НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.2 »

*Звiт з лабораторної роботи №5*

*Тема: «Класи. Агрегація. Композиція. Ключові слова typedef та auto.»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Асистент Віктор ЧЕЛАК

Асистент Владислав ЯЛОВЕГА

Харків – 2020

***Мета:*** *Порівняти поняття агрегація та композиція. Отримати знання*

*про призначення ключових слів typedef та auto.*

Варіант 6.

6 Самостійні роботи студентів Розрахунково-графічне завдання

Дослідити заздалегідь визначені типи даних з бібліотеки <cstddef> /  
<stddef.h>. Модернізувати розроблені у попередній роботі класи наступним  
чином:  
− замінити типи даних, що використовуються при індексуванні на типи  
з вказаної бібліотеки;  
− створити власний синонім типу, визначивши його необхідність;  
− створити/оновити функцію сортування масиву, де крім поля, по  
якому виконується сортування, передається і вказівник на функцію,  
яка визначає напрям сортування;  
− в базовий клас додати два поля, що мають кастомний тип даних (тип  
даних користувача) та які будуть відображати відношення  
«агрегація» та «композиція», при цьому оновити методи читання та  
запису об’єкта;  
− ввести використання ключового слова auto як специфікатор  
зберігання типу змінної. Визначити плюси та мінуси цього  
використання

prototaype.h

……….

SR.h

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

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

{

private:

string predmet;

string tema;

int bal;

int zavdanj;

int termin;

int cathedra;

std::stringstream ss;

class cathedr//композиция

{

private:

number\_SR cathedra;

public:

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

int Get\_cathedra() { return cathedra; }

};

…….

Arbeit.cpp

#include "prototaype.h"

#include "SR.h"

…………………..

oid SR::sortSTR(RGZ\* pr)//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;

while (a)

{

RGZ\* p = pr;

RGZ\* p1 = pr->next;

RGZ\* p2 = p1->next;

a = false;

switch (b)

{

case 1:

while (p2)

{

if (m == 1)

{

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

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}

else

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

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}

break;

case 2:

while (p2)

{

if (m == 1)

{

if (p1->rgz.Get\_teme() > p2->rgz.Get\_teme())

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}

else

if (p1->rgz.Get\_teme() < p2->rgz.Get\_teme())

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}break;

case 3:

while (p2)

{

{if (p1->rgz.Get\_termin()\*k > p2->rgz.Get\_termin()\*k)

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}

}break;

case 4:

while (p2)

{

if (p1->rgz.Get\_zavdanj()\*k > p2->rgz.Get\_zavdanj()\*k)

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}break;

case 5:

while (p2)

{

if (p1->rgz.Get\_bal()\*k > p2->rgz.Get\_bal()\*k)

{

sortN(p, p1, p2);

a = true;

}

else

sortK(p, p1, p2);

}break;

}

}

number d = 0;

char p[] = "Бажаєте вивести на екран список";

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

if (d == 1)

pr->rgz.printTableHead(),Print\_run(pr);

}void SR::sortN(RGZ\*& p, RGZ\*& p1, RGZ\*& p2)

{

p1->next = p2->next;

p2->next = p1;

p->next = p2;

p = p2;

p2 = p1->next;

}

void SR::sortK(RGZ\*& p, RGZ\*& p1, RGZ\*& p2)

{

p = p1;

p1 = p2;

p2 = p2->next;

}

Dialog.cpp

nt SR::kriter(number n)

{

printf("\nЗа яким крiтерiєм хочете сортувати список \n\n");

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

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

printf(" 1.Предмет\n");

printf(" 2.Тема\n");

printf(" 3.Термін \n");

printf(" 4.завдання\n");

printf(" 5.бал\n");

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

printf(" \nваш вибiр: ");

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

return n;

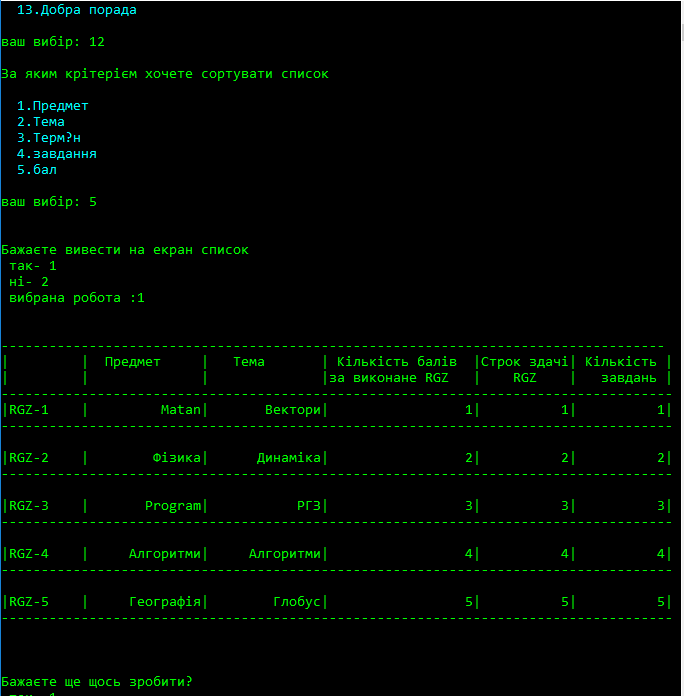
}

Main.cpp

……..

