork& CopyObj): Num(CopyObj.Num),
Name(CopyObj.Name), Lvl(CopyObj.Lvl)
    {
        cout << "Copy LabWork obj from another obj" << endl;
        CopyTasks(CopyObj.Tasks, CopyObj.TaskCount);
    }

~LabWork()
{
    if (Tasks!=nullptr)
    {
        cout << "Destruct LabWork obj " + to_string(this->Num) <<
endl;

        for (int i = 0; i < TaskCount; i++)
        {

```

```

        delete Tasks[i];
    }
    delete[] Tasks;

    Tasks = nullptr;
}

}

void Print()
{
    cout << endl;
    cout << "        Laboratory work " + to_string(this->Num) <<
endl;
    cout << "    Title: " + this->Name << endl;
    cout << "Level: " + LvlToStr(this->Lvl) << endl;
    cout << "Tasks: " << endl;

    for (int i = 0; i < TaskCount; i++)
    {
        this->Tasks[i]->Print();
    }
    cout << endl;
}

private:
void CopyTasks(Task** tasks, int tasksCount)
{
    this->Tasks = new Task * [tasksCount];
    for (int i = 0; i < tasksCount; i++)
    {
        Tasks[i] = new Task(*tasks[i]);
    }
}

```

```

        this->TaskCount = tasksCount;
    }
};

int main()
{
    int taskCount = 3;
    Task** tasks = new Task*[taskCount];
    for (int i = 0; i < taskCount; i++)
    {
        tasks[i] = new Task(i+1, "Some text for task");
        tasks[i]->Print();
    }

    LabWork lab1 = LabWork();
    lab1.Print();

    LabWork lab2 = LabWork(2, "Name of lab 2", normal, tasks,
taskCount);
    lab2.Print();

    LabWork lab3 = LabWork(lab2);
    lab3.Print();

    cout << "delete tasks"<<endl;
    for (int i = 0; i < taskCount; i++)
    {
        delete tasks[i];
    }
    delete[] tasks;
    cout<< endl;

    cout << "delete lab3 (copy of lab 2)" << endl;

```

```
lab3.~LabWork();  
cout << endl;  
}
```

### ***Тестирование программы***

Тестирование задачи 1 представлено на рисунках 1, 2.



```
Консоль отладки Microsoft Visual Studio
Create Task obj 1
  1. Some text for task
Create Task obj 2
  2. Some text for task
Create Task obj 3
  3. Some text for task
Create default LabWork obj
  Laboratory work 1
  Title: Some lab name
Level: easy
Tasks:

Create LabWork obj with params
Copy Task obj 1
Copy Task obj 2
Copy Task obj 3
  Laboratory work 2
  Title: Name of lab 2
Level: normal
Tasks:
  1. Some text for task
  2. Some text for task
  3. Some text for task

