НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №7*

*Тема:* Поліморфізм

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** *Отримати знання про парадигму ООП – поліморфізм.*

*Навчитися застосовувати отримані знання на практиці.*

Варіант 6.

6 Самостійні роботи студентів Розрахунково-графічне завдання

− додавання ще одного класу-спадкоємця до базового класу. Поля  
обрати самостійно;  
− базовий клас зробити абстрактним. Додати абстрактні поля;  
− розроблені класи-списки поєднуються до одного таким чином,  
щоб він міг працювати як з базовим класом, так і з його  
спадкоємцями. При цьому, серед полів класу-списку повен бути лише  
один масив, що містить усі типи класів ієрархії. Оновити методи, що  
працюють з цим масивом.

Тексt програми

prototaype.h

……….

SR.h

#pragma once

#include "prototaype.h"

#define ARRAY\_SIZE 255

class List\_SR;

class obkladynka ////агрегація

{

private:

boo\_SR cover;

public:

//конструктор

obkladynka() { cover = rand() % 2; }

~obkladynka()//деструктор

{

} void Set\_obkladynka(boo\_SR value) { this->cover = value; }

int Get\_obkladynka() { return cover; }

void kəvər() {

if (cover == true) cout << "\n\nВашому РГЗ потрiбна обкладинка обов'язково\n та не забутьте правильно написати Прiзвище та групу" << endl;

else cout << "Вашому РГЗ не потрiбна обкладинка взагалі" << endl;

}

};

class SR

{

public:

virtual SR\* clone() = 0;

virtual void Print\_run(List\_SR&)= 0;///< віртуальна функція для вивода списка

virtual void Print\_el(List\_SR&) = 0;///< віртуальна функція для вивода одного елементу спуска

virtual void add\_el(List\_SR&) = 0;///< віртуальна функція для додавання данних з клавіатури

virtual void Riad\_FL(List\_SR&) = 0;///< віртуальна функція для додавання данних із файлу

virtual int main\_menu(number) = 0;///< віртуальна функція для реалізацію головного меню для класу

virtual void printTableHead() = 0;///< віртуальна функція для реалізації вивода шапки таблиці

virtual int vstup\_info(number) = 0;///< віртуальна функція для реалізації початку програми

virtual void dop\_INFO(List\_SR&) = 0;

virtual void sortSTR(List\_SR&) = 0;

virtual int kriter(number n) = 0;

};

class RGZ : public SR

{

private:

string predmet;///<предмет

string tema;///<тема

int bal;///<кількість балів за завдання

int zavdanj;///<кількість завдань

int termin;///<строк здачи РГЗ у днях

int cathedra;///<кафедра

class cathedr//композиция

{

private:

number\_SR cathedra;

public:

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

int Get\_cathedra() { return cathedra; }

};

public :

RGZ()

{

predmet = "NO name";

tema = "No name";

bal = NULL;

zavdanj = NULL;

termin = NULL;

cathedra = NULL;

}

~RGZ() { ; }//деструктор

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; }

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 Print\_run(List\_SR&) override;

void Print\_el(List\_SR&) override;

void add\_el(List\_SR&) override;

void Riad\_FL(List\_SR&) override;

int main\_menu(number) override;

void printTableHead() override;

int vstup\_info(number) override;

void dop\_INFO(List\_SR&) override;

void sortSTR(List\_SR&) override;

SR\* clone() { return (SR\*)new RGZ(\*this);}

void delet\_el(List\_SR&);

void error(number& a, number& d);

int kriter(number n);

//void delete\_list(List\_SR&);

void curriculum(List\_SR&);

void Print\_two\_words\_in\_line(List\_SR&);

void save\_for\_Fl(List\_SR&);

void print\_saved\_for\_Fl();

cathedr cathedr;//композіция

obkladynka AG;//агрегація

};

class Scientific\_SR final : public RGZ

{

private:

string surname;///<прізвище викладача

string cientific\_topic;///<наукова тема

public:

