**Лабораторна робота №6**

**Тема**: Композиція об’єктів в ООП

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

**Завдання .**

1. 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.

2. 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..\*

3. 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**

#include <iostream>

#pragma once

#include <string>

using namespace std;

class **Student**

{

string name;

string surname;

string thirdName;

string gradeBook;

bool isFree;

public:

**Student**();

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

void **set**(string,string,string,string,bool);

friend ostream& operator << (ostream&, const Student&);

};

Student::**Student**()

{

cout << "Enter full name" << endl;

cin >> name >> surname >> thirdName;

cout << "Enter gradeBook number" << endl;

cin >> gradeBook;

char type;

cout << "Press y if student is statesman" << endl;

cin >> type;

isFree = type =='y';

}

void Student::**set**(string name, string surname, string thirdName, string gradeBook, bool isFree)

{

this->name = name;

this->surname = surname;

this->thirdName = thirdName;

this->gradeBook = gradeBook;

this->isFree = isFree;

}

Student::**Student**(string name, string surname, string thirdName, string gradeBook, bool isFree)

{

set(name,surname,thirdName,gradeBook,isFree);

}

ostream& operator << (ostream& os,const Student& stud)

{

os << stud.name << ' ' << stud.surname << ' ' << stud.thirdName << endl;

os << stud.gradeBook << endl;

os << (stud.isFree ? string("is statesman?"): string("no")) << endl;

return os;

}

class **Group**

{

string name;

string speciality;

int studentCount;

Student \*\*students;

void **set**();

public:

**Group**(string name, string speciality, int studentCount);

**Group**();

friend ostream& operator << (ostream&, const Group&);

~**Group**();

};

Group::**Group**(string name, string speciality, int studentCount)

{

this->name = name;

this->speciality = speciality;

this->studentCount= studentCount;

set();

void Group::set()

{

if (studentCount < 1) studentCount =0;

else students = new Student\*[studentCount]();

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

{

students[i] = new Student();

}

}

ostream& operator << (ostream& os,const Group& group)

{

os << "Group: " << group.name << endl;

os << "Speciality: " << group.speciality << endl;

os << "StudentCount: " << group.studentCount;

for(int i = 0;i<group.studentCount;i++)

os << \*group.student[i];

return os;

}

}

class **Facultet**

{

string name;

int facultetCount;

Group \*\*groups;

void **set**();

public:

**Facultet**();

**Facultet**(string name);

friend ostream& operator << (ostream&,const Facultet&);

~**Facultet**();

};

Facultet::**Facultet**()

{

cout << "Enter facultet name" << endl;

cin >> name;

cout << "Enter facultet count" << endl;

cin >> facultetCount;

set();

}

int **main**()

{

Student \*t1 = new Student();

Student t2;

Student t3("Kot","Andrei","Stepanovich","KH-221",false);

cout << \*t1;

cout << t2;

cout << t3;

Group f2;

f2.set();

f2.name;

f2.students;

f2.speciality;

f2.studentCount;

Facultet n4;

n4.Facultet();

n4.set();

n4.name;

n4.groups;

n4.facultetCount;

return 0;

}