**Лабораторна робота №6**

Тема: Композиція об’єктів в ООП

Мета: ознайомитись із способами та механізмами об’єктної композиції в ООП.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Ознайомитись з теоретичним матеріалом.

2. 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.

Код програми:

student.h

#ifndef STUDENT\_H #define STUDENT\_H #include<string>   
#include<iostream>   
  
  
class **Student**   
{   
private:   
 std::string m\_name;   
 std::string m\_familyName;   
 std::string m\_fatherName;   
 int m\_numberMarkBook;   
 bool m\_studyType;   
public:   
 **Student**(std::string,std::string,std::string,int,bool);   
 **Student**();   
 *friend* std::ostream& *operator*<<(std::ostream& os, Student& dt);   
};   
#endif *//* *STUDENT\_H*

Student.cpp

#include "student.h"   
Student::**Student**(std::string name,std::string familyName,std::string fathername,int number,bool type) :m\_name(name), m\_familyName(familyName), m\_fatherName(fathername), m\_numberMarkBook(number), m\_studyType(type){}   
Student::**Student**()   
{   
 std::cout<<"Name-";   
 std::cin>>m\_name;   
 std::cout<<"Familyname-";   
 std::cin >>m\_familyName;   
 std::cout<<"Father name-";   
 std::cin>>m\_fatherName;   
 std::cout<<"Number Mark Book <-";   
 std::cin>>m\_numberMarkBook;   
 std::cout<<"studytype(1-derzhavnyk,0-platnyk)-";   
 std::cin>>m\_studyType;   
}   
std::ostream& *operator*<<(std::ostream& os, Student& dt) { os << dt.m\_name << ' ' << dt.m\_familyName << ' ' << dt.m\_fatherName<<' '<<dt.m\_numberMarkBook<<' '<<dt.m\_studyType<<std::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 із кардинальністю 0..01 – 1.

group.h

#ifndef GROUP\_H   
#define GROUP\_H   
#include<string>   
#include<iostream>   
#include<student.h>   
  
  
class **Group**   
{   
private:   
std::string m\_name;   
Student \*m\_students;   
int m\_count;   
public:   
Group();   
~**Group**();   
friend std::ostream& *operator*<<(std::ostream& os, Group& dt);   
};   
#endif *//* *GROUP\_H*

group.cpp

#include "group.h"   
Group::**Group**()   
{   
 std::cout<<"name group-";   
 std::cin>>m\_name;   
 std::cout<<"enter count students-";   
 std::cin>>m\_count;   
 m\_students=*new* Student[m\_count];   
 *for*(int i=0;i<m\_count;i++){   
 m\_students[i];   
 } }   
Group::~**Group**(){*delete* m\_students;}   
std::ostream& *operator*<<(std::ostream& os, Group& dt)   
{   
 os << dt.m\_name<<std::endl;   
 *for*(int i =0;i<dt.m\_count;i++)   
 os<<dt.m\_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 – композиція із кардинальністю 1 – 1..\*.

Код програми:

facultet.h

#ifndef FACULTET\_H   
#define FACULTET\_H   
#include<group.h>   
#include<student.h>   
#include<string>   
  
class **Facultet**   
{   
private:   
 std::string m\_name;   
 Group \*m\_groups;   
 int m\_count;   
public:   
 **Facultet**();   
 ~**Facultet**();   
 *friend* std::ostream& *operator*<<(std::ostream& os, Facultet& dt);   
}; #endif

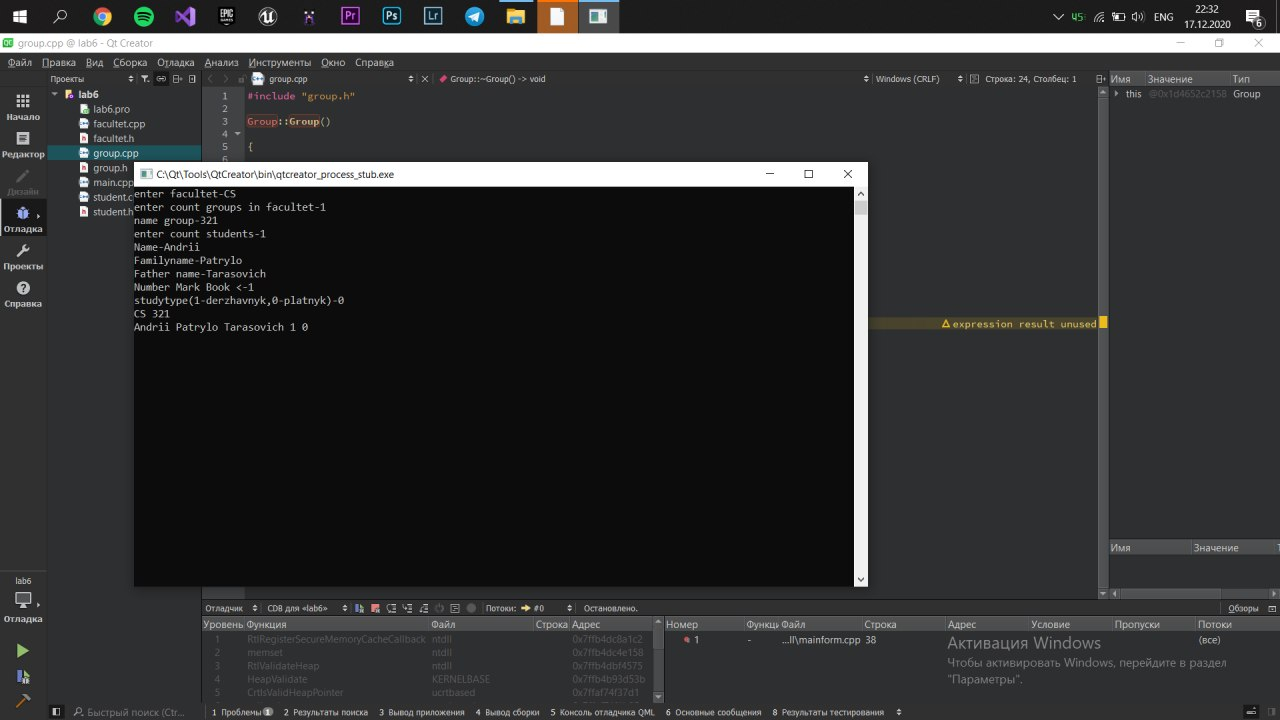
facultet.cpp

#include "facultet.h"   
Facultet::**Facultet**()   
{   
 std::cout<<"enter facultet-";   
 std::cin>>m\_name;   
 std::cout<<"enter count groups in facultet-";   
 std::cin>>m\_count;   
 m\_groups=*new* Group[m\_count];   
 *for*(int i=0;i<m\_count;i++)   
 m\_groups[i];   
}   
Facultet::~**Facultet**(){*delete* m\_groups;}   
std::ostream& *operator*<<(std::ostream& os, Facultet& dt)   
{   
 os<<dt.m\_name<<" ";   
 *for*(int i=0;i<dt.m\_count;i++)   
 os<<dt.m\_groups[i];   
 *return* os;   
}

Main.cpp

#include<facultet.h>   
int **main**() { Facultet a; std::cout << a;   
  
}

Результат виконання програми:



Висновок:Я ознайомилась із способами та механізмами об’єктної композиції в ООП.