Scientific\_SR()//конструктор

{

surname = "NO name";

cientific\_topic = "No name";

}

~Scientific\_SR()//деструктор

{}

string Get\_surname() { return surname; }

string Get\_cientific\_topic() { return cientific\_topic; }

void Set\_surname(string value1) { surname = value1; }

void Set\_cientific\_topic(string value) { cientific\_topic = value; }

SR\* clone() {return (SR\*)new Scientific\_SR(\*this); }

void Print\_run(List\_SR&) override;

void Print\_el(List\_SR&) override;

void add\_el(List\_SR&) override;

void Riad\_FL(List\_SR&) override;

int main\_menu(number) override;

void printTableHead() override;

int vstup\_info(number) override;

void dop\_INFO(List\_SR&) override;

void sortSTR(List\_SR&) override;

int kriter(number n) override;

};

class kolektsiya\_SR :public SR

{

private:

string predmet;

string author;///<Автор збірника

string collection;///<Назва збірника

string Vydavnytstvo;///<Видавництво

int data;///<дата публікації

public:

string Get\_predmet() { return predmet; }

string Get\_author() { return author; }

string Get\_collection() { return collection; }

string Get\_Vydavnytstvo() { return Vydavnytstvo; }

int Get\_data() { return data; }

void Set\_predmet(string value) { predmet = value; }

void Set\_data(int value1) { data = value1; }

void Set\_author(string value1) { author = value1; }

void Set\_collection(string value) { collection = value; }

void Set\_Vydavnytstvo(string value1) { Vydavnytstvo = value1; }

SR\* clone() { return (SR\*)new kolektsiya\_SR(\*this); }

void Print\_run(List\_SR&) override;

void Print\_el(List\_SR&) override;

void add\_el(List\_SR&) override;

void Riad\_FL(List\_SR&) override;

int main\_menu(number) override;

void printTableHead() override;

int vstup\_info(number) override;

void dop\_INFO(List\_SR&) override;

void sortSTR(List\_SR&) override;

int kriter(number n) override;

void poshuk(List\_SR& );

void delet\_el(List\_SR&);

void error(number& a, number& d);

// void delete\_list(List\_SR& A);

};

class List\_SR {

protected:

int index = -1;

SR\* data[ARRAY\_SIZE];

public:

List\_SR() { ; }

List\_SR(List\_SR& rhs) {

for (int i = 0; i < rhs.index; i++) {

data[i] = rhs.data[i]->clone();

}

index = rhs.index;

}

~List\_SR()

{

while (index >= 0)

delete data[index--];

}

void add(SR\* a) {

if (index + 1 >= 255) return;

data[++index] = a->clone();

}

SR\* Get\_Elem(int Index) {return data[Index]->clone();}

int Get\_kol\_Index() {return index+1 ;}

List\_SR& operator=(const List\_SR& rhs) {

if (this == &rhs) // a = a;

return \*this;

while (index >= 0)

delete data[index--];

for (int i = 0; i < rhs.index; i++) {

data[i] = rhs.data[i]->clone();

}

index = rhs.index;

}

void del() {

if (index - 1 < -1) return;

delete data[index--];

}

void del\_el(int i)

{

for ( i--; i < index; i++)

{

data[i] = data[i + 1];

}

del();

}

void sort(int i) {

SR\* temp;

temp = data[i];

data[i] = data[i + 1];

data[i + 1] = temp;

}

void delete\_list(List\_SR& A)//

{

while (A.Get\_kol\_Index() > 0)

A.del();

}

};

class mass

{

private:

RGZ A;

Scientific\_SR B;

kolektsiya\_SR C;

public:

int operator[] (const int index)

{

if (index == 1) return A.vstup\_info(index);

else if (index == 2) return B.vstup\_info(index);

else if (index == 3) return C.vstup\_info(index);

else return 0;

}

};RGZ.cpp

………………

Dialog.cpp

**. . . . . . . .**

Main.cpp

……..

