НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №8*

*Тема:* Перевантаження операторів

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** Отримати знання про призначення операторів, визначити їх  
ролі в житті об’єкта та можливість перевизначення.

Варіант 6.

6 Самостійні роботи студентів Розрахунково-графічне завдання

Загальне завдання. Поширити попередню лабораторну роботу

наступним чином:

 в базовому класі, та класі/класах-спадкоємцях перевантажити:

 оператор присвоювання;

 оператор порівняння (на вибір: == , < , > , >= , <= , != );

 оператор введення/виведення;

Тексt програми

prototaype.h

……….

SR.h

------------

class SR {

…..

/////Перегрузки//////

RGZ& operator = (const RGZ&);

RGZ& operator = (const RGZ\*);

friend bool operator > (RGZ& p, RGZ&);

friend bool operator < (RGZ& p, RGZ&);

friend bool operator == (RGZ& p, RGZ&);

friend bool operator != (RGZ& p, RGZ& p2);

friend bool operator <= (RGZ& p, RGZ& p2);

friend bool operator >= (RGZ& p, RGZ& p2);

friend ostream& operator << (ostream&, RGZ&);

friend ifstream& operator >> (ifstream&,RGZ&);

class Scientific\_SR final: public SR{

…. Scientific\_SR& operator = (const Scientific\_SR&);

Scientific\_SR& operator = (const Scientific\_SR\*);

friend bool operator > (Scientific\_SR& p, Scientific\_SR&);

friend bool operator < (Scientific\_SR& p, Scientific\_SR&);

friend bool operator == (Scientific\_SR& p, Scientific\_SR&);

friend bool operator != (Scientific\_SR& p, Scientific\_SR&);

friend bool operator <= (Scientific\_SR& p, Scientific\_SR&);

friend bool operator >= (Scientific\_SR& p, Scientific\_SR&);

friend ostream& operator << (ostream&, Scientific\_SR&);

friend ifstream& operator >> (ifstream&, Scientific\_SR&); }

class kolektsiya\_SR : public SR

{

/////Перегрузки//////

kolektsiya\_SR& operator = (const kolektsiya\_SR&);

kolektsiya\_SR& operator = (const kolektsiya\_SR\*);

friend bool operator > (kolektsiya\_SR& p, kolektsiya\_SR&);

friend bool operator < (kolektsiya\_SR& p, kolektsiya\_SR&);

friend bool operator == (kolektsiya\_SR& p, kolektsiya\_SR&);

friend bool operator != (kolektsiya\_SR& p, kolektsiya\_SR&);

friend bool operator <= (kolektsiya\_SR& p, kolektsiya\_SR&);

friend bool operator >= (kolektsiya\_SR& p, kolektsiya\_SR&);

friend ostream& operator << (ostream&, kolektsiya\_SR&);

friend ifstream& operator >> (ifstream&, kolektsiya\_SR&);

class List\_SR {

…..

/////Перегрузки//////

friend ostream& operator << (ostream& out, List\_SR& a)

{

if (a.Get\_kol\_Index() == 0)

out << "Масив пустий :" << endl;

else

{

out << "Ваш масив :" << endl;

for (int i = 0; i < a.Get\_kol\_Index(); i++)

{

if (typeid(\*a[i]).name() == typeid(RGZ).name())

{

RGZ temp;

temp= (RGZ\*)a[i];

out << temp << endl;

}

else if (typeid(\*a[i]).name() == typeid(Scientific\_SR).name())

{

Scientific\_SR temp;

temp = (Scientific\_SR\*)a[i];

out << temp << endl;

}

else if (typeid(\*a[i]).name() == typeid(kolektsiya\_SR).name())

{

kolektsiya\_SR temp;

temp = (kolektsiya\_SR\*)a[i];

out << temp<< endl;

}

}

out << endl;

}

return out;

} friend ifstream& operator >> (std::ifstream& is, List\_SR& obj)

{

while (!is.eof())

{

RGZ x;

is >> x;

obj.add(&x);