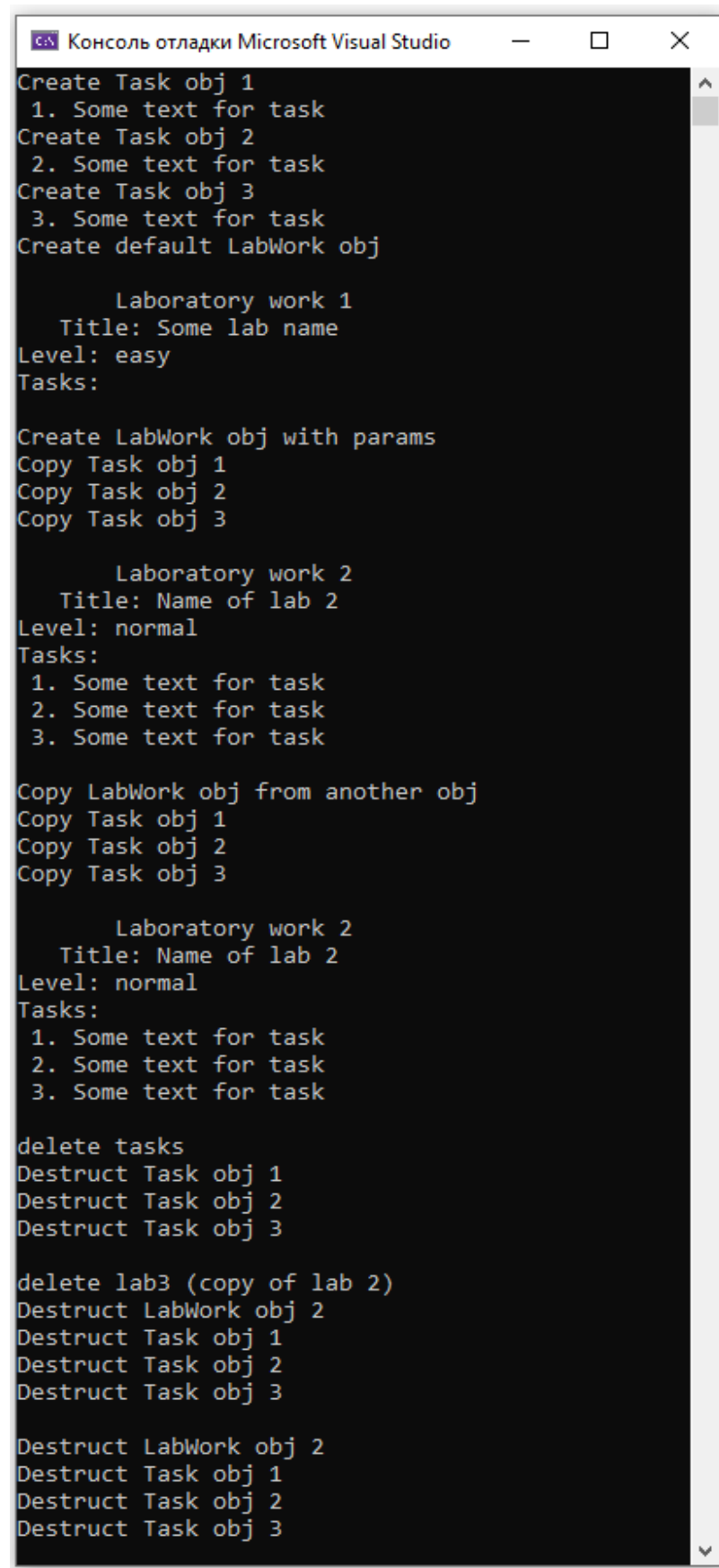
Copy LabWork obj from another obj
Copy Task obj 1
Copy Task obj 2
Copy Task obj 3
  Laboratory work 2
  Title: Name of lab 2
Level: normal
Tasks:
  1. Some text for task
  2. Some text for task
  3. Some text for task

delete tasks
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3

delete lab3 (copy of lab 2)
Destruct LabWork obj 2
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3

Destruct LabWork obj 2
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3
```

Рисунок 1 - Тест 1 задачи 1



```
Консоль отладки Microsoft Visual Studio

Create Task obj 1
  1. Some text for task
Create Task obj 2
  2. Some text for task
Create Task obj 3
  3. Some text for task
Create default LabWork obj

  Laboratory work 1
  Title: Some lab name
Level: easy
Tasks:

Create LabWork obj with params
Copy Task obj 1
Copy Task obj 2
Copy Task obj 3

  Laboratory work 2
  Title: Name of lab 2
Level: normal
Tasks:
  1. Some text for task
  2. Some text for task
  3. Some text for task

Copy LabWork obj from another obj
Copy Task obj 1
Copy Task obj 2
Copy Task obj 3

  Laboratory work 2
  Title: Name of lab 2
Level: normal
Tasks:
  1. Some text for task
  2. Some text for task
  3. Some text for task

delete tasks
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3

delete lab3 (copy of lab 2)
Destruct LabWork obj 2
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3

Destruct LabWork obj 2
Destruct Task obj 1
Destruct Task obj 2
Destruct Task obj 3
```

Рисунок 2 - Тест 2 задачи 1