НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №2*

*Тема: «*Класи. Конструктори та деструктори. Перевантаження методів.*»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** Отримати базові знання про класи, конструктори та деструктори. Дослідити механізм створення та видалення об’єктів.

***Індивідуальне завдання***

Варіант 6.

6 Самостійні роботи студентів Розрахунково-графічне завдання

Поширити попередню лабораторну роботу наступним чином:

− в базовому класі необхідно додати:

− мінімум одне поле типу char\*;

− конструктор за замовчуванням, копіювання та конструктор з аргументами;

− деструктор;

-зробити індивілуальне завдання

**Текст програми**

prototaype.h

#define kon clock()/60000//хвилини

#define kon2 (clock()/1000)%60//секунди

#include <ctime>

#include "stdlib.h"

#include "string.h"

#include "stdio.h"

#include "locale.h"

#include <crtdbg.h>

#include <fstream>

#include <string>

#include "time.h"

#include <random>

#include <iostream>

#define \_CRTDBG\_MAP\_ALLOC

#include <stdlib.h>

#include "crtdbg.h"

#include <iostream>

#include <Windows.h>

#define DDEBUG

using namespace std;

struct RGZ;

class SR;

void Init(RGZ\*&);

int sche(int&, char[]);

int VSTUProzmova(int);

void Riad\_FL(RGZ\*);

void dop\_INFO(RGZ\*);

int vstup\_info(int);

SR.h

#pragma once

#include "prototaype.h"//бібліотека та прототипи

class SR

{

private:

string predmet;

string tema;

int bal;

int zavdanj;

int termin;

int cathedra;

public:

SR()//конструктор

{

predmet = "NO name";

tema = "No name";

bal = NULL;

zavdanj = NULL;

termin = NULL;

cathedra = NULL;

cout << "\nВизвано конструктор за впровадженням"<<endl;

}

SR(const SR& obg) ://конструктор копіювання

predmet(obg.predmet), tema(obg.tema), cathedra(obg.cathedra), termin(obg.termin), zavdanj(obg.zavdanj), bal(obg.bal) {

cout << "\nВизвано конструктор копiювання" << endl;

}

SR(string predmet,string tema,int bal,int zavdanj,int termin,int cathedra)//конструктор з аргуменьами

{

this->predmet= predmet;

this->tema= tema;

this->bal=bal;

this->zavdanj=zavdanj;

this->termin=termin;

this->cathedra=cathedra;

cout << "\nКонструктор параметрами" << endl;

}

~SR()//деструктор

{

cout << "\nВизвано деструктор " << endl;

}

void printTableHead();

void Print\_run(RGZ\* p);//1 відобразити елементи

void Print\_el(RGZ\* p);

void add\_el(int ,string, string, int, int, int, RGZ\* p);//додати елемент

void add\_el(RGZ\* p);

void curriculum(RGZ\*);//навчальний план

void error( int&,int&);

void checking\_the\_constructor();//перевірка конструктора

void delet\_el(RGZ\* p);//удалить елемент

void dead\_el(RGZ\* p);//удалить список

string Get\_predmet() { return predmet; }

string Get\_teme() { return tema; }

int Get\_bal() { return bal; }

int Get\_zavdanj() { return zavdanj; }

int Get\_termin() { return termin; }

int Get\_cathedra() { return cathedra; }

void Set\_predmet(string value1) { this->predmet = value1; }

void Set\_teme(string value1) { this->tema = value1; }

void Set\_bal(int value) { this->bal = value; }

void Set\_zavdanj(int value) { this->zavdanj = value; }

void Set\_termi(int value) { this->termin = value; }

void Set\_cathedra(int value) { this->cathedra = value; }

};

struct RGZ {

SR rgz;

RGZ\* next;

};

Arbeit.cpp

#include "prototaype.h"

#include "SR.h"

int vstup\_info(int nov)

{

RGZ\* h=NULL ;

int a = 1, u = 0;

Init(h);

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

printf("Доброго дня користувач\n ");

printf("\nЗараз: ");

SetConsoleTextAttribute(hConsole, (WORD)(11));

printf(\_\_DATE\_\_);

printf("\nЧас початку роботи програми: ", SetConsoleTextAttribute(hConsole, (WORD)(10)));

printf(\_\_TIME\_\_, SetConsoleTextAttribute(hConsole, (WORD)(11)));

SetConsoleTextAttribute(hConsole, (WORD)(10));

printf("\n\n Почнемо роботу\n\n Ввести данi\n ");

while (a != 0)