Scientific\_SR**.cpp**

**………………**

Scientific\_SR **\_menu**

………………….

kolektsiya\_SR.сpp

void kolektsiya\_SR::add\_el(List\_SR& A)

{

kolektsiya\_SR add;

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

int value5;

std::string value1, value2, value3, value4;

bool condition = true;

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

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву предмета : ";

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

std::cin >> value1;

if ((regex\_match(value1, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value1, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву збирника завдань : ";

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

std::cin >> value2;

if ((regex\_match(value2, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value2, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть призвище автора : ";

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

std::cin >> value3;

if ((regex\_match(value3, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value3, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true; condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

std::cout << "Введiть назву видаництва : ";

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

std::cin >> value4;

if ((regex\_match(value4, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(value3, regex\_integer)) == 0)

{

value1[0] -= 32;

std::cout << "Програма виявила помилку в написанні та успішно випрвила її" << std::endl;

}

condition = false;

}std::cout << "додано" << std::endl;

condition = true;

while (condition) {///////регулярка дати

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введiть рік видання : ";

std::cin >> value5;

if (std::cin.fail()) {

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

cout << "Спробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if (value5 > 0)

condition = false;

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

std::cout << " додано" << std::endl;

add.Set\_predmet(value1);

add.Set\_collection(value2);

add.Set\_author(value3);

add.Set\_Vydavnytstvo(value4);

add.Set\_data(value5);

A.add(&add);

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

std::cout << "\nВашi данi успiшно записанi\n";

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

}

void kolektsiya\_SR::Riad\_FL(List\_SR& A)

{

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

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

kolektsiya\_SR add;

std::ifstream fOpen;

int value, temp = 0;

std::string text;

fOpen.open("zbyrnk\_zavdan.txt", std::ios\_base::in);

if (!fOpen)

{

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

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

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

return;

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

while (!fOpen.eof())

{

fOpen >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

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

std::cout << "1Критична помилка в написаному" << std::endl;

text = "Eror(No name)";

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

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

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

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

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

} add.Set\_predmet(text);////туту

fOpen >> text;

add.Set\_collection(text);////туту

fOpen >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

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

std::cout << "3Критична помилка в написаному" << std::endl;

text = "Eror(No name)";

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

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

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

std::cout << "Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

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

} add.Set\_author(text);/

fOpen >> text;

if ((regex\_match(text, regex\_integer2)) == 0)

{

text = "Eror(No name)";

}

else if ((regex\_match(text, regex\_integer)) == 0)

{

text[0] -= 32;

std::cout << "4Програма виявила помилку в написаному та успішно випрвила її" << std::endl;

}add.Set\_Vydavnytstvo(text);

/\*fOpen >> text;\*/

fOpen >> value;

add.Set\_data(value);

A.add(&add);

}

fOpen.close();

}

void kolektsiya\_SR::Print\_el(List\_SR& A)

{

kolektsiya\_SR\* pr = NULL;

number d, a = 1;

std::cout << "Який елемент хочете вiдобразити: ";

std::cin >> d;

while (a)

{

a = 0;

if (d > 0 && d < A.Get\_kol\_Index())

{

pr = (kolektsiya\_SR\*)A.Get\_Elem(d - 1);

}

else { error(a, d); }

}

pr->printTableHead();

std::cout << "|RGZ-" << d << " |" << std::setw(14) << pr->Get\_predmet() << "|" << std::setw(14) << pr->Get\_collection() << "|" << std::setw(18) << pr->Get\_author()

<< "|" << std::setw(11) << pr->Get\_Vydavnytstvo() << "|" << std::setw(11) << pr->Get\_data() << "|" << std::endl;

std::cout << "-----------------------------------------------------------------------------------\n";

}

void kolektsiya\_SR::Print\_run(List\_SR& A)///спадок вивести

{

std::stringstream f;

int kol = A.Get\_kol\_Index();