}

return is;

}

SR\* operator[] (const int index) { return data[index]->clone(); }

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;

}

};

RGZ.cpp

/////Перегрузки//////

RGZ& RGZ::operator = (const RGZ& SR)

{

// cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

predmet = SR.predmet;

tema = SR.tema;

termin = SR.termin;

bal = SR.bal;

zavdanj = SR.zavdanj;

return \*this;

}

RGZ& RGZ::operator = (const RGZ\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

predmet = SR->predmet;

tema = SR->tema;

termin = SR->termin;

bal = SR->bal;

zavdanj = SR->zavdanj;

return \*this;

}

bool operator < (RGZ& p, RGZ& p2) {

int i=0;

if (p.Get\_bal() < p2.Get\_bal())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_teme() < p2.Get\_teme())i++;

if (p.Get\_zavdanj() < p2.Get\_zavdanj())i++;

if (p.Get\_termin() < p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (RGZ& p, RGZ& p2)

{

int i=0;

if (p.Get\_bal() > p2.Get\_bal())i++;

if (p.Get\_predmet()>p2.Get\_predmet())i++;

if (p.Get\_teme() > p2.Get\_teme())i++;

if (p.Get\_zavdanj()> p2.Get\_zavdanj())i++;

if (p.Get\_termin() > p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (RGZ& p, RGZ& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (RGZ& p, RGZ& p2) { return !(p == p2) ; }

bool operator <= (RGZ& p, RGZ& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (RGZ& p, RGZ& p2) { return (p == p2) || !(p2 > p); }

/////////////////////////////

Dialog.cpp

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

Main.cpp

……..

Scientific\_SR**.cpp**

/////Перегрузки//////

Scientific\_SR& Scientific\_SR::operator = (const Scientific\_SR& SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

surname = SR.surname;

cientific\_topic = SR.cientific\_topic;

predmet = SR.predmet;

zavdanj = SR.zavdanj;

termin =SR.termin;

return \*this;

}

Scientific\_SR& Scientific\_SR::operator = (const Scientific\_SR\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

surname = SR->surname;

cientific\_topic = SR->cientific\_topic;

predmet = SR->predmet;

zavdanj = SR->zavdanj;

termin = SR->termin;

return \*this;

}

bool operator < (Scientific\_SR& p, Scientific\_SR& p2) {

int i = 0;

if (p.Get\_surname() < p2.Get\_surname())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_cientific\_topic() < p2.Get\_cientific\_topic())i++;

if (p.Get\_zavdanj() < p2.Get\_zavdanj())i++;

if (p.Get\_termin() < p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (Scientific\_SR& p, Scientific\_SR& p2)

{

int i = 0;

if (p.Get\_surname() > p2.Get\_surname())i++;

if (p.Get\_predmet() > p2.Get\_predmet())i++;

if (p.Get\_cientific\_topic() > p2.Get\_cientific\_topic())i++;

if (p.Get\_zavdanj() > p2.Get\_zavdanj())i++;

if (p.Get\_termin() > p2.Get\_termin())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (Scientific\_SR& p, Scientific\_SR& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (Scientific\_SR& p, Scientific\_SR& p2) { return !(p == p2); }

bool operator <= (Scientific\_SR& p, Scientific\_SR& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (Scientific\_SR& p, Scientific\_SR& p2) { return (p == p2) || !(p2 > p); }

//вивод на екран

ostream& operator<< (ostream& out, Scientific\_SR& pr)

{

out << " |" << std::setw(14) << pr.Get\_surname() << "|" << std::setw(14) << pr.Get\_predmet()

<< "|" << std::setw(18) << pr.Get\_cientific\_topic()

<< "|" << std::setw(11) << pr.Get\_termin() << "|" << std::setw(11) << pr.Get\_zavdanj() << "|"

<< "\n------------------------------------------------------------------------------------\n";

return out;

}

//читання з файлу

ifstream& operator >> (ifstream& stream, Scientific\_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