{

SetConsoleTextAttribute(hConsole, (WORD)(12));

printf("\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ");

scanf("%i", &a);

switch (a)

{

case 1:SetConsoleTextAttribute(hConsole, (WORD)(10));

printf("Скiльки елементiв ви хочете додати? :");

scanf("%i", &a);

for (int i = 0; i < a; i++)

h->rgz.add\_el(h);

a = 0;

break;

case 2:SetConsoleTextAttribute(hConsole, (WORD)(10));

Riad\_FL(h), a = 0;

break;

}

}

a = main\_menu(a);

while (a)

{

switch (a)

{

case 1:

h->rgz.printTableHead();

h->rgz.Print\_run(h);

break;

case 2: dop\_INFO(h);

break;

case 3:h->rgz.delet\_el(h);

break;

case 4: h->rgz.Print\_el(h);

break;

case 5: {h->rgz.dead\_el(h);

printf("Успішно");

Init(h);

break; }

case 6:h->rgz.curriculum(h);

break;

case 7:nov = 0;

h->rgz.dead\_el(h);

SR();

return 0;

case 8:

h->rgz.dead\_el(h);

{system("cls");

nov = 1;

return 1;

}

case 9:

h->rgz.checking\_the\_constructor();

break;

}

char pit[] = "Бажаєте ще щось зробити?";

a = yes\_and\_no(a, pit);

if (a == 1)

a = main\_menu(a);

else a = 7;

}

}

void Init(RGZ\*& p) {

p = new RGZ;

p->next = NULL;

}

void SR::Print\_run(RGZ\* p)

{

RGZ\* pr = p->next;

int i = 1;

while (pr != NULL)

{

printf("|RGZ %-3d |", i);

cout << pr->rgz.Get\_predmet()<<"\t|"<< pr->rgz.Get\_teme()<<"\t";

printf("|%-18i|%-11i|%-11i|\n", pr->rgz.Get\_bal(), pr->rgz.Get\_termin(), pr->rgz.Get\_zavdanj());

printf("-----------------------------------------------------------------------------------\n");

i++;

pr = pr->next;

}

}

void SR::Print\_el(RGZ\* p)

{

RGZ\* pr = p;

int d;

printf("Який елемент хочете вiдобразити: ");

scanf("%i", &d);

int a = 1;

while (a)

{

a = 0;

if (d > 0)

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

if (pr->next != NULL)

pr = pr->next;

else

{

pr = p->next;

error(a, d);

}

}

else

{ pr = p->next;

error(a, d);

}

}

p->rgz.printTableHead();

printf("|RGZ %-3d |", d);

cout << pr->rgz.Get\_predmet() << "\t|" << pr->rgz.Get\_teme() << "\t";

printf("|%-18i|%-11i|%-11i|\n", pr->rgz.Get\_bal(), pr->rgz.Get\_termin(), pr->rgz.Get\_zavdanj());

printf("-----------------------------------------------------------------------------------\n");

}

void SR::add\_el(int cathedra,string predmet, string tema, int bal, int zavdanj, int termin, RGZ\* p)

{

RGZ\* add = new RGZ;

RGZ\* head = p;

add->next = NULL;

add->rgz.Set\_cathedra(cathedra);

add->rgz.Set\_predmet(predmet);

add->rgz.Set\_teme(tema);

add->rgz.Set\_bal(bal);

add->rgz.Set\_termi(termin);

add->rgz.Set\_zavdanj(zavdanj);

add->next = head->next;

head->next = add;

}

void SR::add\_el(RGZ\* p)

{

string predmet, tema;

int bal, termin, zadanj, cathedra;

cout << "Предмет :";

cin >> predmet;

cout << "Тема:";

cin >> tema;

cout << "Кiлькiсть балiв за завдання : ";

scanf("%i", &bal);

cout << "Перiод (у днях) : ";

scanf("%i", &termin);

cout << "Кiлькiсть завдань : ";

scanf("%i", &zadanj);

cout << "Факультет/Кафедра(лише номер): ";

scanf("%i", &cathedra);

p->rgz.add\_el(cathedra,predmet, tema, bal, termin, zadanj, p);

cout << "\nУспішно додано\n\n" << endl;

/////невідома точка зупунку на 2

}

void SR::delet\_el(RGZ\* p)

{

RGZ\* pr = p->next;

RGZ\* temp = p;

int d;

printf("Який елемент хочете видалити");

scanf("%i", &d);

int a = 1;

while (a)