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

{

kolektsiya\_SR\* pr = (kolektsiya\_SR\*)A.Get\_Elem(i);

f << "|RGZ-" << i++ << " |" << std::setw(14) << pr->Get\_predmet() << "|" << std::setw(14) << pr->Get\_collection() << "|" << std::setw(18) << pr->Get\_author()

<< "|" << std::setw(11) << pr->Get\_Vydavnytstvo() << "|" << std::setw(11) << pr - Get\_data() << "|"

<< "\n------------------------------------------------------------------------------------\n" << std::endl;

}

cout << f.str() << endl;

}

void kolektsiya\_SR::dop\_INFO(List\_SR& A)

{

kolektsiya\_SR h;

number a;

std::cout << "\n 1: З клавiатири\n 2: 3 Файлу\n Ваш вiбiр: ";

std::cin >> a;

switch (a)

{

case 1:

h.add\_el(A);

break;

case 2:

h.Riad\_FL(A), a = 0;

break;

}

}

void kolektsiya\_SR::poshuk(List\_SR& A)

{

system("cls");

kolektsiya\_SR\* p = NULL;

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

std::regex regex\_integer(("^[А-ЯA-Z][А-Яа-яa-zA-Z0-9 -\_\\.,]\*$"));//проверка на великий символ

std::regex regex\_integer2(("^[A-za-zА-Яа-я][A-za-zА-Яа-я0-9]{1,20}[\\.,]?[-\_ ]?[А-Яа-яA-za-z]{1,20}$"));//перевірка на правильний ввод

string autor;

int perevirka = 0;

bool condition = true;

while (condition)

{

SetConsoleTextAttribute(hConsole, (WORD)(14));

cout << "Введіть прізвище автора :";

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

std::cin >> autor;

if ((regex\_match(autor, regex\_integer2)) == 0)

{

std::cout << "pomylka";

SetConsoleTextAttribute(hConsole, (WORD)(15));

cout << "\nСпробуй ще раз\n";

std::cin.clear();

std::cin.ignore(std::cin.rdbuf()->in\_avail());

continue;

}

if ((regex\_match(autor, regex\_integer)) == 0)

{

autor[0] -= 32;

}

condition = false;

}

cout << endl << endl;

int kol = A.Get\_kol\_Index();

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

{

kolektsiya\_SR\* pr = (kolektsiya\_SR\*)A.Get\_Elem(i);

if (autor == p->Get\_author())

{

perevirka++;

cout << "|RGZ-" << i + 1 << " |" << std::setw(14) << p->Get\_predmet() << "|" << std::setw(14) << p->Get\_collection() << "|" << std::setw(18) << p->Get\_author()

<< "|" << std::setw(11) << p->Get\_Vydavnytstvo() << "|" << std::setw(11) << p->Get\_data() << "|" << std::endl;

std::cout << "-----------------------------------------------------------------------------------\n";

}

}

if (perevirka == 0)

cout << "\nТакого автора незнайдено\n";

system("pause");

}

void kolektsiya\_SR::sortSTR(List\_SR& A)//сортувати елементи

{

number b = 0;

b = kriter(b);

bool a = true;

cout << "Як сортувати\n 1: Вiд меншого до бiльшого \n 2: Вiд бiльшого до меншого \n\n Вибiр: ";

int k = 0, m;

cin >> m;

if (m == 1) k = 1;

else k = -1;

kolektsiya\_SR\* p1 = (kolektsiya\_SR\*)A.Get\_Elem(0);

kolektsiya\_SR\* p2 = (kolektsiya\_SR\*)A.Get\_Elem(0);

while (a)

{

a = false;

switch (b)

{

case 1:

for (int i = 0; i < A.Get\_kol\_Index() - 1; i++)

{

p1 = (kolektsiya\_SR\*)A.Get\_Elem(i);

p2 = (kolektsiya\_SR\*)A.Get\_Elem(i + 1);

if (m == 1)

{

if (p1->Get\_predmet() > p2->Get\_predmet())

