МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ТЕХНІЧНИЙ КОЛЕДЖ

ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА ПУЛЮЯ

**ЗВІТ**

**з лабораторних робіт**

**з дисципліни:**

**«Об'єктно-орієнтоване програмування»**

Розробила

студентка групи КН-221

Лазаровська Вікторія Сергіївна

Перевірив

Викладач

\_\_\_\_\_\_\_\_\_ Слободян Р.О.

Тернопіль 2019

ЛАБОРАТОРНА РОБОТА №1

**Тема:** Вивчення базових понять класу.

**Мета:** Закріпити базові знання про клас. Навчитись створювати класи засобами мови С++.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Створіть клас Int, що імітує стандартний тип int. Єдине поле цього класу повинно мати тип int. Створіть методи, які будуть встановлювати значення поля рівним нулю, ініціалізувати його цілим значенням, виводити значення поля на екран і складати два значення типу Int. Напишіть програму, в якій будуть створені три об'єкти класу Int, два з яких будуть ініціалізованими. Додайте два ініціалізованних об'єкта, надайте результат третьому, а потім відобразіть результат на екрані.

#include <iostream>

#include <string.h>

using namespace std;

class **Int**

{

int value;

public:

void **setzero**()

{

value = 0;

}

int **getzero**()

{

return value;

}

void **set**(int a)

{

value = a;

}

void **summa**(Int a, Int b)

{

this->value = a.value + b.value;

}

void **print**(){

cout << "Value=" << this->value;

}

};

int **main**()

{

Int r1, r2, r3;

r1.set(5);

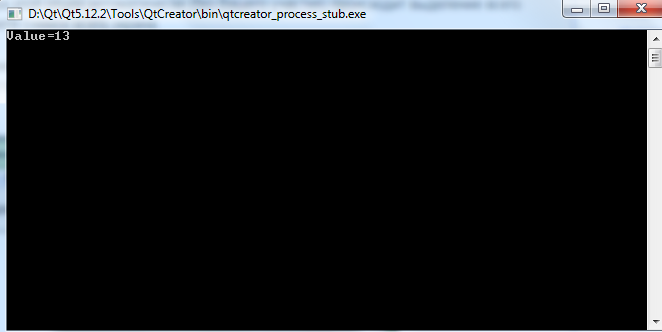
r2.set(8);

r3.summa(r1, r2);

r3.print();

return 0;

}



**Завдання 2.** Уявіть пункт для прийому платежів за проїзд по автостраді. Кожна проїжджаюча машина повинна заплатити за проїзд 50 центів, однак частина машин платить за проїзд, а частина проїжджає безкоштовно. У касі ведеться облік числа проїхавших машин і сумарна виручка від плати за проїзд. Створіть модель такої каси за допомогою класу Kasa. Клас повинен містити два поля. Одне з них, типу unsigned int, призначене для обліку кількості проїхали автомобілів, а друге, що матиме тип double, міститиме сумарну виручку від оплати проїзду. Конструктор повинен ініціалізувати обидва поля нульовими значеннями. Метод payingCar () інкрементує число машин і збільшує на 0,50 сумарну виручку. Інший метод, nорауСаг (), збільшує на одиницю число автомобілів, але залишає без зміни виручку. Метод display () виводить обидва значення на екран. Там, де це можливо, зробіть методи константними. Створіть програму, яка продемонструє роботу класу. Програма повинна запропонувати користувачеві натиснути одну клавішу для того, щоб зімітувати оплату водієм, і іншу клавішу, щоб зімітувати несумлінного водія. Натискання клавіші Esc повинно привести до видачі поточних значень кількості машин і виручки, і до завершення програми.

#include <iostream>

using namespace std;

class **Kasa**

{

unsigned int avto;

double summa;

public:

**Kasa**()

{

avto = 0;

summa = 0.0;

}

void **payingCar**()

{

avto++;

summa+=0.5;

}

void **nopayCar**()

{

avto++;

}

void **Display**()

{

cout << "amount car=" << avto << "\n" << "Summa=" << summa << endl;

}

};

int **menu**()

{

int variant;

cout << "Variant\n" << endl;

cout << "1.Paying Car\n" << endl;

cout << "2.No Paying Car\n" << endl;

cout << "3.Summa\n" << endl;

cin >> variant;

return variant;

}

int **main**()

{

int variant;

Kasa kasa;

do{

variant = menu();

switch (variant) {

case 1:

kasa.payingCar();

cout << "Car pay\n" << endl;

break;

case 2:

kasa.nopayCar();

cout << "Car no pay\n" << endl;

break;

case 3:

cout << "Suma\n" << endl;

kasa.Display();

break;

}

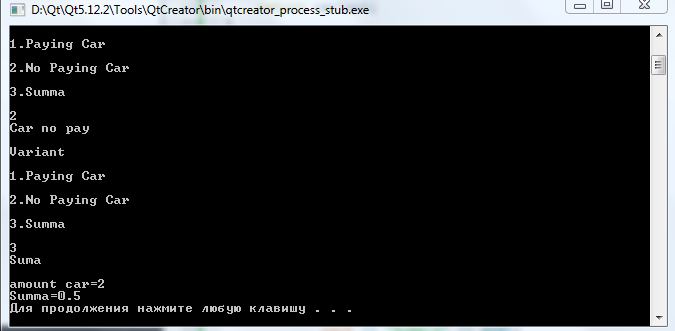
}

while (variant!=3);

