НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звіт з лабораторної роботи №12*

*Тема:* STL

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** Отримати базові знання про STL контейнери. Освоїти основні

механізми роботи з STL контейнерами

Варіант 6.

Загальне завдання.Маючи класи з прикладної області РГЗ (тільки базовий клас та

клас/класи спадкоємці), створити діалогове меню, що дозволяє

продемонструвати роботу STL контейнерів (додавання / видалення /

отримання даних, показ всіх елементів) та показати їх принципову різницю:

* vector;
* set;
* list;
* map .

При цьому врахувати, що контейнери містять елементи одного типу,

наприклад, базового.

Прохід по всьому контейнеру повинен виконуватися за допомогою

циклу мови С++11 – range-for.

Тексt програми

prototaype.h

……….

SR.h

class RGZ

{

private:

string predmet;///<предмет

string tema;///<тема

int bal;///<кількість балів за завдання

int zavdanj;///<кількість завдань

int termin;///<строк здачи РГЗ у днях

public :

RGZ()

{

predmet = "NO name";

tema = "No name";

bal = NULL;

zavdanj = NULL;

termin = NULL;

}

~RGZ() { ; }//деструктор

virtual void show()

{

cout << " |" << std::setw(14) << Get\_predmet() << "|" << std::setw(14) << Get\_teme()

<< "|" << std::setw(18) << Get\_bal() << "|" << std::setw(11) << Get\_termin() << "|" << std::setw(11)

<< Get\_zavdanj() << "|"

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

}

string Get\_predmet()const { return predmet; }

string Get\_teme() const { return tema; }

int Get\_bal() const { return bal; }

int Get\_zavdanj()const { return zavdanj; }

int Get\_termin() const { 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\_termin(int value) { this->termin = value; }

virtual void el(RGZ& add);

virtual int vstup\_info(number nov, int N, vector<RGZ>& A, set<RGZ>& B, list<RGZ>& C, map<int, RGZ>& D,vector<RGZ\*>& E);

template<typename T> void Print\_run(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void Print\_el(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void add\_el(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void Riad\_FL(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void dop\_INFO(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void delet\_el(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E);

template<typename T> void PR(T a);

int main\_menu(number);

void printTableHead();

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

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

/////Перегрузки//////

RGZ& operator = (const RGZ&);

RGZ& operator = (const RGZ\*);

bool operator>(const RGZ& right) const {

return Get\_predmet() > right.Get\_predmet();

}

bool operator<(const RGZ& right) const {

return Get\_predmet() < right.Get\_predmet();

}

bool operator==(const RGZ& right) const {

return Get\_predmet() == right.Get\_predmet() && Get\_teme() == right.Get\_teme() && Get\_bal() == right.Get\_bal() && Get\_zavdanj() == right.Get\_zavdanj() && Get\_zavdanj() == right.Get\_zavdanj();

}

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&,const RGZ&);

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

};

///клас ,що наслідується від RGZ основна задача поширити РГЗ додається два поля surname та cientific\_topic

class Scientific\_SR final : public RGZ

{

private:

string surname;///<прізвище викладача

string cientific\_topic;///<наукова тема

string predmet;

int zavdanj;

int termin;

public:

Scientific\_SR()//конструктор

{

surname = "NO name";

cientific\_topic = "No name";

predmet = "No name";

zavdanj=NULL;

termin=NULL;

}

~Scientific\_SR()//деструктор

{}

void show() {

cout << " |" << std::setw(14) << Get\_surname() << "|" << std::setw(14) << Get\_predmet()

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

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

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

}

string Get\_surname() const { return surname; }

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

string Get\_predmet() const { return predmet; }

int Get\_zavdanj() const { return zavdanj; }

int Get\_termin() const { return termin; }

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

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

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

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

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

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

// int main\_menu(number)

void el(Scientific\_SR& add);

int vstup\_info( number nov, int N, vector<Scientific\_SR>& A, set<Scientific\_SR>& B, list<Scientific\_SR>& C, map<int, Scientific\_SR>& D, vector<RGZ\*>& E);

void printTableHead();

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

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

bool operator>(const Scientific\_SR& right) const {

return Get\_predmet() > right.Get\_predmet();

}

bool operator<(const Scientific\_SR& right) const {

return Get\_predmet() < right.Get\_predmet();

}

bool operator==(const Scientific\_SR& right) const {

return Get\_predmet() == right.Get\_predmet() && Get\_surname() == right.Get\_surname() && Get\_predmet() == right.Get\_predmet() && Get\_zavdanj() == right.Get\_zavdanj() && Get\_termin() == right.Get\_termin();

}

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&);

};

RGZ.cpp

template<typename T> void RGZ::Riad\_FL(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E)

{

T add;

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

std::ifstream fOpen;

int temp = 0;

std::string text, adresa;

if (typeid(A).name() == typeid(vector<RGZ>).name())

{

std::cout << "\nЯкий файл відкрити?\n\n 1. RGZ.txt\n 2. add\_RGZ.txt\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "RGZ.txt";

break;

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

break;

}

}

else adresa = "follower.txt";

fOpen.open(adresa, 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));

//RGZ\* add2 = RGZ;

while (!fOpen.eof()) {

fOpen >> add;

if (N == 1) A.push\_back(add);

else if (N == 2) B.insert(add);

else if (N == 3) C.push\_back(add);

else if (N == 4) D.insert(pair<int, T>(D.size() + 1, add));

else if (N == 5) E.push\_back(new T(add));

}

fOpen.close();

}

template<typename T> void RGZ::Print\_run(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E)//?????