{

a = 0;

if (d > 0)

{

for (int i = 1; i < d; i++)

if (pr->next != NULL)

{

temp = temp->next;

pr = pr->next;

}

else

{

temp = p;

pr = p->next;

printf("\nНажаль неможливо видалити неiснуючий елемент\n");

printf("Введiть iнше значення ");

scanf("%d", &d);

a = 1;

}

}else{

temp = p;

pr = p->next;

printf("\nНажаль неможливо видалити неiснуючий елемент\n");

printf("Введiть iнше значення ");

scanf("%d", &d);

a = 1;

}

}

{

temp->next = pr->next;

delete(pr);

}

}

void SR::dead\_el(RGZ\* p)//

{

RGZ\* pr = p->next;

if (p->next != 0)

{

while (p != 0)

{

p = pr->next;

delete(pr);

pr = p;

}

}

delete(p, pr, this);

}

void Riad\_FL(RGZ\* h)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

RGZ\* p = h;

string text, adresa;

int value, temp = 0;

ifstream fOpen;

printf("\nЯкий файл відкрити?\n\n 1. RGZ.txt\n 2. add\_RGZ.txt\n Ваш вiбiр:");

scanf("%i", &temp);

switch (temp)

{

case 1:adresa = "RGZ.txt";

break;

case 2:adresa = "add\_RGZ.txt";

break;

}

fOpen.open(adresa, ios\_base::in);

if (!fOpen)

{

SetConsoleTextAttribute(hConsole, (WORD)(5));

cout << "File is not open\n\n";

SetConsoleTextAttribute(hConsole, (WORD)(10));

return;

}SetConsoleTextAttribute(hConsole, (WORD)(10));

while (!fOpen.eof()) {

RGZ\* add=new RGZ;

add->next = p->next;

p->next = add;

fOpen >> text;

add->rgz.Set\_predmet(text);

fOpen >> text;

add->rgz.Set\_teme(text);

fOpen >> value;

add->rgz.Set\_bal(value);

fOpen >> value;

add->rgz.Set\_termi(value);

fOpen >> value;

add->rgz.Set\_zavdanj(value);

fOpen >> value;

add->rgz.Set\_cathedra(value);

p = p->next;

}

fOpen.close();

}

void dop\_INFO(RGZ\* h)

{

int a;

printf("\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ");

scanf("%i", &a);

switch (a)

{

case 1:

h->rgz.add\_el(h);

break;

case 2:

Riad\_FL(h), a = 0;

break;

}

}

void SR::curriculum(RGZ\*p)

{

RGZ\* pr=p->next;

int cu = 0, i = 0;

cout << "\n\nВведіть спецiальнiсть/кафедру(номер) :";

cin >> cu;

while (pr->next != 0)

{

if (cu == pr->rgz.Get\_cathedra())

i++;

pr = pr->next;

}

if (i > 0)

{

cout << "Вашому студенту у цьому навчальному році треба зробити: " << i << " РГЗ"<<endl;

cout << "Вам цiкаво з яких предметiв вам потрiбно робити \n 1:да\n 2:нет \n Ваш вибiр:";

int king;

cin >> king;

if (king == 1)

{

cout << "\n З таких предметiв :";

pr = p;

while (pr->next != 0)

{

if (cu == pr->rgz.Get\_cathedra())

if ((i++) % 4 != 0)

cout << " | " << pr->rgz.Get\_predmet();

else cout << "\n | " << pr->rgz.Get\_predmet();

pr = pr->next;

}

}

else

return;

}

else

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(11));

cout << "\n\nВiтаю у цьому роцi ваи непотрiбно робити РГЗ з жодних фредметiв!!!\n\n ";

SetConsoleTextAttribute(hConsole, (WORD)(10));

}

}

void SR::checking\_the\_constructor()

{

SR konst("Предмет", "тема", 3, 4, 4, 134);//для конструктора з параметрами

cout <<"елемент класу: "<< konst.Get\_teme() << endl;

SR konst2;//для конструктора

cout <<"елемент класу: "<< konst2.Get\_teme()<<endl;

SR konst3(konst2);//для конструктора копіювання

cout <<"елемент класу : " <<konst3.Get\_teme()<<endl;

}

Dialog.cpp

#include "SR.h"

#include "prototaype.h"

int main\_menu(int e)