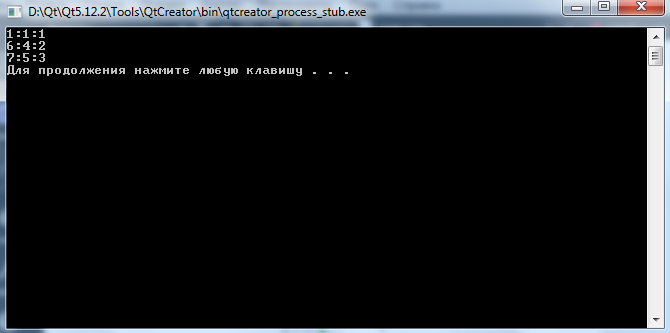
system("pause");

return 0;

}



**Завдання 3.** Створіть клас з ім'ям time, що містить три поля типу int, призначених для зберігання годин, хвилин і секунд. Один з конструкторів класу повинен ініціалізувати поля нульовими значеннями, а інший конструктор - заданим набором значень. Створіть метод класу, який буде виводити значення полів на екран у форматі 11:59:59, і метод, складає значення двох об'єктів типу time, переданих в якості аргументів. У функції main () слід створити два ініціалізованих об'єкта (подумайте, чи повинні вони бути константними) і один неініціалізованний об'єкт. Потім складіть два ініціалізованих значення, а результат надайте третьому об'єкту і виведіть його значення на екран. Де можливо, зробіть методи константними.



**Завдання 4.** У кожному завданні потрібно реалізувати клас. У програмі обов'язково повинні бути продемонстровані різні способи створення об'єктів і масивів об'єктів. Програма повинна демонструвати використання всіх функцій і методів. У всіх завданнях обов’язково повинні бути присутні:

• метод ініціалізації Init( ), метод повинен контролювати значення аргументів на коректність;

• введення з клавіатури Read( );

• виведення на екран Display( ).

Атрибут first –дробове додатнє число , ціна товару; Атрибут second – ціле додатнє число , кількість одиниць товару. Реалізувати метод cost() – обчислення вартості товару.

#include <iostream>

using namespace std;

class **Class**{

double first;

int second;

public:

double **cost**()

{

return first \* second;

}

void **Init**(int f, int s)

{

first = f;

second = s;

}

void **Read**()

{

cout << "Enter first" << endl;

cin >> first;

cout << "Enter second" << endl;

cin >> second;

}

void **Display**()

{

cout << "first\t" << first << endl;

cout << "second\t" << second << endl;

}

};

int **main**()

{

Class a;

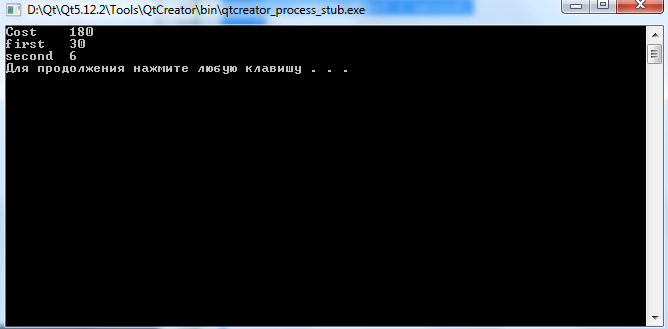
a.Init(30.3, 6);

cout << "Cost\t" << a.cost() << endl;

a.Display();

system("pause");

}



**Висновок:** На лабораторній роботі закріпили базові знання про клас. Навчились створювати класи засобами мови С++

ЛАБОРАТОРНА РОБОТА №2

**Тема:** Робота з лінійними списками. Конструктор і деструктор класу

**Мета:** Навчитись використовувати конструктори і деструктори класів, створювати класи для опису лінійних списків

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Реалізувати один простий клас згідно варіанту індивідуального завдання, що містить закриті данні, а саме два типа даних: числове та рядкове, реалізоване через покажчик на char. Потрібно створити декілька екземплярів класу статично і динамічно, а також масив та продемонструвати дію всіх конструкторів і методів. Реалізувати методи:

• Scan - ввід даних з клавіатури у поля класу;

• Print - констатний метод виводу даних на екран;

• Конструктор по замовчуванню;

• Конструктор ініціалізації Клас(сhar\*, int);

• Конструктор копіювання Клас(const Клас&); •

Деструктор •

Методи доступу та закритих данних Get та Set.

#include <iostream>

using namespace std;

class **Food**

{

char \*Type;

int Calories;

public:

**Food**(){}

**Food**( char \* Type , int Calories)

{

this->Type = Type;

this->Calories = Calories;

}

**Food**( const Food& obj)

{

this->Type = obj.Type;

this->Calories = obj.Calories;

}

~**Food**()

{

cout << "Delete" << endl ;

delete Type;

}

char \***GetType**()

{

return this->Type;

}

void **SetType**(char \*Type)

{

this->Type = Type;

}

int **GetCalories**()

{

return this->Calories;

}

void **SetCalories**(int Calories)

{

this->Calories = Calories;

}

void **Print**()

{

cout << "Type:" << this->Type << endl;

cout << "Calories:" << this->Calories << endl;

}

void **Scan**()

{

cout << "Enter type:";

cin >> this->Type;

cout << "Enter calories:";

cin >> this->Calories;

}

};

int **main**()

{

Food food;

food.Scan();

food.Print();

Food f("test", 10);

f.Print();

Food a;

a.SetType("type");

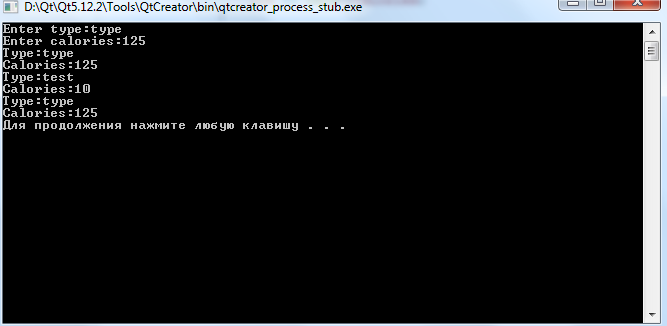
a.SetCalories(125);

a.Print();

system("pause");

return 0;

}



**Висновок:** На лабораторній роботі навчились використовувати конструктори і деструктори класів, створювати класи для опису лінійних списків.

