

Лабораторна робота №6

Тема: Композиція об'єктів в ООП

Мета: ознайомитись зі способами та механізмами.

Завдання 1.

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

Studentt.h

```
#pragma once
#include <iostream>
#include <string>
using namespace std;

class Student
{
private:
    string lastName;
    string firstName;
    string surName;
    int numberZalikBook;
    bool statesman;
    bool payer;
public:
    Student();
    Student(string lastName, string firstName, string surName, int numberZalikBook,
    bool statesman, bool payer);
    friend ostream& operator <<(ostream&, Student&);
};
```

Studentt.cpp

```

#pragma once
#include "Student.h"

Student::Student()
{
    cout << endl<< "ENTER INFORMATION ABOUT THE STUDENT " << endl<<endl;
    cout << "Input lastName "; cin >> lastName; cout << endl;
    cout << "Input firstName "; cin >> firstName; cout << endl;
    cout << "Input surName "; cin >> surName; cout << endl;
    cout << "Input numberZalikBook "; cin >> numberZalikBook; cout << endl;
    cout << "Input statesmen (1->true, 0->false) "; cin >> statesman; cout << endl;
    cout << "Input payer (1->true, 0->false) "; cin >> payer; cout << endl;
}

Student::Student(string lastName, string firstName, string surName, int numberZalikBook,
bool statesman, bool payer)
{
    this->lastName = lastName;
    this->firstName = firstName;
    this->surName = surName;
    this->numberZalikBook = numberZalikBook;
    this->statesman = statesman;
    this->payer = payer;
}

ostream & operator<<(ostream & s, Student & object)
{
    cout << "DATA ABOUT THE STUDENT !!!" << endl; cout << "1)";
    s << object.lastName; cout << endl; cout << "2)";
    s << object.firstName; cout << endl; cout << "3)";
    s << object.surName; cout << endl; cout << "4)Number = ";
    s << object.numberZalikBook; cout << endl; cout << "5)statesman ";
    if (object.statesman == 1) { cout << "->TRUE"<<endl; } else { cout << "-
>FALSE"<<endl; }
    cout << "5)payer ";
    if (object.payer == 1) { cout << "->TRUE" << endl; } else { cout << "->FALSE" <<
endl; }
    return s;
}

```

Main.cpp

```

#include <iostream>
#include "Student.h"
using namespace std;

int main()
{
    Student *Sasha = new Student;
    cout << *Sasha;
    cout << "-----" << endl << endl;
    Student Nazar;
    cout << Nazar;
    cout << "-----" << endl << endl;
    Student Ivan = Student();
    cout << Ivan;
}

```

```
}  
    return 0;  
}
```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

```
Select Microsoft Visual Studio Debug Console  
ENTER INFORMATION ABOUT THE STUDENT  
  
Input lastName Balko  
  
Input firstName Sasha  
  
Input surName Ivanovych  
  
Input numberZalikBook 18  
  
Input statesmen (1->true, 0->>false) 1  
  
Input payer (1->true, 0->>false) 0  
  
DATA ABOUT THE STUDENT !!!  
1)Balko  
2)Sasha  
3)Ivanovych  
4)Number = 18  
5)statesman ->TRUE  
5)payer ->FALSE  
-----  
  
ENTER INFORMATION ABOUT THE STUDENT  
  
Input lastName Golinka  
  
Input firstName Sergii  
  
Input surName Petrovich  
  
Input numberZalikBook 10  
  
Input statesmen (1->true, 0->>false) 0  
  
Input payer (1->true, 0->>false) 1  
  
DATA ABOUT THE STUDENT !!!  
1)Golinka  
2)Sergii  
3)Petrovich  
4)Number = 10  
5)statesman ->FALSE  
5)payer ->TRUE
```

```

ENTER INFORMATION ABOUT THE STUDENT

Input lastName Tataryn

Input firstName Oleg

Input surName Olegovych

Input numberZalikBook 17

Input statesmen (1->true, 0->false) 1

Input payer (1->true, 0->false) 0

DATA ABOUT THE STUDENT !!!
1)Tataryn
2)Oleg
3)Olegovych
4)Number = 17
5)statesman ->TRUE

```

Завдання 2.

3. Розробіть клас Група, 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 `Група` і `Student` – агрегація.

