НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №3*

*Тема: «Потоки»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** *Отримати знання про основи роботи з потоковим введенням /*

*виведенням на мові С++, роботу з файлами та рядками типу string.*

*Індивідуальне завдання*

Варіант 6.

6 Самостійні роботи студентів Розрахунково-графічне завдання

Загальне завдання. Поширити попередню лабораторну роботу  
наступним чином:  
− використання функцій printf/scanf замінити на використання cin/cout;  
− усі конкатенації рядків замінити на використання stringstream;  
− замінити метод виводу інформації про об’єкт на метод, що повертає  
рядок-інформацію про об’єкт, який далі можна виводити на екран;  
− замінити метод вводу інформації про об’єкт на метод, що приймає  
рядок з інформацією про об’єкт, обробляє його та створює об’єкт на  
базі цієї інформації;  
− поширити клас-список, шляхом реалізації методів роботи з файлами  
за допомогою файлових потоків (fstream) (якщо використовувалися  
функції fprintf/fscanf – замінити їх на класи ifsteam/ofstream), при  
цьому сигнатури методів повинні виглядати наступним чином:  
− читання: void CList::readFromFile(string fileName);  
де CList – клас-список об’єктів, при цьому слід пам’ятати, що  
при повторному читанні з файлу, попередні дані списку  
повинні бути очищені;  
− запис: void CList::writeToFile(string fileName);

**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>

#include <ostream>

#include <sstream>

#define \_CRTDBG\_MAP\_ALLOC

#include <stdlib.h>

#include "crtdbg.h"

#include <iomanip>

#include <Windows.h>

using std::cout;

#define DDEBUG

//using namespace std;

struct RGZ;

class SR;

void Init(RGZ\*&);

int vstup\_info(int);

void dop\_INFO(RGZ\*);

void Riad\_FL(RGZ\*);

void addProgram1(RGZ\*);

int main\_menu(int);

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

SR.h

#pragma once

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

class SR

{

private:

std::string predmet;

std::string tema;

int bal;

int zavdanj;

int termin;

int cathedra;

std::stringstream ss;

public:

SR()//конструктор

{

predmet = "NO name";

tema = "No name";

bal = NULL;

zavdanj = NULL;

termin = NULL;

cathedra = NULL;

// std::cout << "\nВизвано конструктор за впровадженням"<< std::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) {

// std::cout << "\nВизвано конструктор копiювання" <<std::endl;

}

SR(std::string predmet,std::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;

//std::cout << "\nКонструктор параметрами" << std::endl;

}

~SR()//деструктор

{

//std::cout << "\nВизвано деструктор " << std::endl;

}

void printTableHead();

void Init(RGZ\*&);

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

void Print\_el(RGZ\* p);

// void add\_el(int ,std::string, std::string, int, int, int, RGZ\* p);//додати //більше не використовується

void add\_el(RGZ\* p);// перероблено та модрнзовано

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

void delete\_list(RGZ\* p);//удалить список

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

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

void save\_for\_Fl(RGZ\*);//зберегти все у файл

void print\_save\_for\_Fl();//вивести файл на екран

void error( int&,int&);

std::string Get\_predmet() { return predmet; }

std::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(std::string value1) { this->predmet = value1; }

void Set\_teme(std::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"

void Init(RGZ\*& p) {

p = new RGZ;

p->next = NULL;

}

int vstup\_info(int nov)

{

RGZ\* h=NULL ;

int a = 1, u = 0;

Init(h);

//addProgram1(h);

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

std::cout << "Доброго дня користувач\n ";

std::cout << "\nЗараз: ";

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

std::cout << \_\_DATE\_\_;

std::cout << "\nЧас початку роботи програми: ", SetConsoleTextAttribute(hConsole, (WORD)(10));

std::cout << \_\_TIME\_\_<< SetConsoleTextAttribute(hConsole, (WORD)(11));

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

std::cout << "\n\n Почнемо роботу\n\n Ввести данi\n ";

while (a != 0)

{

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

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

std::cin >> a;

switch (a)

{

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

std::cout << "Скiльки елементiв ви хочете додати? :";

std::cin>> 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.delete\_list(h);

std::cout << "Успішно";