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

std::ifstream fOpen;

int value;

std::string text;

stream >> text;

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

{

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

std::cout << "Критична 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\_surname(text);////туту

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

}add.Set\_predmet(text);

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

}add.Set\_cientific\_topic(text);

stream >> value;

add.Set\_termin(value);

stream >> value;

add.Set\_zavdanj(value);

return stream;

}

////////////////////////////////////////}

Scientific\_SR\_**menu.cpp**

**………**

**kolektsiya\_SR.cpp**

kolektsiya\_SR& kolektsiya\_SR::operator = (const kolektsiya\_SR& SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == &SR)

return \*this;

predmet = SR.predmet;

collection = SR.collection;

author = SR.author;

Vydavnytstvo = SR.Vydavnytstvo;

data = SR.data;

return \*this;

}

kolektsiya\_SR& kolektsiya\_SR::operator = (const kolektsiya\_SR\* SR)

{

//cout << "Перегрузка оператора '='" << endl;

if (this == SR)

return \*this;

predmet = SR->predmet;

collection = SR->collection;

author = SR->author;

Vydavnytstvo = SR->Vydavnytstvo;

data = SR->data;

return \*this;

}

bool operator < (kolektsiya\_SR& p, kolektsiya\_SR& p2) {

int i = 0;

if (p.Get\_collection() < p2.Get\_collection())i++;

if (p.Get\_predmet() < p2.Get\_predmet())i++;

if (p.Get\_author() < p2.Get\_author())i++;

if (p.Get\_Vydavnytstvo() < p2.Get\_Vydavnytstvo())i++;

if (p.Get\_data() < p2.Get\_data())i++;

if (i < 3)return 0;

else return 1;

}

bool operator > (kolektsiya\_SR& p, kolektsiya\_SR& p2)

{

int i = 0;

if (p.Get\_collection() > p2.Get\_collection())i++;

if (p.Get\_predmet() > p2.Get\_predmet())i++;

if (p.Get\_author() > p2.Get\_author())i++;

if (p.Get\_Vydavnytstvo() > p2.Get\_Vydavnytstvo())i++;

if (p.Get\_data() > p2.Get\_data())i++;

if (i < 3)return 0;

else return 1;

}

bool operator== (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return !(p < p2) && !(p > p2); }

bool operator!= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return !(p == p2); }

bool operator <= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return (p == p2) || !(p2 < p); }

bool operator >= (kolektsiya\_SR& p, kolektsiya\_SR& p2) { return (p == p2) || !(p2 > p); }

//вивод на екран

ostream& operator<< (ostream& out, kolektsiya\_SR& pr)

{

out << " |" << 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" ;

return out;

}

//читання з файлу

ifstream& operator >> (ifstream& stream, 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

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

std::ifstream fOpen;

int value;

std::string text;

stream >> 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

stream >> text;

add.Set\_collection(text);////туту

stream >> 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);////туту

stream >> 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;\*/

stream>>add.data;

return stream;