ЛАБОРАТОРНА РОБОТА №3

**Тема:** Перевантаження операцій класу

**Мета:** Ознайомитись зі способами перевантаження операцій та навчитись використовувати їх при роботі з об’єктами.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Ознайомитись зі способами перевантаження операцій у С++.

**Завдання 2.** Проаналізувати приклад програми, яка оголошує клас та перевантажує низку операцій над точками у тривимірному просторі. Визначити результат її роботи.

**Завдання 3.** В класі Int, який розроблений в завданні №1 лабораторної роботи №1, перевизначте чотири цілочисельні арифметичні операції («+», «-», «\*» , «/») так, щоб їх можна було використовувати для операцій з об'єктами класу Int. Якщо результат будь-якої з операцій виходить за межі типу int (в 32-бітній системі), що може мати значення від 2 147 483 648 до -2 147 483 648, то операція повинна послати повідомлення про помилку і завершити програму. Такі типи даних корисні там, де помилки можуть бути викликані арифметичним переповненням, яке неприпустимо. Напишіть програму для перевірки цього класу. Підказка: для полегшення перевірки переповнення виконуйте обчислення з використанням типу long double.

#include <iostream>

using namespace std;

class **Int**{

int value;

public:

**Int**()

{

value= 0;

}

**Int** (int \_value){

value= \_value;

}

Int operator +(Int& r2){

long double temp = r2.value + this->value;

if (temp <= 2147483648 && temp >= -2147483647){

return Int(temp);

}

else cout << "Fatal error" << endl;

}

Int operator -(Int& r2){

Int temp;

temp.value= this->value - r2.value;

return temp;

}

Int operator \*(Int& r2){

Int temp;

temp.value= r2.value \* this->value;

return temp;

}

Int operator /(Int& r2){

Int temp;

temp.value= this->value / r2.value;

return temp;

}

void **setzero**(){

value=0;

}

int **getzero**(){

return value;

}

void **set**(int a){

value=a;

}

void **summa**(Int a, Int b){

this->value = a.value +b.value;

}

void **print**(){

cout << "Value=" << this->value << endl;

}

};

int **main**()

{

setlocale(LC\_ALL, "ukr");

Int r1,r2,r3;

r1.set(15);

r2.set(24);

r3.setzero();

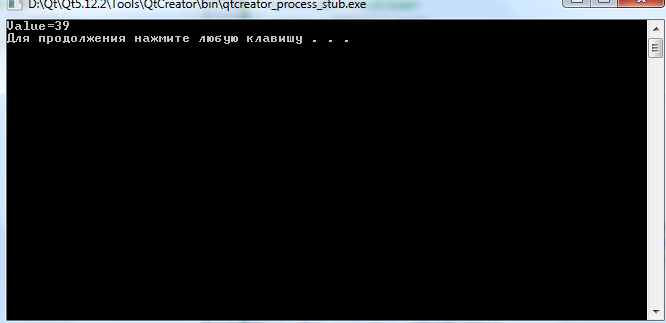
r3 = r1 + r2;

r3.print();

system("pause");

return 0;

}



**Висновок:** На лабораторній роботі ознайомились зі способами перевантаження операцій та навчились використовувати їх при роботі з об’єктами.

ЛАБОРАТОРНА РОБОТА №4

**Тема:** Успадковування класів

**Мета:** Ознайомитись зі способами та механізмами успадкування класів та навчитись використовувати їх для побудови об’єктно-орієнтованих програм

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Уявіть собі видавничу компанію, яка торгує книгами і аудіо-записами цих книг. Створіть клас publication, в якому зберігаються назва (рядок) і ціна (тип float) книги. Від цього класу успадковуються ще два класи: book, який містить інформацію про кількість сторінок у книзі (типу int), і type, який містить час запису книги у хвилинах (тип float). У кожному з цих трьох класів повинен бути метод getdata(), через який можна отримувати дані від користувача з клавіатури, і putdata(), призначений для виведення цих даних. Напишіть функцію main() програми для перевірки класів book і type. Створіть їх об'єкти в програмі і запросіть користувача ввести і вивести дані з використанням методів getdataQ і putdata().