КОД ПРОГРАМИ

```

#include <iostream>
#include <string>
using namespace std;
class Student;

class Student
{
private:
    string lastName;
    string firstName;
    string surName;
    int numberZalikBook;
    bool statesman;
    bool payer;
public:
    friend ostream& operator <<(ostream&, Student&);

```

```

Student()
{
    cout << "Input lastName "; cin >> lastName; cout << endl;
    cout << "Input firstName "; cin >> firstName; cout << endl;
}

Student(string lastName, string firstName, string surName, int numberZalikBook,
bool statesman, bool payer)
{
    this->lastName = lastName;
    this->firstName = firstName;
    this->surName = surName;
    this->numberZalikBook = numberZalikBook;
    this->statesman = statesman;
    this->payer = payer;
}

};

ostream & operator<<(ostream & s, Student & object)
{
    cout << endl;
    s << object.lastName; cout << " ";
    s << object.firstName; cout << endl;;

    return s;
}

class Grupa
{
    string nameGroup;
    string specialty;
public:
    Student *st1;
    Student *st2;
    Student *st3;
    Student *st4;
    Student *st5;

    Grupa(string nameGroup, string specialty, Student & student1, Student & student2,
Student &student3, Student & student4, Student &student5)
    {
        this->nameGroup = nameGroup;
        this->specialty = specialty;
        this->st1 = &student1;
        this->st2 = &student2;
        this->st3 = &student3;
        this->st4 = &student4;
        this->st5 = &student5;
    }

    friend ostream & operator << (ostream & o, Grupa & g);

    void printObject()
    {
        cout << *st1;
        cout << *st2;
    }
}

```

```

        cout << *st3;
        cout << *st4;
        cout << *st5;
    }
};

ostream & operator<<(ostream & o, Grupa & g)
{
    cout << "    GROUP-->";
    o << g.nameGroup; cout << endl;
    cout << "SPECIALTY-->";
    o << g.specialty; cout << endl;
    g.printObject();

    return o;
}

int main()
{
    cout << endl << "ENTER INFORMATION ABOUT THE STUDENTS " << endl << endl;
    cout << "-----" << endl;
    Student student1;

    Student student2;

    Student student3;

    Student student4;

    Student student5;

    string nameGroup;
    cout << "Enter a name for the group "; cin >> nameGroup;

    string nameSpecialty;
    cout << "Enter a name for the specialty"; cin >> nameSpecialty;
    cout << endl << "-----" << endl;
    Grupa g(nameGroup, nameSpecialty, student1, student2, student3, student4,
student5);
    cout << g;

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

Microsoft Visual Studio Debug Console

ENTER INFORMATION ABOUT THE STUDENTS

Input lastName Balko

Input firstName Ivan

Input lastName Petrishin

Input firstName Oleg

Input lastName Kulik

Input firstName Petro

Input lastName Golinka

Input firstName Sergii

Input lastName Babkin

Input firstName Vlad

Enter a name for the group KH-321

Enter a name for the specialty programmer

GROUP--> KH-321
SPECIALTY--> programmer

Balko Ivan

Petrishin Oleg

Kulik Petro

Golinka Sergii

Babkin Vlad

C:\Users\ASUS\source\repos\TEST\Debug\TEST.exe (proce

Завдання 3.

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.

КОД ПРОГРАМИ

```
#include <iostream>
#include <string>
using namespace std;
class Student;

class Student
{
private:
    string lastName;
    string firstName;
    string surName;
    int numberZalikBook;
    bool statesman;
    bool payer;
public:
    friend ostream& operator <<(ostream&, Student&);

    Student()
    {
        cout << "Input lastName "; cin >> lastName; cout << endl;
        cout << "Input firstName "; cin >> firstName; cout << endl;
    }

    Student(string lastName, string firstName, string surName, int numberZalikBook,
bool statesman, bool payer)
    {
        this->lastName = lastName;
        this->firstName = firstName;
        this->surName = surName;
        this->numberZalikBook = numberZalikBook;
        this->statesman = statesman;
        this->payer = payer;
    }
}
```



```

};

ostream & operator<<(ostream & s, Student & object)
{
    cout << endl;
    s << object.lastName; cout << " ";
    s << object.firstName; cout << endl;;

    return s;
}

class Grupa
{
    string nameGroup;
    string specialty;
public:
    Student *st1;
    Student *st2;

