**Лабораторна робота №6**

Тема: Композиція об’єктів в ООП

Мета: Ознайомитись зі способами та механізмами ООП.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Ознайомитись з теоретичним матеріалом.

Ознайомився з теоретичним матеріалом.

1. Розробіть клас Student (в окремих файлах student.h і student.cpp) ізатрибутами: 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.

studednt.h

#ifndef STUDENT\_H

#define STUDENT\_H

#include<string>

#include<iostream>

using namespace std;

class **Student**

{

private:

string name;

string surName;

string fatherName;

int numberMarkBook;

bool studyType;

public:

**Student**(string,string,string,int,bool);

**Student**();

friend ostream& operator<<(ostream& os, Student& dt);

};

#endif // STUDENT\_H

studednt.cpp

#include "student.h"

Student::**Student**(string \_name,string \_surnmae,string \_fathername,int \_number,bool \_type){

name=\_name;

surName=\_surnmae;

fatherName=\_fathername;

numberMarkBook=\_number;

studyType=\_type;

}

Student::**Student**()

{

cout<<"Name-";

cin>>name;

cout<<"Surname-";

cin>>surName;

cout<<"Father name-";

cin>>fatherName;

cout<<"Number Mark Book <-";

cin>>numberMarkBook;

cout<<"studytype(1-derzh,0-platnyk)-";

cin>>studyType;

}

ostream& operator<<(ostream& os, Student& dt)

{

os << dt.name << ' ' << dt.surName << ' ' << dt.fatherName<<' '<<dt.numberMarkBook<<' '<<dt.studyType<<endl;

return os;

}

3. 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.

group.h

#ifndef GROUP\_H

#define GROUP\_H

#include<string>

#include<iostream>

#include<student.h>

using namespace std;

class **Group**

{

private:

string name;

Student \*students;

int count;

public:

**Group**();

~**Group**();

friend ostream& operator<<(ostream& os, Group& dt);

};

#endif // GROUP\_H

gropup.cpp

#include "group.h"

Group::**Group**()

{

cout<<"name group-";

cin>>name;

cout<<"enter count students-";

cin>>count;

students=new Student[count];

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

students[i];

}

}

Group::~**Group**(){delete students;}

ostream& operator<<(ostream& os, Group& dt)

{

os << dt.name<<endl;

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

os<<dt.students[i];

return os;

}

4. 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.

facultet.h

#ifndef FACULTET\_H

#define FACULTET\_H

#include<group.h>

#include<student.h>

#include<string>

using namespace std;

class **Facultet**

{

private:

string name;

Group \*groups;

int count;

public:

**Facultet**();

~**Facultet**();

friend ostream& operator<<(ostream& os, Facultet& dt);

};

#endif // FACULTET\_H

facultet.cpp

#include "facultet.h"

Facultet::**Facultet**()

{

cout<<"enter facultet-";

cin>>name;

cout<<"enter count groups in facultet-";

cin>>count;

groups=new Group[count];

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

groups[i];

}

Facultet::~**Facultet**(){delete groups;}

ostream& operator<<(ostream& os, Facultet& dt)

{

os<<dt.name<<" ";

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

os<<dt.groups[i];

return os;

}

main.cpp

#include <iostream>

#include<student.h>

#include <group.h>

#include<facultet.h>

using namespace std;

int **main**()

{

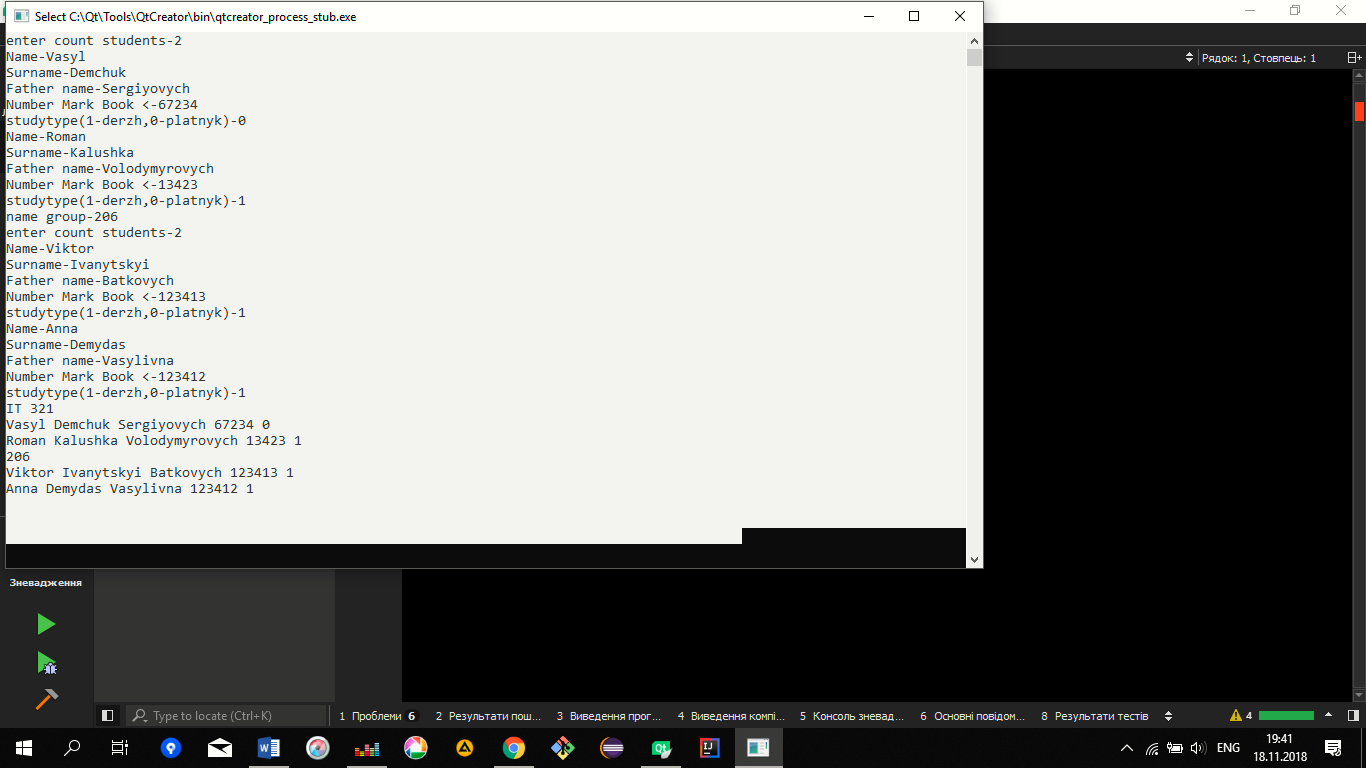
Facultet a;

cout<<a;

return 0;

}

Вивід програми



Висновок: ознайомився зі способами та механізмами композиції в ООП.