#include <iostream>

#include <string>

using namespace std;

class **publication**

{

string name;

float price;

public:

void **putdata**()

{

cout << "Name: ";

cout << name << endl;

cout << "Price: " << price << endl;

}

void **getdata**()

{

cout << "Enter name - ";

cin >> name;

cout << "Enter price - ";

cin >> price;

}

};

class **book**:public publication

{

int pages;

public:

void **putdata**()

{

publication::putdata();

cout << "Pages:" << pages << endl;

}

void **getdata**()

{

publication::getdata();

cout << "Enter pages - ";

cin >> pages ;

}

};

class **type**:public publication

{

float minutes;

public:

void **putdata**()

{

publication::putdata();

cout << "Minutes:";

}

void **getdata**()

{

publication::getdata();

cout << "Enter minutes - ";

cin >> minutes;

}

};

int **main**()

{

book ObjP;

type ObjM;

ObjP.getdata();

ObjP.putdata();

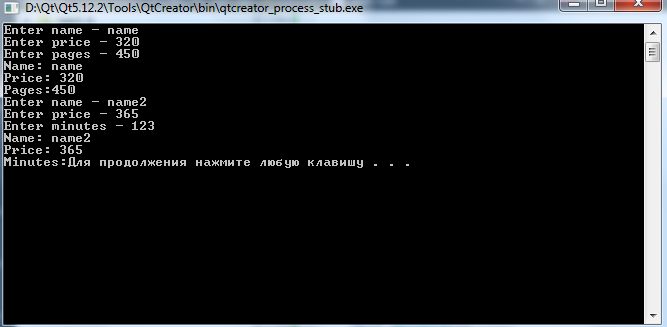
ObjM.getdata();

ObjM.putdata();

system("pause");

return 0;

}



**Завдання 2.** До класів з попереднього завдання (попередньо зберігши окремо код) додайте базовий клас sales, в якому міститься масив, що складається з трьох значень типу float, куди можна записати загальну вартість проданих книг за останні три місяці. Включіть в клас методи getdata() для отримання значень вартості від користувача і putdata() для виведення цих цифр. Змініть класи book і type так, щоб вони стали похідними обох класів: publication і sales. Об'єкти класів book і type повинні вводити і виводити дані про продажі разом з іншими своїми даними. Напишіть функцію main() для створення об'єктів класів book і type, щоб протестувати можливості введення/виведення даних.

#include <iostream>

#include <string>

using namespace std;

class **publication**

{

string name;

float price;

public:

void **putdata**()

{

cout << "Name: ";

cout << name << endl;

cout << "Price: " << price << endl;

}

void **getdata**()

{

cout << "Enter name - ";

cin >> name;

cout << "Enter price - ";

cin >> price;

}

};

class **sales**

{

float vart[2];

public:

void **putdata**()

{

cout << "Vartist za pershyy misyaz : " << vart[0] << endl;

cout << "Vartist za druhi misyaz: " << vart [1] << endl;

cout << "Vartist za tretii misyaz: " << vart[2] << endl;

}

void **getdata**()

{

for(int i=0; i<3; i++)

{

cout << "Enter vartist - ";

cin >> vart[i];

}

}

};

class **book**:public publication,public sales

{

int pages;

public:

void **putdata**()

{

publication::putdata();

cout << "Pages:" << pages << endl;

sales::putdata();

}

void **getdata**()

{

publication::getdata();

cout << "Enter pages - ";

cin >> pages ;

sales::getdata();

}

};

class **type**:public publication, public sales

{

float minutes;

public:

void **putdata**()

{

publication::putdata();

cout << "Minutes:";

sales::putdata();

}

void **getdata**()

{

publication::getdata();

cout << "Enter minutes - ";

cin >> minutes;

sales::getdata();

}

};

int **main**()

{

book ObjP;

type ObjM;

ObjP.getdata();

ObjP.putdata();

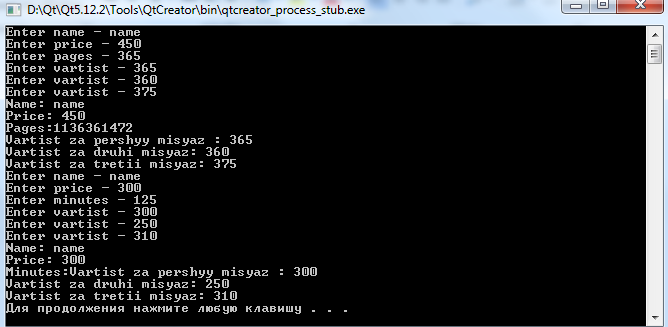
ObjM.getdata();

ObjM.putdata();

system("pause");

return 0;

}



**Завдання 2.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення. Створити похідний клас ПАРАЛЕЛЕПІПЕД з полями-сторонами. Визначити конструктори за замовчуванням і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення об’єму та площі поверхні.

#include <iostream>

using namespace std;

class **three\_numbers**

{

protected:

int a, b, c;

public:

**three\_numbers**();

void **getdata**()

{

cout << "Enter a - ";

cin >> a;

cout << "Enter b - ";

cin >> b;

cout << "Enter c - ";

cin >> c;

}

void **putdata**()

{

cout << "a: "<< a << endl;

cout << "b: " << b << endl;

cout << "c: " << c << endl;

}

~**three\_numbers**();

};

three\_numbers::**three\_numbers**()

{

cout << "Constructor" << endl;

}

three\_numbers::~**three\_numbers**()

{

cout << "Delete" << endl;

}

class **parallelepiped**:public three\_numbers

{

int S, V;

public:

**parallelepiped**();

void **getdata**()

{

three\_numbers::getdata();

}

void **putdata**()

{

three\_numbers::putdata();

}

void **Ploscha**()

{

S=2\*(b+b\*c+a\*c);

cout << "Ploscha = " << S << endl;

}

;

void **Obem**()

{

V=a\*b\*c;

cout << "Obem = " << V << endl;

}

~**parallelepiped**();

};

parallelepiped::**parallelepiped**()

{

cout << "Constructor" << endl;

}

parallelepiped::~**parallelepiped**()

{

cout << "Delete" << endl;

}

int **main**()

