

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 fathename, int
number, bool type) :m_name(name), m_familyName(familyName), m_fatherName(fathename),
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:

```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.  
enter facultet-C  
enter count groups in facultet-2  
name group-KN  
enter count students-2  
Name-Anhelina  
Familyname-Makarchuk  
Father name-Victor  
Number Mark Book <-1  
studytype(1-derzhavnyk,0-platnyk)-1  
Name-DDDDD  
Familyname-GGGG  
Father name-SSSS  
Number Mark Book <-2  
studytype(1-derzhavnyk,0-platnyk)-1  
name group-KI  
enter count students-1  
Name-01  
Familyname-Ko  
Father name-Va  
Number Mark Book <-3  
studytype(1-derzhavnyk,0-platnyk)-1  
C KN  
Anhelina Makarchuk Victor 1 1  
DDDDD GGGG SSSS 2 1  
KI  
01 Ko Va 3 1
```

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