{

if (N == 1)PR(A);

else if (N == 2)PR(B);

else if (N == 3)PR(C);

else if (N == 4)

for (pair<int, T> keys : D) {

cout << "|RGZ-";

cout << keys.first;

keys.second.show();

}

else if (N == 5)

{

int i = 0;

for (auto x : E)

try {

cout << "|RGZ-" << ++i;

x->show();

}

catch (std::invalid\_argument e) { ; }

}

}

template<typename T> void RGZ::PR(T A)

{

int i = 0;

for (auto x : A)

try {

cout << "|RGZ-" << ++i;

x.show();

}

catch (std::invalid\_argument e) { ; }

}

template<typename T> void RGZ::add\_el(int N, vector<T>& A, set<T>& B, list<T>& C, map<int, T>& D, vector<RGZ\*>& E)

{

T add;

el(add);

if (N == 1) A.push\_back(add);

else if (N == 2) B.insert(add);

else if (N == 3) C.push\_back(add);

else if (N == 4) D.insert(pair<int, T>(D.size() + 1, add));

else if (N == 5) E.push\_back(new T(add));

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

std::cin.clear();

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

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

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

}

Dialog.cpp

int Start\_Menu(number n)

{

vector<RGZ> a; // at, push,pop,insert,[]

set<RGZ> b; // insert, begin, end

list<RGZ> c; // push,pop,insert

map<int ,RGZ> d; // at,insert

vector<RGZ\*> e; // На "отлично"

RGZ A;

vector<Scientific\_SR> q; // at, push,pop,insert,[]

set<Scientific\_SR> w; // insert, begin, end

list<Scientific\_SR> r; // push,pop,insert

map<int, Scientific\_SR> t; // at,insert

Scientific\_SR B;

while (n != 0)

{

n = 1;

int k;

cout << "\nВиберіть список, яким контейнером керувати: \n\n";

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

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

cout << " vektor ---> 1\n set ------> 2\n list -----> 3\n map ------> 4\n vektor\* --> 5\n ";

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

cout << " \nваш вибiр: ";

cin >> k;

system("cls");

while (n != 3)

{

cout << "\nВиберіть список, яким ви хочете керувати: \n\n";

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

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

cout << " список RGZ ------------------------ >1\n список Наслідник наукові робот ---- >2\n попереднє меню -------------------- >3 \n Завершити роботу------------------- >4 \n ";

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

cout << " \nваш вибiр: ";

cin >> n;

do

{

switch (n)

{

case 1: n = A.vstup\_info(n, k, a, b, c, d, e);

break;

case 2: n = B.vstup\_info(n, k, q, w, r, t, e);

break;

case 3: n = 3, system("cls");

break;

case 4:a.clear(), b.clear(), c.clear(), d.clear(), q.clear(), w.clear(), r.clear(), t.clear();

for (RGZ\* key : e) {

delete key;

cout << endl;

}

e.clear();

return n;

break;

}

} while (n>3);

}

}

}

Scientific\_SR

--------

Scientific\_SR\_menu

--------

Main.cpp

--------

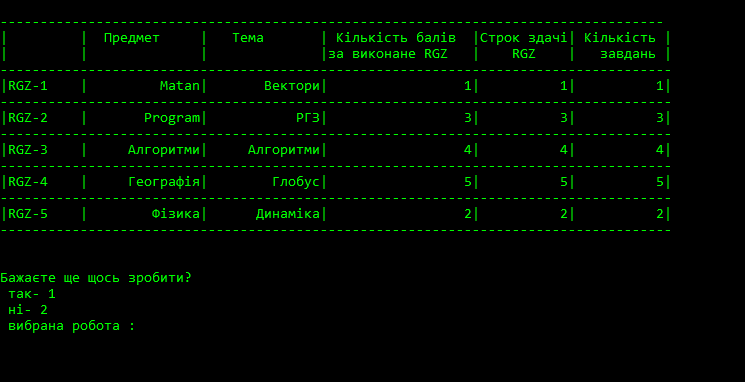
**Результати роботи програми:** 

Рисунок 1 – Результати роботи виводу на екран контейнера

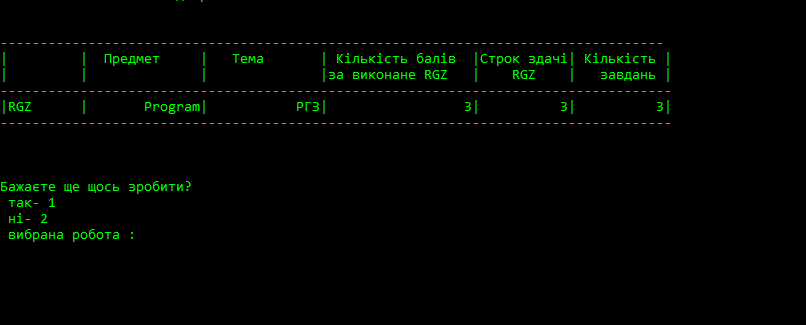


Рисунок 2 – Результати роботи пошуку за номером

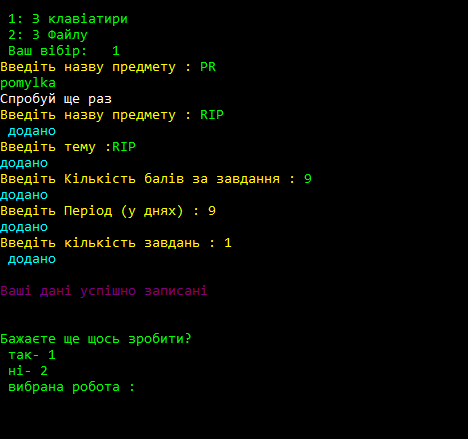


Рисунок 2 – Результати роботи ручого додавання

**Висновки:**

В результаті виконання лабораторної роботи базові знання про STL контейнери. Освоїв основні механізми роботи з STL контейнерами