{

setlocale(LC\_ALL, "ukr");

parallelepiped obj;

obj.getdata();

obj.putdata();

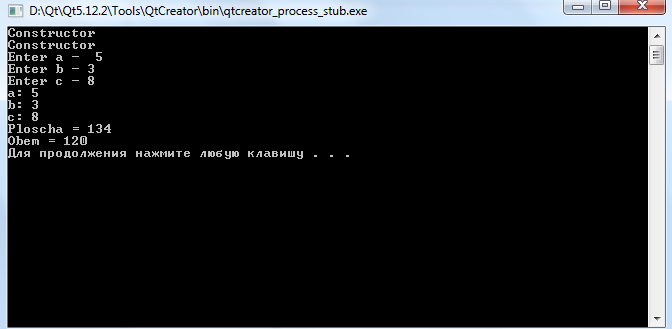
obj.Ploscha();

obj.Obem();

system("pause");

return 0;

}



**Висновок:** На лабораторній роботіознайомились зі способами та механізмами успадкування класів та навчились використовувати їх для побудови об’єктно-орієнтованих програм.

ЛАБОРАТОРНА РОБОТА №5

**Тема:** Тема: Віртуальні функції та поліморфізм.

**Мета:** Практично ознайомитись з поняттям поліморфізму, його застосуванням та вивчити механізм його реалізації за допомогою віртуальних функцій

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 2.** Взявши за основу програму із завдання 1, додайте метод типу bool, який називається isOveersize (), до класів book і tape. Припустимо, книга, в якій більше 800 сторінок, або касета з часом програвання якої більше 90 хвилин, будуть вважатися об'єктами з перевищенням розміру. До цієї функції можна звертатися з main(), а результат її роботи виводити у вигляді рядка «Перевищення розміру!» Для відповідних книг і касет. Припустимо, об'єкти класів book і tape повинні бути доступні через покажчики на них, що зберігаються в масиві типу publication.

#include <iostream>

#include <string>

using namespace std;

class **Publication**{

public:

string name;

float price;

void **get\_info**(){

cout<<"Enter name: ";

cin>>name;

cout<<"Enter price: ";

cin>>price;

};

void **put\_info**(){

cout<<"Name: "<<name << endl;

cout<<"Price: "<<price << endl;

};

};

class **Book** :public Publication{

public:

int pages;

void **getdata**(){

cout<<"Enter page: ";

cin>>pages;

isOveersize();

};

void **putdata**(){

cout<<"Page: "<<pages<<endl;

};

bool **isOveersize**(){

if(pages > 800){

cout << "Exceeding the broad book" << endl;

exit(0);

}

};

};

class **Tape** :public Publication{

public:

int minutes;

void **getdata**(){

cout<<"Enter minutes: ";

cin>>minutes;

isOveersize();

};

void **putdata**(){

cout<<"Minutes : "<< minutes <<endl;

};

bool **isOveersize**(){

if(minutes > 90) {

cout << "Exceeding minutes" << endl;

exit(0);

}

};

};

int **main**(int argc, char const \*argv[])

{

setlocale(LC\_ALL, "ukr");

Book a;

Tape b;

Publication c;

int who\_books;

cout<<"Enter number page: ";

cin>>who\_books;

for (int i = 1; i <= who\_books; i++) {

cout<<endl;

cout<<"Number:"<<i<<endl;

c.get\_info();

a.getdata();

b.getdata();

cout<<endl;

cout<<"Результат:"<<i<<endl;

cout<<endl;

c.put\_info();

a.putdata();

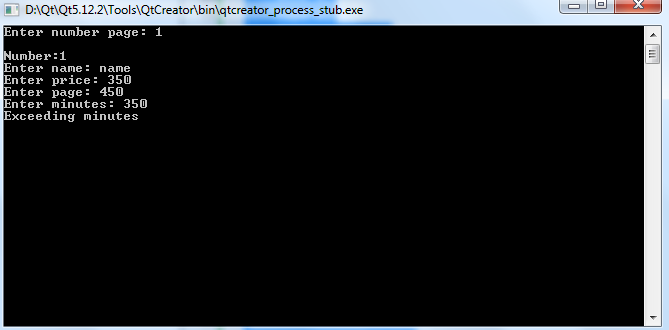
b.putdata();

};

system ("pause");

return 0;

}



ЛАБОРАТОРНА РОБОТА №6

**Тема:** Композиція об’єктів в ООП

**Мета:** Ознайомитись зі способами та механізмами.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Ознайомитись з теоретичним матеріалом. Розробіть клас Student (в окремих файлах student.h і student.cpp) із атрибутами: прізвище, ім’я, по батькові, номер залікової книжки, державник/платник (тип bool). Визначте для даного класу конструктор по замовчуванню, який буде запитувати у користувача дані для заповнення атрибутів об’єкта; параметризований конструктор; операцію виводу у потік. У головній функції виконайте перевірку функціонування методів класу створивши три об’єкти різними способами і вивівши їх на екран за допомогою оператора виводу у потік. Розробіть клас Grupa, який міститиме як атрибут назву групи (тип char \* або std::string), спеціальність і список студентів групи, студенти описуються за допомогою класу Student, який визначений у попередньому завданні. Визначте для даного класу всі можливі конструктори, деструктор, операції виводу в потік. Тип відношення між класами Grupa і Student – агрегація. Розробіть клас Facultet, який міститиме наступні атрибути: назву факультету (тип char \* або std::string) і список груп, групи описуються за допомогою класу Grupa, який визначений у попередньому завданні. Визначте для даного класу всі можливі конструктори, деструктор, операції виводу в потік. Тип відношення між класами Facultet і Grupa і Student – композиція . Оформіть звіт про виконання лабораторної роботи.