Init(h);

break; }

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

break;

case 7:nov = 0;

h->rgz.delete\_list(h);

SR();

return 0;

case 8:

h->rgz.delete\_list(h);

{system("cls");

nov = 1;

return 1;

}

case 9:

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

break;

case 10:

h->rgz.save\_for\_Fl(h);

break;

case 11: h->rgz.print\_save\_for\_Fl();

break;

}

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

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

if (a == 1)

a = main\_menu(a);

else a = 7;

}

}

void dop\_INFO(RGZ\* h)

{

int a;

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

std::cin>> a;

switch (a)

{

case 1:

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

break;

case 2:

Riad\_FL(h), a = 0;

break;

}

}

void Riad\_FL(RGZ\* h)

{

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

RGZ\* p = h;

std::ifstream fOpen;

int value, temp = 0;

std::string text, adresa;

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;

}

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));

while (!fOpen.eof()) {

RGZ\* add=new RGZ;

add->next = p->next;

p->next = add;

fOpen >> text;

if ( text[0] >= 'a' && text[0] <= 'z' || text[0] >= 'а' && text[0] <= 'я' || text[0] == 'ё')

text[0] -= 32;

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

fOpen >> text;

if (text[0] >= 'a' && text[0] <= 'z' || text[0] >= 'а' && text[0] <= 'я' || text[0] == 'ё')

text[0] -= 32;

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

{

RGZ\* pr = p->next;

int i = 1;

std::stringstream f;

while (pr != NULL)

{

f <<"|RGZ-"<<i++<<" |"<< std::setw(14) << pr->rgz.Get\_predmet() << "|" << std::setw(14) << pr->rgz.Get\_teme() << "|" << std::setw(18) << pr->rgz.Get\_bal() << "|" << std::setw(11) << pr->rgz.Get\_termin() << "|" << std::setw(11) << pr->rgz.Get\_zavdanj() << "|" <<"\n------------------------------------------------------------------------------------\n"<<std::endl;

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

std::cout << std::setw(14) << pr->rgz.Get\_predmet() << "|" << std::setw(14) << pr->rgz.Get\_teme() << "|" << std::setw(18) << pr->rgz.Get\_bal() << "|" << std::setw(11) << pr->rgz.Get\_termin() << "|" << std::setw(11) << pr->rgz.Get\_zavdanj() << "|" << std::endl;

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

pr = pr->next;

}

std::cout << f.str();

}

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

{

RGZ\* pr = p;

int d;

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

std::cin>> 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);

std::cout << std::setw(14) << pr->rgz.Get\_predmet() << "|" << std::setw(14) << pr->rgz.Get\_teme() << "|" << std::setw(18) << pr->rgz.Get\_bal() << "|" << std::setw(11) << pr->rgz.Get\_termin() << "|" << std::setw(11) << pr->rgz.Get\_zavdanj() << "|" << std::endl;

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

}

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

{

RGZ\* pr = p->next;

RGZ\* temp = p;

int d;

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

std::cin>> 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;

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

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

std::cin>>d;

a = 1;

}

}else{

temp = p;

pr = p->next;

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

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

std::cin >> d;

a = 1;

}

}

{

temp->next = pr->next;

delete(pr);

}

}

void SR::delete\_list(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 SR::curriculum(RGZ\*p)

{

RGZ\* pr=p->next;

int cu = 0, i = 0;

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

std::cin >> cu;

while (pr->next != 0)

{

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

i++;

pr = pr->next;

}

if (i > 0)

{

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

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

int king;

std::cin >> king;

if (king == 1)

{

std::cout << "\n З таких предметiв :";

pr = p;

while (pr->next != 0)

{

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

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

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

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

pr = pr->next;

}

}

else

return;

}

else

{

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

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

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

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

}

}

void SR::checking\_the\_constructor()

{

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

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

SR konst2;//для конструктора

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

SR konst3(konst2);//для конструктора копіювання

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

}

void SR::save\_for\_Fl(RGZ\* h)//зберегти все у файл

{

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

RGZ\* p = h->next;

std::string adresa;

int temp = 0;

std::fstream fOpen;

std::cout << "\nЯкий файл відкрити?\n\n 1. txt.txt(текстовий файл)\n 2. bib.bin(бінарний файл)\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "txt.txt";

break;

case 2:adresa = "bin.bin";

break;