{

printf(" \n 1.Вивести на екран список\n");

system("ping 1.1.1.1 -n 0.001 -w 0.001 > nul");

printf(" 2.Додати елемент в список\n");

printf(" 3.Видалити елемент з списку за номереом\n");

printf(" 4.Відобразити елемент списку за номером\n");

printf(" 5.Очистити список \n");

printf(" 6.Визначити кiлькiсть РГЗ, що виконує студент за весь\n перiод навчання в iнститутi у вiдповiдностi до\n навчального плану\n");

printf(" 7.Завершити роботу \n");

printf(" 8.Перезавантажити програму програму \n");

printf(" 9.Перевірити роботу конструкторів \n");

printf(" \nваш вибiр: ");

scanf\_s(" %d", &e);

return e;

}

int yes\_and\_no(int& t, char p[])

{

printf("\n\n%s\n ", p);

printf("так- 1\n ");

printf("нi- 2\n ");

printf("вибрана робота :");

scanf("%i", &t);

if (t == 2)

{

int oo = 12;

}

return t;

}

void SR::printTableHead()

{

printf("\n\n-----------------------------------------------------------------------------------\n");

printf("| | Предмет | Темa | Кiлькiсть балiв |Строк здачi| Кiлькiсть |\n");

printf("| | | |за виконане RGZ | RGZ | завдань |\n ");

printf("----------------------------------------------------------------------------------\n");

}

void SR::error( int& a,int& d )

{

printf("\nНажаль неможливо вiдобразити неiснуючий елемент\n");

printf("Введiть iнше значення ");

scanf("%d", &d);

a = 1;

}

Main.cpp

#include "prototaype.h"

#include "SR.h"

int main()

{

cout << "\t\t\t-----------------------------------------\n";

cout << "\t\t\t| | LABA2.0-2 | |\n";

cout << "\t\t\t-----------------------------------------\n";

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, (WORD)(10));

int poku = 1;

setlocale(LC\_ALL, "rus");

while (poku != 0)

poku=vstup\_info(poku);

int ram = \_CrtDumpMemoryLeaks();

cout << "\nВитоку памятi:" << ram << endl;

SetConsoleTextAttribute(hConsole, (WORD)(13));

cout << "\nРоботу завершено\nТривалiсть роботи:" << kon << " хв " << kon2 << " сек" << endl;

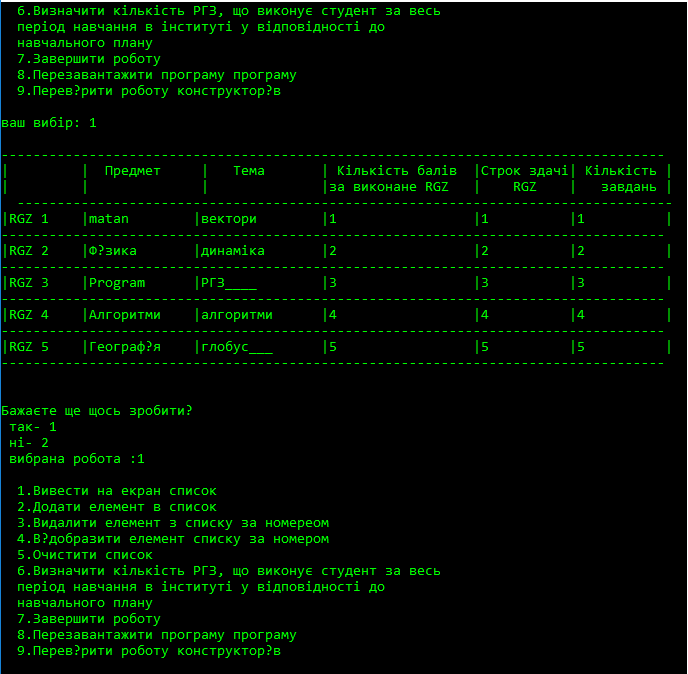
SetConsoleTextAttribute(hConsole, (WORD)(10));

return 0;

}

**Результати роботи програми:**

Рисунок 1 – Результати роботи програми:1 Рисунок 2 – Результати роботи програми:2