{

A.sort(i);

a = true;

}

}

else

{

if (p1->Get\_predmet() < p2->Get\_predmet())

{

A.sort(i);

a = true;

}

}

}

break;

case 2:

for (int i = 0; i < A.Get\_kol\_Index() - 1; i++)

{

p1 = (kolektsiya\_SR\*)A.Get\_Elem(i);

p2 = (kolektsiya\_SR\*)A.Get\_Elem(i + 1);

if (m == 1)

{

if (p1->Get\_collection() > p2->Get\_collection())

{

A.sort(i);

a = true;

}

}

else

{

if (p1->Get\_collection() < p2->Get\_collection())

{

A.sort(i);

a = true;

}

}

}break;

case 3:

for (int i = 0; i < A.Get\_kol\_Index() - 1; i++)

{

p1 = (kolektsiya\_SR\*)A.Get\_Elem(i);

p2 = (kolektsiya\_SR\*)A.Get\_Elem(i + 1);

if (m == 1)

{

if (p1->Get\_author() > p2->Get\_author())

{

A.sort(i);

a = true;

}

}

else

{

if (p1->Get\_author() < p2->Get\_author())

{

A.sort(i);

a = true;

}

}

}break;

case 4:

for (int i = 0; i < A.Get\_kol\_Index() - 1; i++)

{

p1 = (kolektsiya\_SR\*)A.Get\_Elem(i);

p2 = (kolektsiya\_SR\*)A.Get\_Elem(i + 1);

if (m == 1)

{

if (p1->Get\_Vydavnytstvo() > p2->Get\_Vydavnytstvo())

{

A.sort(i);

a = true;

}

}

else

{

if (p1->Get\_Vydavnytstvo() < p2->Get\_Vydavnytstvo())

{

A.sort(i);

a = true;

}

}

}break;

case 5:

for (int i = 0; i < A.Get\_kol\_Index() - 1; i++)

{

p1 = (kolektsiya\_SR\*)A.Get\_Elem(i);

p2 = (kolektsiya\_SR\*)A.Get\_Elem(i + 1);

if (p1->Get\_data() \* k > p2->Get\_data()\* k)

{

A.sort(i);

a = true;

}

}break;

}

}

number d = 0;

string p = "Бажаєте вивести на екран список";

d = yes\_and\_no(d, p);

if (d == 1)

{

Scientific\_SR pr = Scientific\_SR();

pr.printTableHead(), pr.Print\_run(A);

}

}

void kolektsiya\_SR::delet\_el(List\_SR& A)

{

kolektsiya\_SR\* pr = NULL;

number d, a = 1;

std::cout << "Який елемент хочете видалити: ";

std::cin >> d;

while (a)

{

a = 0;

if (d > 0 && d < A.Get\_kol\_Index())

{

A.del\_el(d);

}

else

{

std::cout << "\nНажаль неможливо видалити неiснуючий елемент\n";

std::cout << "Введiть iнше значення ";

std::cin >> d;

a = 1;

}

}

}

kolektsiya\_SR \_menu.cpp

**--------------------------**

**Результати роботи програми:**

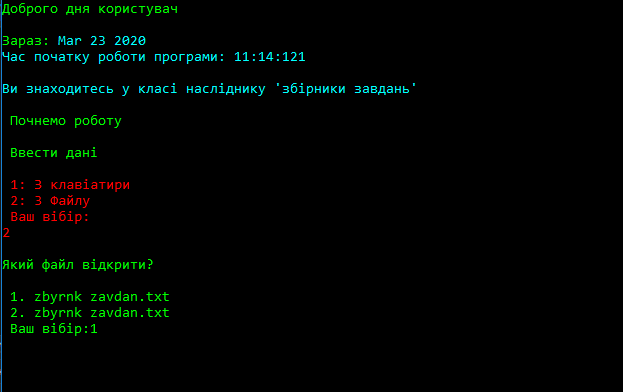
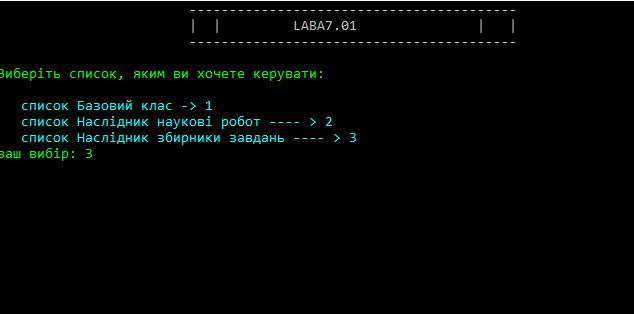


