

Автор: Татаренко А., КІТ-119а

Дата: 8 січня 2019

Лабораторна робота №2.

ПЕРЕВАНТАЖЕННЯ МЕТОДІВ

Тема роботи: Класи. Конструктори та деструктори. Перевантаження методів.

Мета роботи: отримати базові знання про класи, конструктори та деструктори. Дослідити механізм створення та видалення об'єктів.

1 Завдання до роботи

Індивідуальне завдання 19.

Поширити попередню лабораторну роботу таким чином:

- 1) в базовому класі необхідно додати: - мінімум одне поле типу `char*`; - конструктор за замовчуванням, копіювання та конструктор з аргументами; - деструктор;
- 2) у клас-список потрібно додати метод обходу масиву для виконання індивідуального завдання.

2 Розробка алгоритму розв'язання задачі.

2.1 Опис змінних

```
Arr stud_array; class Student; class Arr;
```

Класи, методи, функції, конструктори

3 Код програми

basic_class.h:

```
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtDBG.h>

#include <iostream>
#define E1 3

using std::cout;
using std::cin;
using std::endl;

class Student
{
private:

    int age;
    int number_stud;
    int middle_mark;
    const char *name;
    bool debt;
    int prog_d;

public:

    Student();

    Student(int a, int n, int m, const char* na, bool d, int pd);

    Student(const Student &other);

    int get_number_stud() const;

    int get_age() const;

    int get_middle_mark() const;

    const char* get_name();

    bool get_debt() const;

    int get_prog_d() const;

    ~Student()
    {
        cout << "Вызвался деструктор!" << endl;
    }
};
```

class_list.h:

```
#pragma once
#include "basic_class.h"

class Arr
{
private:

    Student* array_stud;
```

```

        int count = 1;
public:
    void create_array();
    void print_array() const;
    void delete_one();
    void add();
    void select() const;
    void fill_array();
    int count_plus();
    int count_minus();
    void print_one(Student) const;
    void delete_array();
    Student Construct(int);
    int prog_d_rand(bool);
    void find_debt();

};

```

menu.h:

```

#pragma once
#include "class_list.h"

void menu(Arr);

```

test.h:

```

#pragma once
#include "class_list.h"

bool test_count_plus(Arr);
bool test_count_minus(Arr);

```

basic_class.cpp:

```

#include "basic_class.h"

Student::Student() : age(0), number_stud(0), middle_mark(0), name("Name"), debt(0),
prog_d(0)
{
    cout << "Вызван стандартный конструктор!" << endl;
}

Student::Student(int a, int n, int m, const char* na, bool d, int pd) : age(a),
number_stud(n), middle_mark(m), name(na), debt(d), prog_d(pd)
{

```

```

        cout << "Вызван конструктор с параметрами!" << endl;
    }

    Student::Student(const Student &other) : age(other.age), number_stud(other.number_stud),
    middle_mark(other.middle_mark), name(other.name), debt(other.debt), prog_d(other.prog_d)
    {
        cout << "Вызван конструктор копирования!" << endl;
    }

    int Student::get_number_stud() const
    {
        return number_stud;
    }

    int Student::get_age() const
    {
        return age;
    }

    int Student::get_middle_mark() const
    {
        return middle_mark;
    }

    const char* Student::get_name()
    {
        return name;
    }

    bool Student::get_debt() const
    {
        return debt;
    }

    int Student::get_prog_d() const
    {
        return prog_d;
    }

```

class_list.cpp:

```

#include "class_list.h"

void Arr::create_array()
{
    array_stud = new Student[count];
}

void Arr::fill_array()
{
    for (int i = 0; i < count; i++)
    {
        array_stud[i] = Construct(i);
    }
}

Student Arr::Construct(int i)
{
    bool d = 0;

    if (i == 0)