}

fOpen.open(adresa, std::ios::out);

if (!fOpen)

{

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

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

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

return;

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

while (p != 0) {

fOpen << p->rgz.Get\_predmet() << " " << p->rgz.Get\_teme() << " " << p->rgz.Get\_bal() << " " << p->rgz.Get\_zavdanj() << " " << p->rgz.Get\_termin() << " " << p->rgz.Get\_cathedra() << std::endl;

p = p->next;

}

fOpen.close();

std::cout << "\nЗбереження успішно завершено" << std::endl;

}

void SR::print\_save\_for\_Fl()

{

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

/\* RGZ\* p = h;\*/

std::ifstream fOpen;

int value, temp = 0;

std::string text, adresa;

std::cout << "\nЯкий файл відкрити?\n\n 1. txt.txt(текстовий файл)\n 2. bib.bin(бінарний файл)\n Ваш вiбiр:";

std::cin >> temp;

switch (temp)

{

case 1:adresa = "txt.txt";

break;

case 2:adresa = "bin.bin";

break;

}

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

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;

std::cout << text << " ";

fOpen >> text;

std::cout << text << " ";

fOpen >> value;

std::cout << value << " ";

fOpen >> value;

std::cout << value << " ";

fOpen >> value;

std::cout << value <<" ";

fOpen >> value;

std::cout << value<< std::endl;

}

}

void SR::add\_el(RGZ\* h) {

RGZ\* add = new RGZ;

RGZ\* head = h;

int value3, value4, value5, value6;

std::string value1, value2;

bool condition = true;

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

while (condition)

{

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

std::cout << "Ввеідіть назву предмету : "<<std::endl;

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

std::cin >> value1;

condition = false;

if (value1[0] >= 'a' && value1[0] <= 'z' || value1[0] >= 'а' && value1[0] <= 'я' || value1[0] == 'ё')

value1[0] -= 32;

}

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

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

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

if ((std::cin.rdbuf()->in\_avail()) != 1);

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

std::cout << "Введіть тему ";

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

std::cin >> value2;

condition = false;

if (value2[0] >= 'a' && value2[0] <= 'z' || value2[0] >= 'а' && value2[0] <= 'я' || value2[0] == 'ё')

value2[0] -= 32;

}

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

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

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

if ((std::cin.rdbuf()->in\_avail()) != 1);

std::cout << "Введіть Кiлькiсть балiв за завдання ";

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

std::cin >> value3;

if (std::cin.fail()) {//Если есть ошибка, выводим сообщение

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

cout << "Спробуйте ще раз\n";

std::cin.clear();//Почистили поток

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

continue;

}

if (value3 > 0)

condition = false;

else

cout << "Ви сильно недооцiнюєте складнiсть цього RGZ";

}

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

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

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

if ((std::cin.rdbuf()->in\_avail()) != 1);

cout << "Введіть Перiод (у днях) : ";

std::cin >> value4;

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

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

cout << "Спробуй ще раз\n";

std::cin.clear();

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

continue;

}

if(value4 > 0)

condition = false;

else

cout << "Навіть професіоналу на це потребується більше часу";

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

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

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

cout << "Введіть кількість завдань : ";

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;

else

cout << "Навiщо взагалi його робити?";

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

std::cout << "додано" << std::endl;

condition = true;

while (condition) {

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

cout << "Введіть факультет/кафедру : ";

std::cin >> value6;

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;

}

condition = false;

if (value6 > 0)

condition = false;

else

cout << "Ви впевнені, що теке існує? ";

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

std::cout << " додано"<<std::endl;

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

add->next = NULL;

add->rgz.Set\_predmet(value1);

add->rgz.Set\_teme(value2);

add->rgz.Set\_bal(value3);

add->rgz.Set\_termi(value4);

add->rgz.Set\_zavdanj(value5);

add->rgz.Set\_cathedra(value6);

head->next = add;

std::cin.clear();

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

}

Dialog.cpp

#include "SR.h"

#include "prototaype.h"

int main\_menu(int e)

{

std::cout << " \n 1. Вивести на екран список\n";