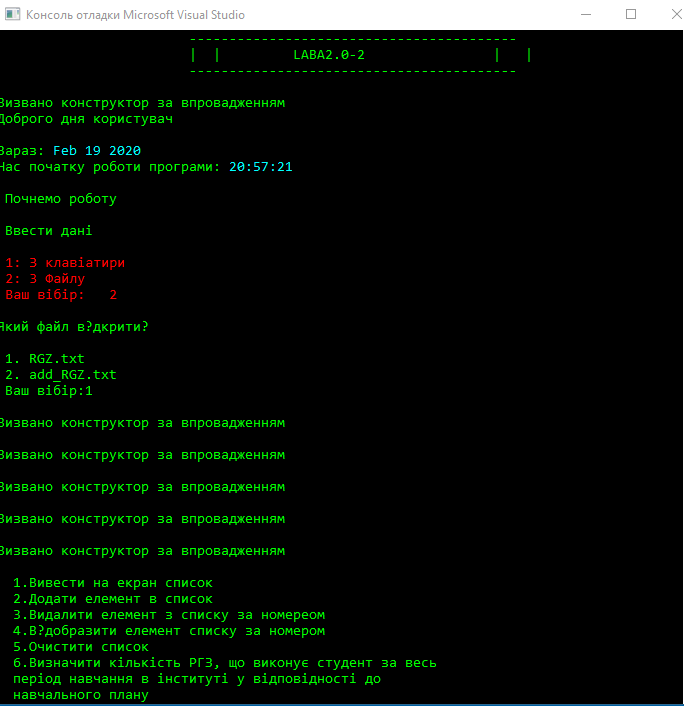


Рисунок 1 – Результати роботи контсруктора Рисунок 2 – Результати роботи програми:1

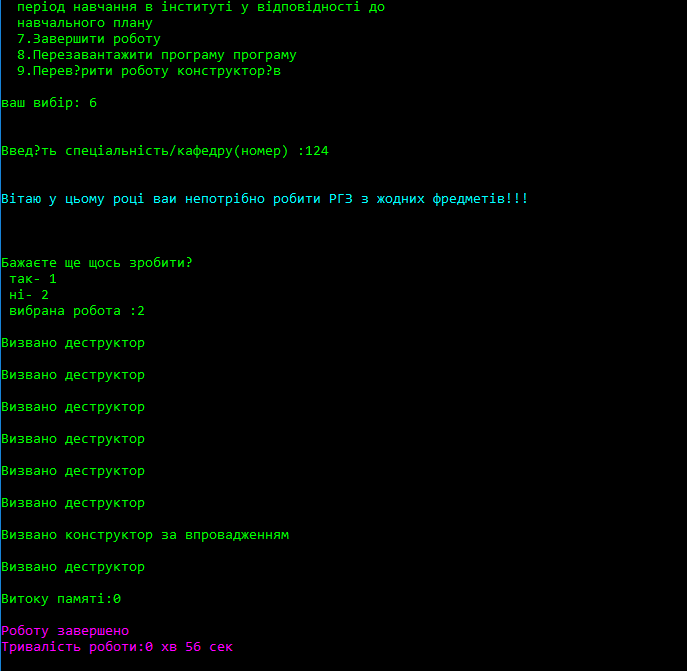
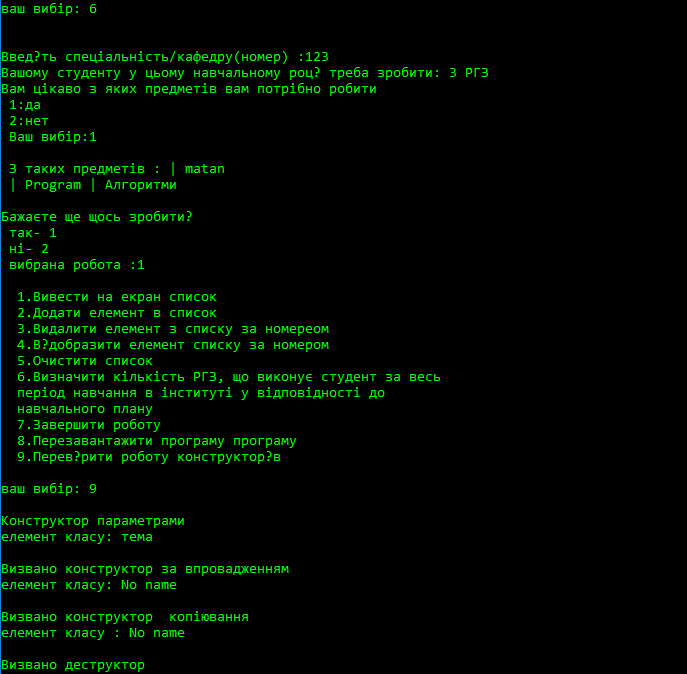


Рисунок 3 – Результати роботи програми:2 Рисунок 4 – Результати роботи деструктор

**Висновки:**

В результаті виконання лабораторної роботи отримано базові знання про класи, конструктори та деструктори.