### МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний Університет “Запорізька політехника ”**

**Кафедра програмних засобів**

### 

### Звіт

### З “Об’єктно-орієнтованого програмування”

**ЛАБОРАТОРНА РОБОТА №2**

**ДИНАМІЧНІ КЛАСОВІ ТИПИ**

Виконував студент (КНТ 119) О.К. Спесивцев

Прийняв доцент Н.О. Миронова

**Мета роботи:**

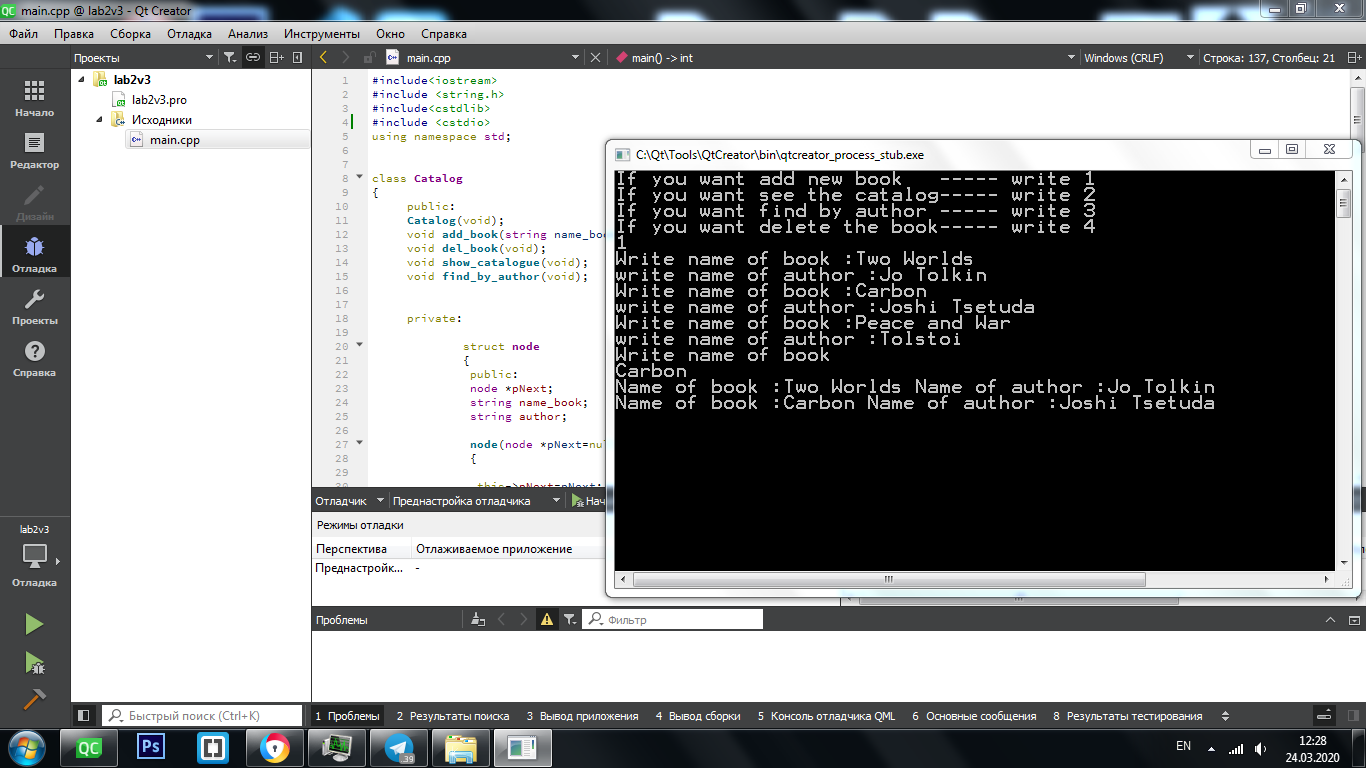
Навчитись використовувати динамічні класи при створенні програм.

**Завдання:**

Варіант 1. Створити динамічний клас Catalog, що базується на зв’язаному списку, де кожний елемент списку – структура типа Book. Клас повинен містити наступні операції: add\_book() – додавання книги до каталогу;

del\_book() – видалення книги з каталогу;

find\_by\_autor() – пошук книги у каталозі за автором; show\_cataloque() - відображення каталогу книг.



1

Заголовочный файл:

#ifndef CATALOG\_H

#define CATALOG\_H

#include<iostream>

#include <string.h>

#include<cstdlib>

#include <cstdio>

using namespace std;

class **Catalog**

{

public:

**Catalog**(void);

void **add\_book**(string name\_book,string author );

void **del\_book**(void);

void **show\_catalogue**(void);

void **find\_by\_author**(void);

private:

struct **node**

{

public:

node \*pNext;

string name\_book;

string author;

**node**(node \*pNext=nullptr, string name\_book ="",string author="")

{

this->pNext=pNext;

this->name\_book=name\_book;

this->author=author;

}

};

node \*head;

int size;

};

#endif // CATALOG\_H

2

Файл описания методов:

#include "catalog.h"

#include <iostream>

#include <string.h>

#include <cstdlib>

#include <cstdio>

using namespace std;

Catalog::**Catalog**()

{

this->head=nullptr;

this->size=0;

}

void Catalog::**add\_book**(string name\_book,string author)

{

node \*temp=new node (nullptr,name\_book,author);

if(this->head==nullptr)

{

this->head=temp;

}

else

{

node \*last=head;

while(last->pNext!=nullptr)

{

last=last->pNext;

}

last->pNext=temp;

}

this->size++;

}

void Catalog::**show\_catalogue**()

{

node \*last=head;

while(last!=nullptr)

{

cout<<"Name of book :"<<last->name\_book<<" "<<"Name of author :"<<last->author<<endl;

last=last->pNext;

}

}

**3**

void Catalog::**find\_by\_author**()

{

string aut;

cout<<"write name of author"<<endl;

cin>>aut;

node \*last=head;

while(last!=nullptr)

{

if(aut==last->author)

{

cout<<"Name of book :"<<last->name\_book<<" "<<"Name of author :"<<last->author<<endl;

}

last=last->pNext;

}

}

void Catalog::**del\_book**()

{

node \*last=head;

node \*temp=head;

string name;

cout<<"Write name of book"<<endl;

cin>>name;

while(temp!=nullptr)

{

if(name==last->name\_book)

{

last=last->pNext;

temp->pNext=last->pNext;

break;

}

last=last->pNext;

temp=temp->pNext;

}

}

**Main File:**

#include<iostream>

#include <string.h>

#include<cstdlib>

#include <cstdio>

#include "catalog.h"

using namespace std;

int **main**(void)

{

Catalog a;

int b;

string aut,name;

cout<<"If you want add new book ----- write 1"<<endl;

cout<<"If you want see the catalog----- write 2"<<endl;

cout<<"If you want find by author ----- write 3"<<endl;

cout<<"If you want delete the book----- write 4"<<endl;

cin>>b;

cin.ignore();

**4**

if(b==1)

{

cout<<"Write name of book :";

getline(*cin*,*name*);

cout<<"write name of author :";

getline(*cin*,*aut*);

a.add\_book(name,aut);

}

else if(b==2)

{

a.show\_catalogue();

}

else if(b==3)

{

a.find\_by\_author();

}

else if(b==4)

{

a.del\_book();

}

}

5