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

std::cout << " 2. Додати елемент в список\n";

std::cout << " 3. Видалити елемент з списку за номереом\n";

std::cout << " 4. Відобразити елемент списку за номером\n";

std::cout << " 5. Очистити список \n";

std::cout << " 6. Визначити кiлькiсть РГЗ, що виконує студент за весь\n перiод навчання в iнститутi у вiдповiдностi до\n навчального плану\n";

std::cout << " 7. Завершити роботу \n";

std::cout << " 8. Перезавантажити програму програму \n";

std::cout << " 9. Перевірити роботу конструкторів \n";

std::cout << " 10.Записати данi y файл \n";

std::cout << " 11.Вивести данi файлу на екран \n";

std::cout << " \nваш вибiр: ";

std::cin>>e;

return e;

}

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

{

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

std::cout << "так- 1\n ";

std::cout << "нi- 2\n ";

std::cout << "вибрана робота :";

std::cin>> t;

if (t == 2)

{

int oo = 12;

}

return t;

}

void SR::printTableHead()

{

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

std::cout << "| | Предмет | Темa | Кiлькiсть балiв |Строк здачi| Кiлькiсть |"<<std::endl;

std::cout << "| | | |за виконане RGZ | RGZ | завдань |"<< std::endl;

std::cout << "------------------------------------------------------------------------------------"<< std::endl;

}

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

{

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

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

std::cin >> d;

a = 1;

}

Main.cpp

#include "prototaype.h"

#include "SR.h"

int main()

{

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

std::cout << "\t\t\t| | LABA3.0-3 | |\n";

std::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();

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

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

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

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

return 0;

}

**Результати роботи програми:**

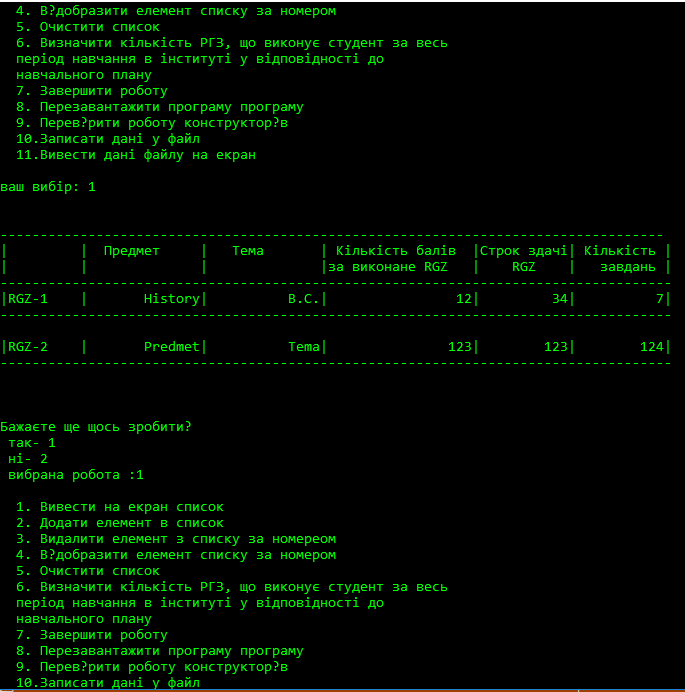
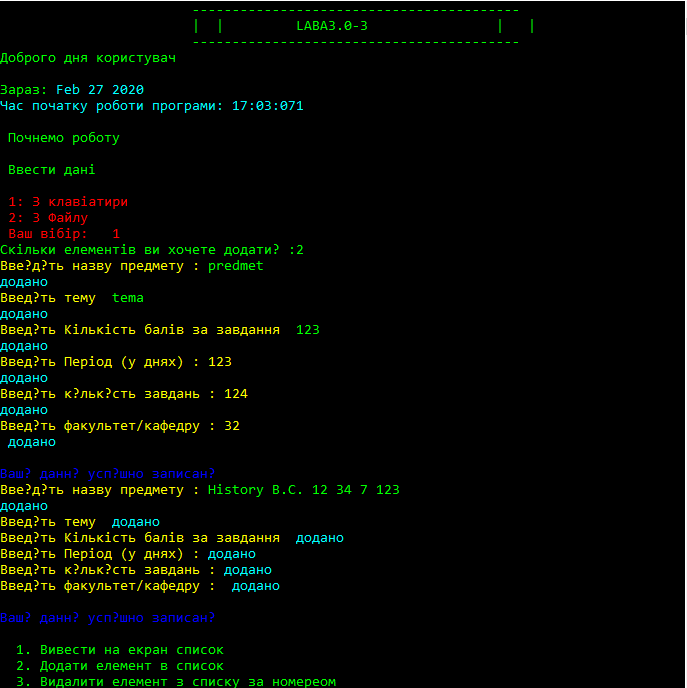