#include <iostream>

#include <string>

#define n 2

using namespace std;

class **Student**

{

protected:

string name, prizv, pobatk;

char s;

int number\_zalik;

bool derzhavnyk;

public:

**Student**()

{

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

{

cout << "Enter name - "<< endl;

cin >> name;

cout << "Enter prizv - "<< endl;

cin >> prizv;

cout << "Enter pobatk - "<< endl;

cin >> pobatk;

cout << "Enter number zalik - "<< endl;

cin >> number\_zalik;

cout << "Srudent derzhavnyk? Y/N"<< endl;

cin >> s;

char s;

s=getchar();

if(s=='Y'){

derzhavnyk=true;}

else

derzhavnyk=false;

}

}

**Student**(string \_name, string \_prizv, string \_pobatk, int \_number\_zalik)

{

name = \_name;

prizv = \_prizv;

pobatk = \_pobatk;

number\_zalik = \_number\_zalik;

}

void **Vyvid**()

{

cout << "Student name - " << name<< endl;

cout << "Student prizv - " << prizv<< endl;

cout << "Student pobatk - " << pobatk<< endl;

cout << "Student number zalik -" << number\_zalik<< endl;

cout << "Student status - " << derzhavnyk<< endl;

}

}a;

class **Grupa**:public Student

{

protected:

string grupa, spezialnist, spysok;

public:

**Grupa**()

{

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

{

cout << "Enter grupa - "<< endl;

cin >> grupa;

cout << "Enter spezialnist - "<< endl;

cin >> spezialnist;

cout << "Enter spysok - "<< endl;

cin >> spysok;}

}

void **Vyvid**()

{

Student::Vyvid();

cout << "Grupa - " << grupa << endl;

cout << "Spezialnist - " << spezialnist<< endl;

cout << "Spysok grupa - " << spysok<< endl;

}

~**Grupa**()

{

cout << "Delete object" << endl;

}

}b;

class **Facultet**:public Student, public Grupa

{

protected:

string facult;

public:

**Facultet**()

{

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

{

cout << "Enter facultet - "<< endl;

cin >> facult;}

}

void **Vyvid**()

{

Grupa::Vyvid();

cout << "Facultet - "<< endl;

}

~**Facultet**()

{

cout << "Delete" << endl;

}

}c;

int **main**()

{

Student();

a.Vyvid();

Grupa();

b.Vyvid();

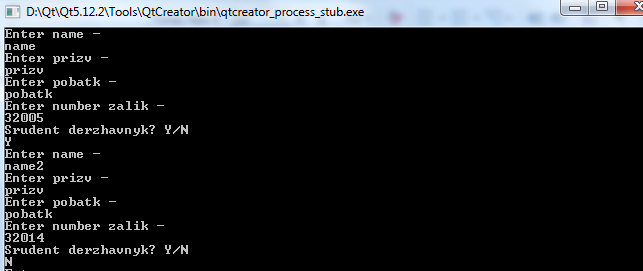
Facultet();

c.Vyvid();

system("pause");

return 0;

}



**Висновок:** На лабораторній роботі ознайомились зі способами та механізмами.

ЛАБОРАТОРНА РОБОТА №7

**Тема:** Шаблони функцій і класів

**Мета:** Навчитись створювати і використовувати шаблонні функції і класи

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Напишіть шаблон функції, що повертає середнє арифметичне всіх елементів масиву. Аргументами функції повинні бути ім'я і розмір масиву (типу int). У функції main() перевірте роботу шаблонної функції з масивами типу int, long, double і char.

#include <iostream>

template<typename T, size\_t N> inline T **avg**(const T (&data)[N])

{

T sum = T();

for(size\_t i = 0; i < N; ++i)

sum += data[i];

return sum / N;

}

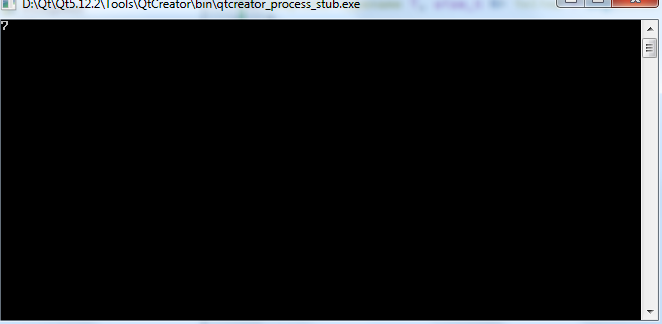
int **main**()

{

double a[] = {6, 8, 7, 5, 9};

std::cout << avg(a) << std::endl;

}



**Завдання 2.** Створіть функцію аmaх(), що повертає значення максимального елемента масиву. Аргументами функції повинні бути адреса і розмір масиву. Зробіть з функції шаблон, щоб вона могла працювати з масивом будь-якого числового типу. Напишіть функцію main(), в якій перевірите роботу функції з різними типами масивів.