Рисунок 1 –головне меню програми Рисунок 2 – початкове меню класу збырник завдань

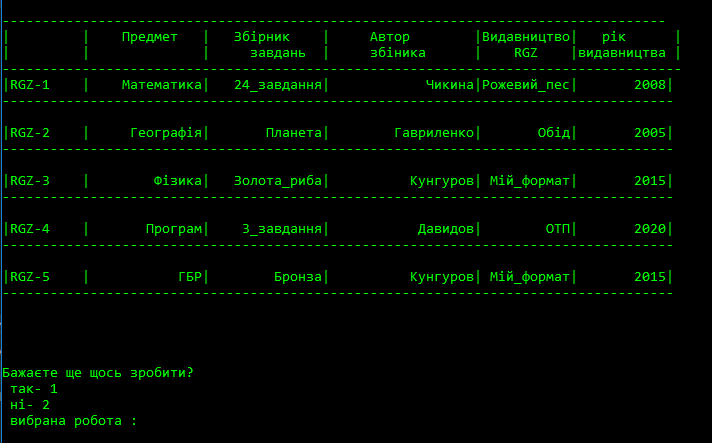
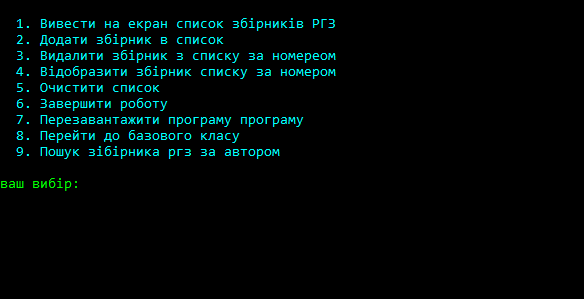
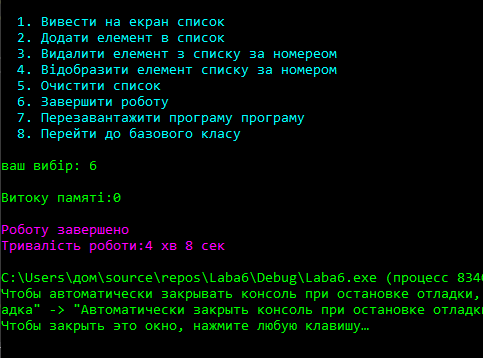


Рисунок 3 –Меню класу збірник задань Рисунок 4 – Результати вивод на екран збірників



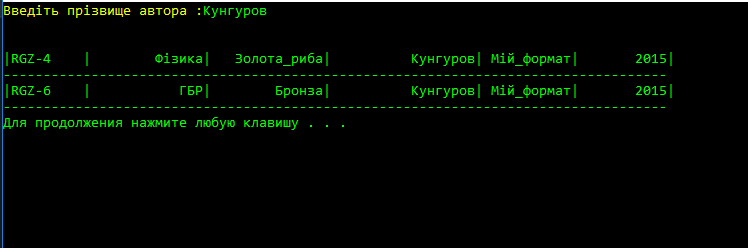


Рисунок 5 –Результат пошуку збірника за автором: Рисунок 6 – Результати кінця програми:

**Висновки:**

В результаті виконання лабораторної роботи отримано  *знання про парадигму ООП – поліморфізм.*