```

```

{
    d = rand() % 2;
    Student st0(i + 17, i + 1, i + 10 / 2, "Dani", d, prog_d_rand(d));
    return st0;
}
else if (i == 1)
{
    d = rand() % 2;
    Student st1(i + 17, i + 1, i + 10 / 2, "Peter", d, prog_d_rand(d));
    return st1;
}
else if (i == 2)
{
    d = rand() % 2;
    Student st2(i + 17, i + 1, i + 10 / 2, "Donald", d, prog_d_rand(d));
    return st2;
}

d = rand() % 2;
Student st(i + 17, i + 1, i + 10 / 2, "Edgar", d, prog_d_rand(d));
return st;
}

void Arr::print_array() const
{
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    for (int i = 0; i < count; i++)
    {
        cout << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст: " <<
array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark() <<
"\t\tИмя: " << array_stud[i].get_name() << endl;
    }
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
}

void Arr::delete_one()
{
    int input, i_arr = 0;
    cout << "Введите номер: ";
    cin >> input;

    for (int i = 0; i < El; i++)
        if (input == array_stud[i].get_number_stud())
        {
            count_minus();

            Student* array_stud_new = new Student[count];

            for (int i = 0; i < count + 1; i++)
            {
                if (input == array_stud[i].get_number_stud())
                    continue;
                array_stud_new[i_arr++] = array_stud[i];
            }

            delete[] array_stud;

            array_stud = new Student[count];
        }
}

```

```

        for (int i = 0; i < count; i++)
        {
            array_stud[i] = array_stud_new[i];
        }

        delete[] array_stud_new;

        print_array();

        return;
    }
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    cout << "Этого студента не существует!" << endl;
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
}

void Arr::add()
{
    count_plus();

    Student* array_stud_new = new Student[count];

    int i = 0;

    for (i; i < count - 1; i++)
    {
        array_stud_new[i] = array_stud[i];
    }

    array_stud_new[count - 1] = Construct(i);

    delete[] array_stud;

    array_stud = new Student[count];

    for (i = 0; i < count; i++)
    {
        array_stud[i] = array_stud_new[i];
    }

    delete[] array_stud_new;

    print_array();
}

int Arr::count_plus()
{
    count = count + 1;

    return count;
}

int Arr::count_minus()
{
    count = count - 1;

    return count;
}

```

```

}

void Arr::select() const
{
    int input = 0;
    cout << "Введите номер: ";
    cin >> input;

    for (int i = 0; i < count; i++)
    {
        if (input == array_stud[i].get_number_stud())
        {
            print_one(array_stud[i]);
            return;
        }
    }
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    cout << "Этого студента не существует!" << endl;
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
}

void Arr::print_one(Student stud) const
{
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    cout << "Номер: " << stud.get_number_stud() << "\tВозраст: " << stud.get_age() <<
"\t\tСредний балл: " << stud.get_middle_mark() << "\t\tИмя: " << stud.get_name() << endl;
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
}

void Arr::delete_array()
{
    delete[] array_stud;
}

int Arr::prog_d_rand(bool d)
{
    int pd = 0;

    if (d == 1)
        pd = rand() % 50;

    return pd;
}

void Arr::find_debt()
{
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    for(int i = 0; i < count; i++)

```

```

        if (array_stud[i].get_debt() == 1)
        {
            cout << "Номер: " << array_stud[i].get_number_stud() << "\tВозраст: " << array_stud[i].get_age() << "\t\tСредний балл: " << array_stud[i].get_middle_mark() << "\t\tИмя: " << array_stud[i].get_name() << "\t\tДолг: Есть" << "\t\tДолг по программированию: " << array_stud[i].get_prog_d() << "%" << endl;
        }
        ///////////////////////////////////
        for (int i = 0; i < 50; i++)
            cout << "-";
        cout << endl;
    }
}