#include <iostream>

using namespace std;

template <typename T>

T **amax**(const T \* array, int count)

{

int max = array[0];

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

{

if(array[i]>max){

max = array[i];

}

}

cout <<"Max:" << max;

cout << endl;

}

int **main**()

{

const int iSize = 10, dSize = 7, fSize = 10;

int iArray[iSize] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

double dArray[dSize] = {1.2345, 2.234, 3.57, 4.67876, 5.346, 6.1545, 7.7682};

float fArray[fSize] = {1.34, 2.37, 3.23, 4.8, 5.879, 6.345, 73.434, 8.82, 9.33, 10.4};

cout << "\t\t MAX element massiva\n\n";

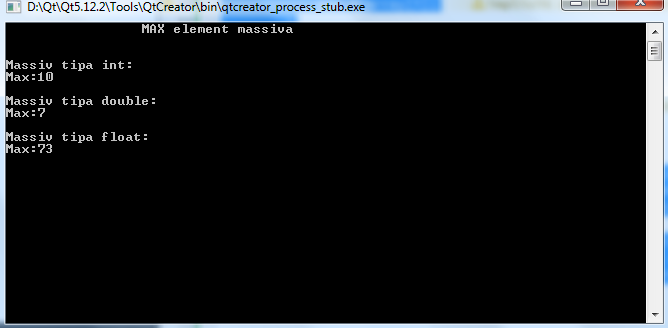
cout << "\nMassiv tipa int:\n"; amax(iArray, iSize);

cout << "\nMassiv tipa double:\n"; amax(dArray, dSize);

cout << "\nMassiv tipa float:\n"; amax(fArray, fSize);

return 0;

}



**Висновок:** На лабораторній роботі навчились створювати і використовувати шаблонні функції і класи.

ЛАБОРАТОРНА РОБОТА №9

**Тема:** Вивчення механізмів оброблення виняткових ситуацій

**Мета:** Навчитись обробляти ситуації появлення виняткових ситуацій, вивчити механізми їх оброблення.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

**Завдання 1.** Додайте клас винятків до програми ARROVER таким чином, щоб індекси, що виходять за межі масиву, викликали генерацію винятку. Блокпастка catch може виводити користувачеві повідомлення про помилку.

#include <iostream>

using namespace std;

#include <process.h> //for exit()

const int LIMIT = 100; //array size

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

class **safearay**

{

private:

int arr[LIMIT];

public:

int& operator [](int n)

{

if( n< 0 || n>=LIMIT )

{ cout << "\nIndex out of bounds"; exit(1); }

return arr[n];

}

};

int **main**()

{

safearay sa1;

int j;

try{

for(int j=0; j<LIMIT; j++)

sa1[j] = j\*10;

if(j>100) throw 100;

cout << "Error";}

catch(int c) {

cout << "Error = " << c << endl;

}

for( j=0; j<LIMIT; j++) //display elements

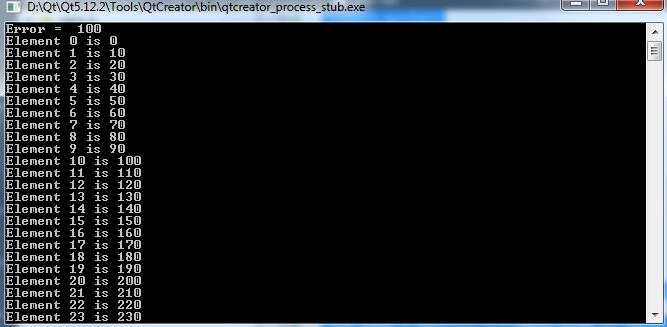
{

int temp = sa1[j]; //\*right\* side of equal sign

cout << "Element " << j << " is " << temp << endl;

}

return 0;}



**Завдання 2.** Модифікуйте програму з попереднього завдання таким чином, щоб в повідомлення про помилку входила інформація про значення індексу, який призвів до збою.

#include <iostream>

using namespace std;

#include <process.h> //for exit()

const int LIMIT = 100; //array size

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

class **safearay**

{

private:

int arr[LIMIT];

public:

int& operator [](int n)

{

if( n< 0 || n>=LIMIT )

{ cout << "\nIndex out of bounds"; exit(1); }

return arr[n];

}

};

int **main**()

{

safearay sa1;

int j;

try{

for(int j=0; j<LIMIT; j++)

sa1[j] = j\*10;

if(j>100) throw 100;

cout << "Error";}

catch(int c) {

cout << "Error = " << c << endl;

cout << "Element 100 is 1000 " << endl;

}

for( j=0; j<LIMIT; j++) //display elements

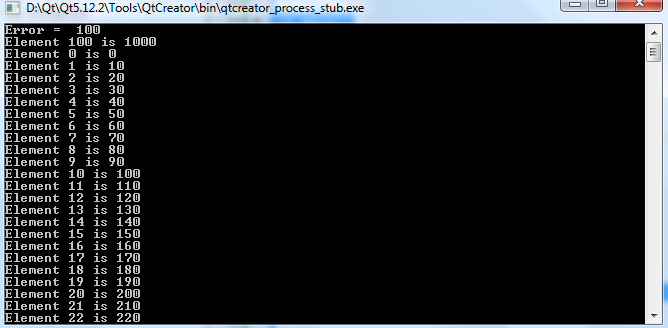
{

int temp = sa1[j]; //\*right\* side of equal sign

cout << "Element " << j << " is " << temp << endl;

}

return 0;}



**Висновок:** На лабораторній роботі навчились обробляти ситуації появлення виняткових ситуацій, вивчили механізми їх оброблення.

ЛАБОРАТОРНА РОБОТА №10

**Тема:** Робота з файлами

**Мета:** Навчитись зчитувати і записувати данні у текстові і двійкові файли за допомогою файлових потоків.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Ознайомитись зі способами читання/запису даних.

2. Написати програму, яка записуватиме ваші дані у текстовий файл «myfile.txt» (П.І.П., вік, номер телефону, дата народження, місце навчання). Дані для запису можна або зчитувати з консолі при виконанні програми або прописати у коді програми.