    Grupa(string nameGroup, string specialty, Student & student1, Student & student2)
    {
        this->nameGroup = nameGroup;
        this->specialty = specialty;
        this->st1 = &student1;
        this->st2 = &student2;
    }

    friend ostream & operator << (ostream & o, Grupa & g);

    void printObject()
    {
        cout << *st1;
        cout << *st2;
    }
};

ostream & operator<<(ostream & o, Grupa & g)
{
    cout << "    GROUP-->";
    o << g.nameGroup; cout << endl;
    cout << "SPECIALTY-->";
    o << g.specialty; cout << endl;
    g.printObject();

    return o;
}

class Facultet
{
private:
    string nameFacultet;
public:
    Grupa *gPoint;
    Grupa *gPoint2;
    Grupa *gPoint3;

```

```

void createGroups()
{
    cout << endl << "ENTER INFORMATION ABOUT STUDENTS " << endl << endl;
    cout << "-----" << endl;
    Student student1;

    Student student2;

    string nameGroup;
    cout << "Enter a name for the group "; cin >> nameGroup;

    string nameSpecialty;
    cout << "Enter a name for the specialty"; cin >> nameSpecialty;
    cout << endl << "-----" << endl;
    Grupa g(nameGroup, nameSpecialty, student1, student2);
    gPoint = &g;

    cout << endl << "ENTER INFORMATION ABOUT STUDENTS " << endl << endl;
    cout << "-----" << endl;
    Student student3;

    Student student4;

    cout << "Enter a name for the group "; cin >> nameGroup;

    cout << "Enter a name for the specialty"; cin >> nameSpecialty;
    cout << endl << "-----" << endl;
    Grupa g2(nameGroup, nameSpecialty, student3, student4);
    gPoint2 = &g2;

    cout << endl << "ENTER INFORMATION ABOUT STUDENTS " << endl << endl;
    cout << "-----" << endl;
    Student student5;

    Student student6;

    cout << "Enter a name for the group "; cin >> nameGroup;

    cout << "Enter a name for the specialty"; cin >> nameSpecialty;
    cout << endl << "-----" << endl;
    Grupa g3(nameGroup, nameSpecialty, student5, student6);
    gPoint3 = &g3;

    cout << endl << "-----" << endl
    << "                LIST OF STUDENTS OF THE "<< nameFacultet<<" FACULTY" <<
    endl << "-----" << endl;
    cout << g;
    cout << endl<<"-----" <<endl;
    cout << g2;
    cout << endl << "-----" << endl;
    cout << g3;

}
friend ostream & operator <<(ostream & o, Facultet & f);

Facultet(string nameFacultet)
{

```

```

        this->nameFacultet = nameFacultet;
        creteGroups();
    }

};

ostream & operator<<(ostream & o, Facultet & f)
{
    cout<<f.gPoint;
    cout << f.gPoint2;
    cout << f.gPoint3;
    return o;
}

int main()
{
    cout << "Enter the name of the faculty " << endl;
    string nameFaculty; cin >> nameFaculty; cout << endl;
    Facultet myFaculty(nameFaculty);
    cout << myFaculty;

    return 0;
}

```

РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ

Microsoft Visual Studio Debug Console

Enter the name of the faculty
Computer_Science

ENTER INFORMATION ABOUT STUDENTS

Input lastName Balko

Input firstName Ivan

Input lastName Bogatyuk

Input firstName Petro

Enter a name for the group KH-321
Enter a name for the specialty programmer

ENTER INFORMATION ABOUT STUDENTS

Input lastName Golinka

Input firstName Sergey

Input lastName Petrishin

Input firstName Nazar

Enter a name for the group KH-121
Enter a name for the specialty programmer

ENTER INFORMATION ABOUT STUDENTS

Input lastName Kulick

Input firstName Igor

Input lastName Stodola

Input firstName Nazar

Enter a name for the group KH-221
Enter a name for the specialty programmer

```
-----  
LIST OF STUDENTS OF THE Computer_Science FACULTY  
-----  
GROUP--> KH-321  
SPECIALTY--> programmer  
  
Balko Ivan  
Bogatyuk Petro  
  
-----  
GROUP--> KH-121  
SPECIALTY--> programmer  
  
Golinka Sergey  
Petrishin Nazar  
  
-----  
GROUP--> KH-221  
SPECIALTY--> programmer  
  
Kulick Igor  
Stodola Nazar
```

Висновок: ознайомитись зі способами та механізмами композиції та агрегації в ООП.