Рисунок 1 – Результати роботи потокового вводу Рисунок 2 – Результати роботи потокового виводу

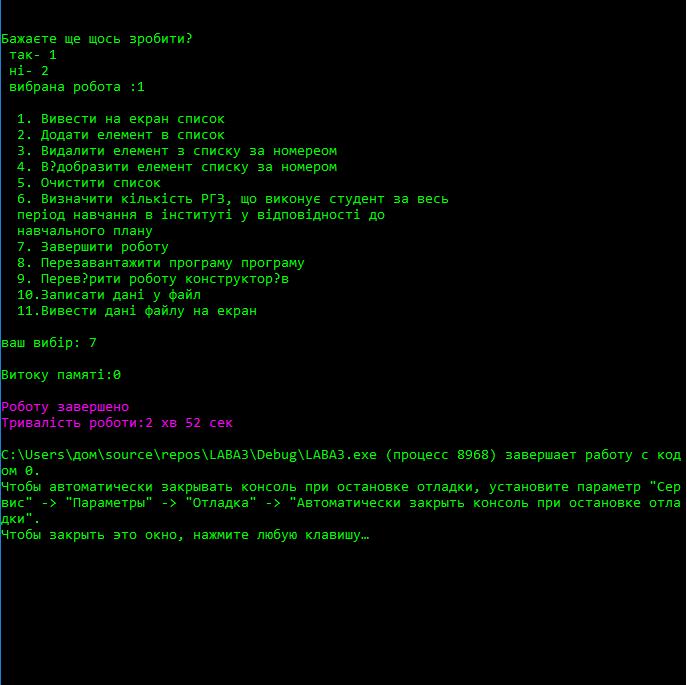
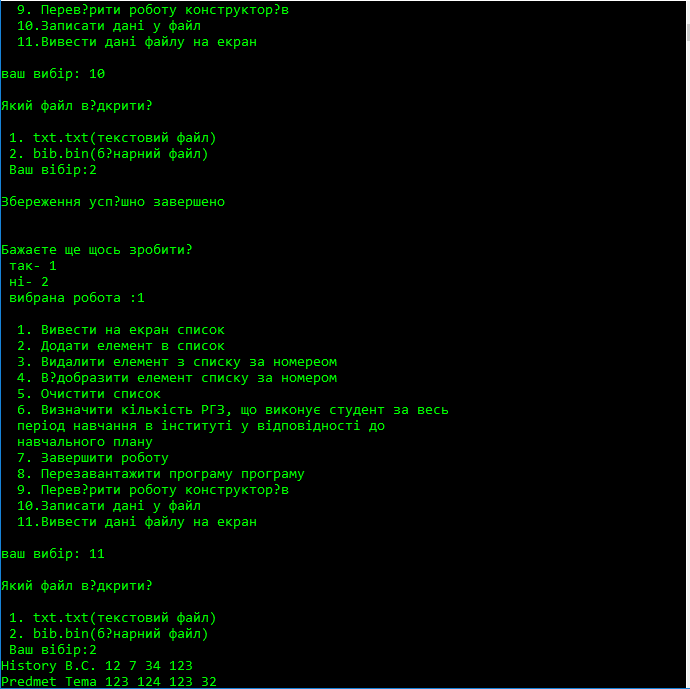


Рисунок 3 – Результати роботи читання файлу Рисунок 4 – Результати роботи програми

**Висновки:**

В результаті виконання лабораторної роботи отримано знання про основи роботи з потоковим введенням /виведенням на мові С++, роботу з файлами та рядками типу string.