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int **main**()

{

setlocale(LC\_ALL, "ukr");

string tempstr;

unsigned long int tempint;

ofstream file("prakt.txt");

while (true)

{

cout << "Enter name or 'end' to exit: ";

cin >> tempstr;

if (tempstr == "end")

break;

file << tempstr << ",";

cout << "Enter prizv: ";

cin >> tempstr;

file << tempstr << ",";

cout << "Enter name: ";

cin >> tempstr;

file << tempstr << ",";

cout << "Enter pobatk: ";

cin >> tempstr;

file << tempstr << ",";

cout << "Enter years: ";

cin >> tempint;

file << tempstr << ",";

cout << "Enter phone number: ";

cin >> tempstr;

file << tempstr << ",";

cout << "Enter date of birth: ";

cin >> tempstr;

file << tempstr << ",";

cout << "Enter place of study: ";

cin >> tempstr;

file << tempint << "\n";

}

file.close();

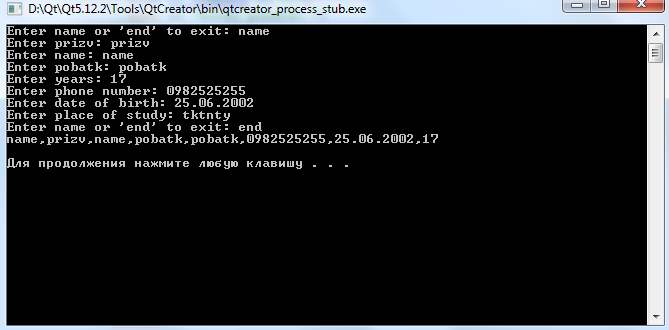
ifstream ffile("prakt.txt");

cout << ffile.rdbuf() << endl;

system("pause");

return 0;

}



**Висновок:** На лабораторній роботі навчились зчитувати і записувати данні у текстові і двійкові файли за допомогою файлових потоків.

ЛАБОРАТОРНА РОБОТА №11

**Тема:** Дослідження контейнерних класів бібліотеки STL

**Мета:** дослідити контейнерні класи vector та list бібліотеки STL, набути навичок їх використання.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. З допомогою контейнера std::vector створити вектор значень типу char, в який записати український алфавіт (малими літерами).

2. Усі голосні букви у векторі замінити прописними (за допомогою циклу).

3. Вивести вміст вектора на екран.

#include<iostream>

#include<vector>

using namespace std;

int **main**()

{

vector<char> vec;

for(int i=0; i<26; ++i){

vec.push\_back('a'+i);

}

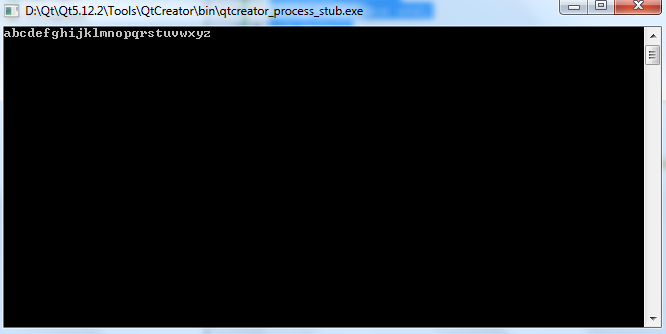
for(int i=0; i<26; ++i){

cout << vec[i];

}

return 0;

}



4. З допомогою контейнера list створити список значень типу char, в який записати по буквах своє прізвище. Вивести вміст списку на екран.

5. Відсортувати список (від А до Я). У консоль вивести повідомлення: «Список сортується!».

6. Вивести вміст списку на екран.

#include<iostream>

#include<list>

#include <iterator>

using namespace std;

int **main**()

{

list <char> l ;

l.push\_back('L');

l.push\_back('a');

l.push\_back('z');

l.push\_back('a');

l.push\_back('r');

l.push\_back('o');

l.push\_back('v');

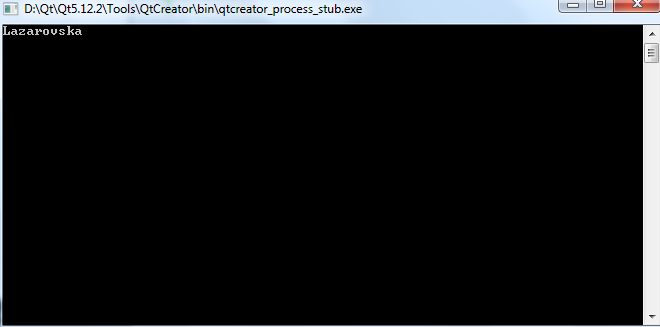
l.push\_back('s');

l.push\_back('k');

l.push\_back('a');

copy(l.begin(), l.end(), ostream\_iterator<int>(*cout*," "));

}



7. Створити вектор об’єктів класу, створеного згідно індивідуального завдання №1 в лабораторній роботі №2. Заповнити вектор десятьма об’єктами.

8. Здійснити вивід значень об’єктів за допомогою індексу вектора.

9. Здійснити вивід значень об’єктів за допомогою ітераторів.

10. \* З допомогою контейнера list створити список студентів групи. В контейнер поміщати об’єкти класу, в якому представлені такі атрибути: прізвище, ім’я, по батькові, вік, номер телефону, а також реалізовані сетери даних атрибутів, і перевизначена операція виводу, яка буде виводити значення атрибутів на екран.