}

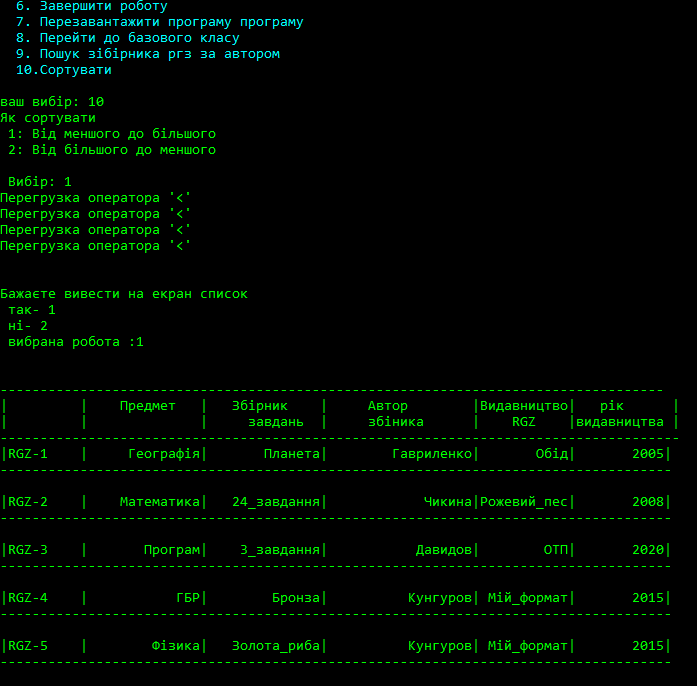
/////////////////////////////

/////////////////////////////////

**kolektsiya\_SR\_menu.cpp**

……………………….

**Результати роботи програми:**



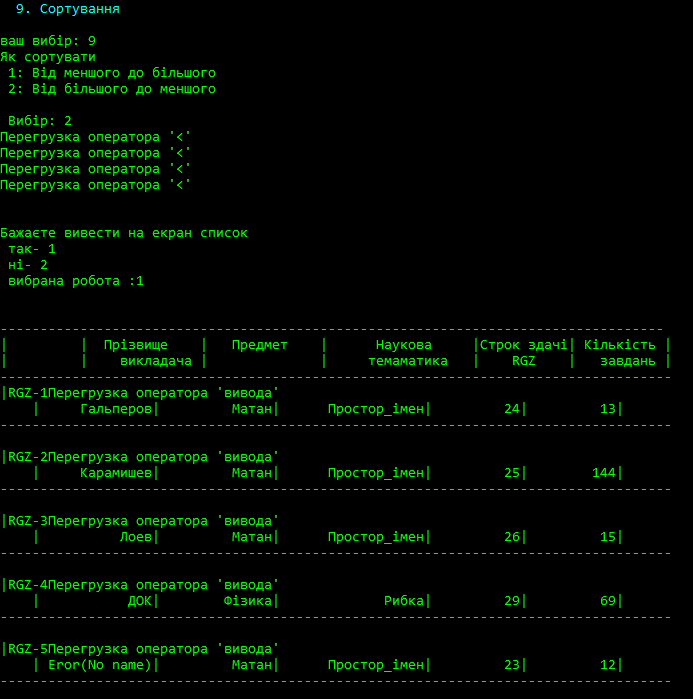
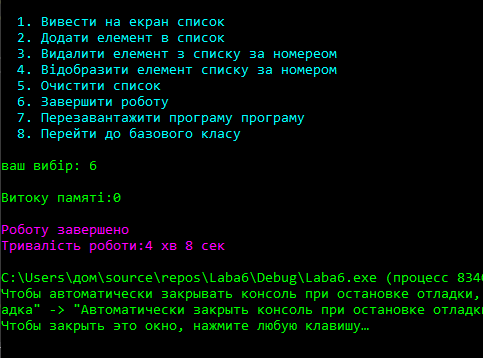


Рис 1 – Результати роботи перегрузки виведення Рис 2 – Результати роботи перегрузки порівняння



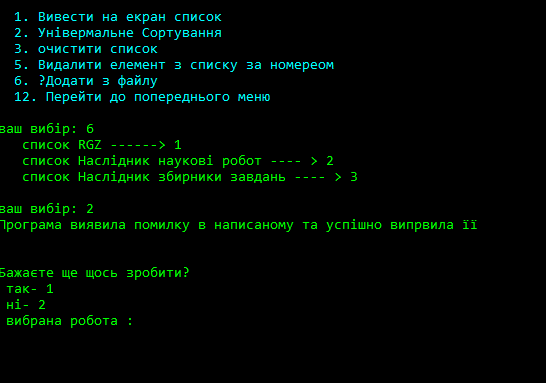


Рисунок 3 – Результат роботи додавання з файлу Рисунок 4 – Результати завершення роботи

**Висновки:**

В результаті виконання лабораторної роботи отримав знання про призначення операторів, та визначив їх ролі в житті об’єкта та можливість перевизначення.