```

main.cpp:

```

/**
 * @mainpage
 * <b> Лабораторна робота № 2. <br/> ПЕРЕВАНТАЖЕННЯ МЕТОДІВ </b>
 * <br/><b><i>Мета роботи:</i></b>: отримати базові знання про класи, конструктори та деструктори. Дослідити механізм створення та видалення об'єктів. <br/>
 * <b><i>Індивідуальне завдання 19:</i></b>
 *     Поширити попередню лабораторну роботу. <br/>
 * <br/>
 * <br/> <b> Arr p; class Student; class Arr; </b> Класи, методи, функції, конструктори </b>
 * @author Tatarenko A.
 * @date 18-mar-2020
 * @version 1.0
 */

#include "menu.h"

int main()
{
    setlocale(LC_ALL, "ru");

    Arr stud_array;

    int input = 0;

    cout << "Введите 1 чтобы создать массив, 2 чтобы выйти: ";
    cin >> input;

    if (input == 1)
    {
        stud_array.create_array();
        cout << "Список созданный с помощью конструктора:" << endl;
        stud_array.print_array();

        cout << "Рандомное заполнение списка:" << endl;
        stud_array.fill_array();
        stud_array.print_array();

        menu(stud_array);

        stud_array.delete_array();
    }

    ///////////////////////////////////
    int l = _CrtDumpMemoryLeaks(); // Контроль витоку пам'яті
    if (l == 0)
        cout << "Утечки памяти не обнаружено!" << endl;
}

```



```

        else
            cout << "Обнаружена утечка памяти!" << endl;

        system("PAUSE");
        return 0;
    }

```

menu.cpp:

```

#include "menu.h"

void menu(Arr stud_array)
{
    int input = 0;

    cout << "1) Добавить объект в конец;" << endl << "2) Удалить объект;" << endl <<
    "3) Выбрать объект;" << endl << "4) Вывести все объекты;" << endl << "5) Поиск должников"
    << endl << "6) Выйти;" << endl << "Ваше действие: ";
    cin >> input;

    switch (input)
    {
        case 1:
            stud_array.add();
            break;
        case 2:
            stud_array.delete_one();
            break;
        case 3:
            stud_array.select();
            break;
        case 4:
            stud_array.print_array();
            break;
        case 5:
            stud_array.find_debt();
            break;
        case 6:
            return;
        default:
            break;
    }

    menu(stud_array);
}

```

test.cpp:

```

#include "test.h"

int main()
{
    Arr stud_array;

    if (test_count_plus(stud_array))
        cout << "Function count_plus is right!" << endl;
    else
        cout << "Function count_plus is not right!" << endl;

    if (test_count_minus(stud_array))
        cout << "Function count_minus is right!" << endl;
    else

```

```

        cout << "Function count_minus is not right!" << endl;

        system("PAUSE");
    }

bool test_count_plus(Arr p)
{
    int exp = E1 + 1;
    int cnt = p.count_plus();

    if (cnt == exp)
        return 1;
    else
        return 0;
}

bool test_count_minus(Arr p)
{
    int exp = E1 - 1;
    int cnt = p.count_minus();

    if (cnt == exp)
        return 1;
    else
        return 0;
}

```

4 Результати тестування

```

E:\KhPI\lab 2\lab01\x64\Debug\lab02.exe
Введите 1 чтобы создать массив, 2 чтобы выйти: 1
Вызван стандартный конструктор!
Вызван стандартный конструктор!
Вызван стандартный конструктор!
Список созданный с помощью конструктора:
-----
Номер: 0      Возраст: 0      Средний балл: 0      Имя: Name
Номер: 0      Возраст: 0      Средний балл: 0      Имя: Name
Номер: 0      Возраст: 0      Средний балл: 0      Имя: Name
-----
Рандомное заполнение списка:
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
Вызван конструктор с параметрами!
Вызван конструктор копирования!
Вызвался деструктор!
Вызвался деструктор!
-----
Номер: 1      Возраст: 17      Средний балл: 5      Имя: Dani
Номер: 2      Возраст: 18      Средний балл: 6      Имя: Peter
Номер: 3      Возраст: 19      Средний балл: 7      Имя: Donald
-----
1) Добавить объект в конец;
2) Удалить объект;
3) Выбрать объект;
4) Вывести все объекты;
5) Поиск должников
6) Выйти;
Ваше действие:

```

5 Опис результатів

Програма виводить список студентів. Виконує видалення, додавання, вибір об'єкта, виводить інформацію про заборгованість.

Висновок:

Протягом цієї роботи було отримано базові знання про конструктори за замовчуванням, конструктори з параметрами, конструктори копіювання та деструктори.