}

**Результати роботи програми:**



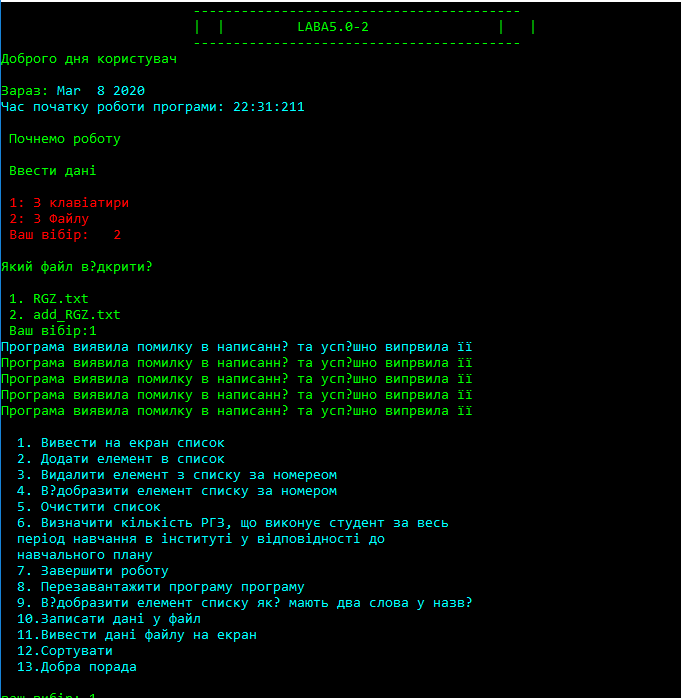
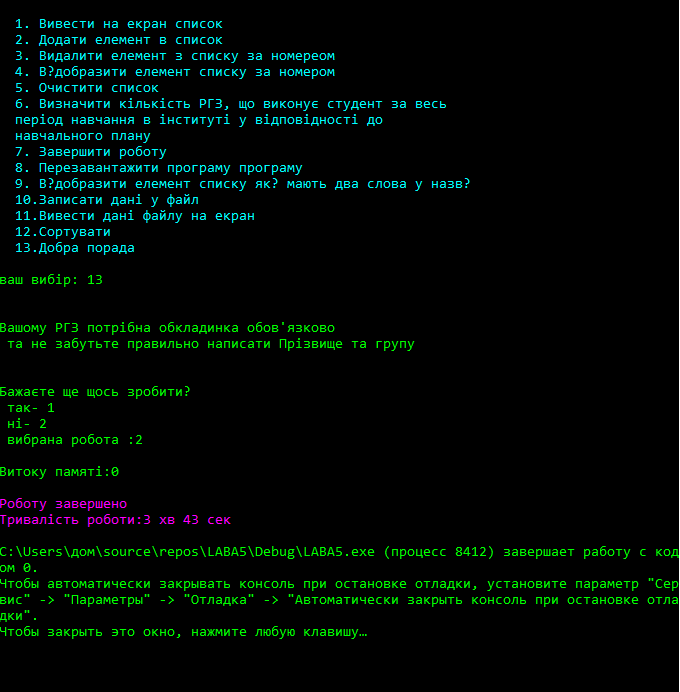


Рисунок 1 – Результати роботи програми:1 Рисунок 2 – Результати роботи програми:2



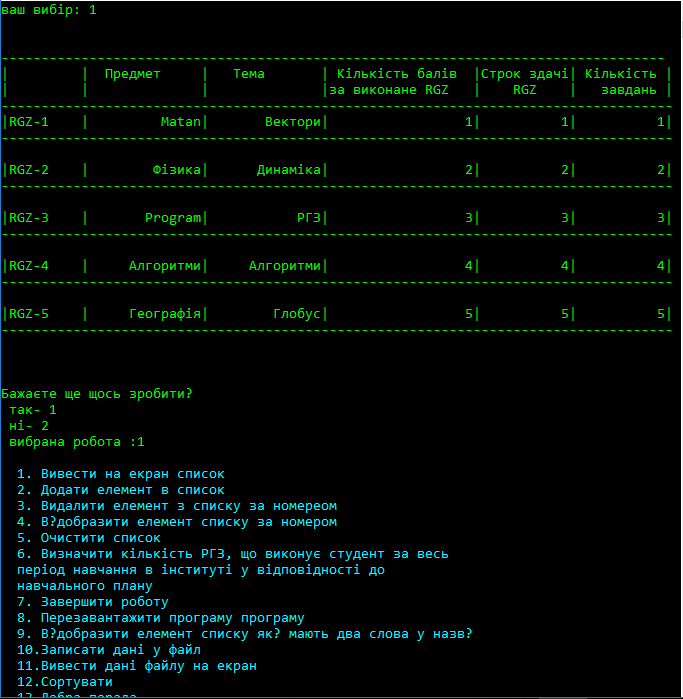


Рисунок 3 – Результати роботи програми:3 Рисунок 4 – Результати роботи програми:4

**Висновки:**

В результаті виконання лабораторної роботи отримано знання про базові регулярні вирази та досвід роботи по застосуванню їх на практиці.