

Автор: Татаренко А., КІТ-119а

Дата: 17 червня 2020

Лабораторна робота №4.

## **РЕГУЛЯРНІ ВИРАЗИ**

Тема. Регулярні вирази.

Мета – отримати знання про базові регулярні вирази та досвід роботи із застосування їх на практиці.

### **1 Завдання до роботи**

#### **Індивідуальне завдання 19.**

Поширити попередню лабораторну роботу таким чином:

- при введенні інформації про базовий клас (нема різниці, чи з клавіатури, чи з файлу), організувати перевірку відповідності таким 28 критеріям з використанням регулярних виразів:
- можна вводити тільки кириличні символи, латинські символи, цифри, пропуски, розділові знаки;
- не повинно бути пропусків та розділових знаків, які повторюються;
- перше слово не повинно починатися з маленького символу;
- у клас-список додати метод, що виводить на екран список усіх об'єктів, які мають одне або більше полів з щонайменше двома словами (перевірку організувати за допомогою регулярних виразів).

### **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>
#include <fstream>
#include <string>
#include <sstream>
#include <regex>

#define E1 3

using std::cout;
using std::cin;
using std::endl;
using std::string;
using std::stringstream;
using std::ofstream;
using std::ifstream;
using std::regex;

class Student
{
private:
    int age;
    int number_stud;
    int middle_mark;
    string name;
    bool debt;
    int prog_d;

public:
    Student();

    Student(int a, int n, int m, string na, bool d, int pd);

    Student(const Student &other);

    int get_number_stud() const;

    int get_age() const;

    int get_middle_mark() const;

    string get_name() const;

    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(int) const;

    void delete_one();

    void add();

    string 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();

    void in_f();

    void from_f(int);

    int get_count() const;

    int str_in_file(string) const;

    void set_count(int);

    void regex_task();

};
```

## 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()
{
    int c = get_count();

    array_stud = new Student[c];
}

void Arr::fill_array()
{
    int c = get_count();

    for (int i = 0; i < c; 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 Tramp", d, prog_d_rand(d));
        return st2;
    }
    else if (i == 3)
    {
        d = rand() % 2;
        Student st2(i + 17, i + 1, i + 10 / 2, "Vladimir", 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(int c) const
{
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////////
    for (int i = 0; i < c; 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 < E1; i++)
        if (input == array_stud[i].get_number_stud())
        {
            count_minus();
            int c = get_count();

            Student* array_stud_new = new Student[c];

            for (int i = 0; i < c + 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[c];

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

            delete[] array_stud_new;

            print_array(c);

            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();
    int c = get_count();

    Student* array_stud_new = new Student[c];

```

```

    int i = 0;

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

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

    delete[] array_stud;

    array_stud = new Student[c];

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

    delete[] array_stud_new;

    print_array(c);
}

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

    return count;
}

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

    return count;
}

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

    for (int i = 0; i < c; i++)
    {
        if (input == array_stud[i].get_number_stud())
        {
            string info;
            stringstream ss;
            ss << "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;
            info = ss.str();

            return info;
        }
    }

    //////////////////////////////////
    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;
    }

    void Arr::in_f()
    {
        ofstream file;
        file.open("of.txt", ofstream::app);

        if (!file.is_open())
        {
            cout << "Файл не открыт!" << endl;
        }
    }

```



```

        return;
    }

    string info;
    stringstream ss;
    int c = get_count();

    for (int i = 0; i < c; i++)
    {
        ss << "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 < 80; i++)
        ss << "-";
    ss << endl;

    info = ss.str();
    file << info;

    file.close();

    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
    cout << "Файл записан успешно!" << endl;
    ////////////
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
    ////////////
}

void Arr::from_f(int size)
{
    int i = 0;

    string s;
    regex varEn("([\\d]* [\\d]* [\\d]* [A-Z]+[\\w,.;:-]* [0|1] [\\d]*)");

    ifstream file;
    file.open("if.txt");

    if (!file.is_open())
    {
        cout << "Файл не открыт!" << endl;
        return;
    }

    delete[] array_stud;
    array_stud = new Student[size];

    for (int i = 0; i < size; i++)
    {
        getline(file, s);
        if (regex_match(s, varEn))
        {
            std::istringstream iss(s);
            int age;
            int number;
            int middle_mark;

```

```

        string name;
        bool debt;
        int prog_d;

        iss >> age;
        iss >> number;
        iss >> middle_mark;
        iss >> name;
        iss >> debt;
        iss >> prog_d;

        Student file_el(age, number, middle_mark, name, debt, prog_d);
        array_stud[i] = file_el;
    }
}

file.close();

cout << "Чтение с файла успешно!" << endl << endl;

set_count(size);

print_array(size);
}

int Arr::get_count() const
{
    return count;
}

int Arr::str_in_file(string fileName) const
{
    int size = 0;
    int c = get_count();
    string line;
    ifstream file(fileName);

    if (!file.is_open() || c == 0)
    {
        cout << "There is no such file" << endl << endl;
        return 0;
    }

    while (getline(file, line))
    {
        size++;
    }
    file.close();

    return size;
}

void Arr::set_count(int c)
{
    count = c;
}

void Arr::regex_task()
{
    for (int i = 0; i < 50; i++)
        cout << "-";
    cout << endl;
}

```

```

//////////
regex regular("(^[A-ZÀ-ß]+[\\wÀ-ßà-ÿ,.;:-]* [\\wÀ-ßà-ÿ,.;:-]+)");
int listSize = get_count();

for (int i = 0; i < listSize; i++)
{
    if (regex_match(array_stud[i].get_name(), regular))
        print_one(array_stud[i]);

}

cout << endl;
//////////
for (int i = 0; i < 50; i++)
    cout << "-";
cout << endl;
}

```

## main.cpp:

```

/**
 * @mainpage
 * <b> Лабораторна робота № 4. <br/> РЕГУЛЯРНІ ВИРАЗИ </b>
 * <br/><b><i>Мета роботи:</i></b>: отримати знання про базові регулярні вирази та
досвід роботи із застосування їх на практиці. <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,
потoki </b>
 * @author Tatarenko A.
 * @date 29-may-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(stud_array.get_count());

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

        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(Array 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) Запись в файл;" << endl << "7) Читать с файла;" << endl <<
    "8) Выйти;" << endl << "Ваше действие: ";
    cin >> input;

    switch (input)
    {
    case 1:
        stud_array.add();
        break;
    case 2:
        stud_array.delete_one();
        break;
    case 3:
        cout << stud_array.select();
        break;
    case 4:
        stud_array.print_array(stud_array.get_count());
        break;
    case 5:
        stud_array.find_debt();
        break;
    case 6:
        stud_array.in_f();
        break;
    case 7:
        stud_array.from_f(stud_array.str_in_file("if.txt"));
        break;
    case 8:
        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

```
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 Tramp
-----
1) Добавить объект в конец;
2) Удалить объект;
3) Выбрать объект;
4) Вывести все объекты;
5) Поиск должников;
6) Запись в файл;
7) Читать с файла;
8) REGEX TASK;
9) Выйти;
Ваше действие:
```

## 5 Опис результатів

Програма виводить список студентів. Виконує видалення, додавання, вибір об'єкта, виводить інформацію про заборгованість. А також виконує запис і читання з файлу, та перевіряє читання через базові регулярні вирази.

Висновок:

Протягом цієї роботи було отримано базові знання про базові регулярні вирази.