

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

BÀI TẬP TUẦN 5

2022 – 2023

Bài tập thực hành Kế thừa (tiếp theo)

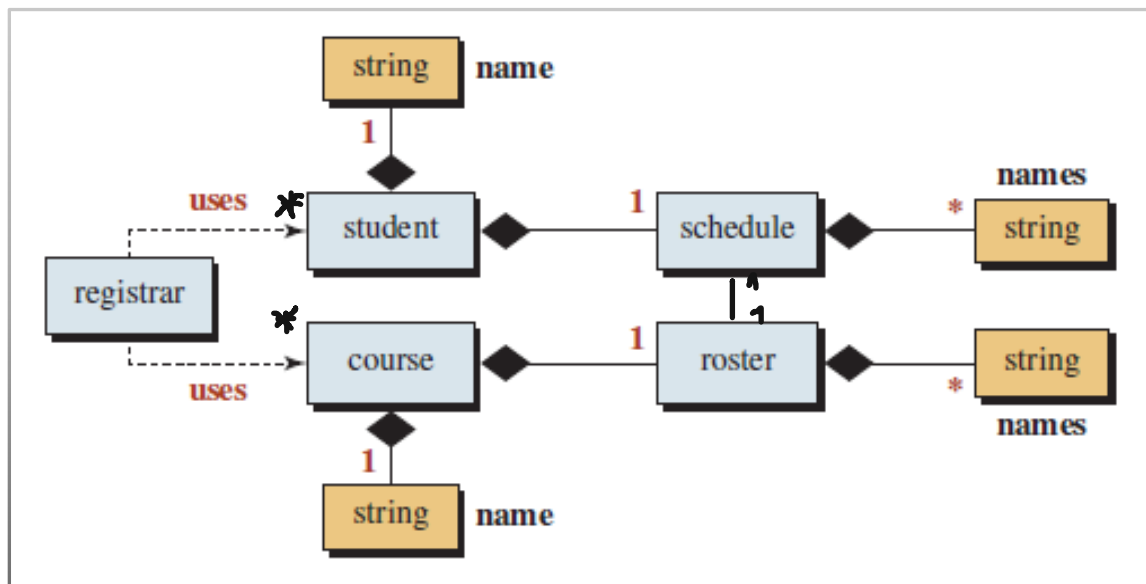
Bài 1 Hiện thực lớp vector sử dụng mảng động một chiều, lớp template vector được định nghĩa như sau:

```
/* *****  
 * Implement vector class template using dynamic array  
 * Define vector class template - vector.h  
 * ***** */  
#ifndef VECTOR_H  
#define VECTOR_H  
#include <iostream>  
#include <string>  
using namespace std;  
  
/* vector class template */  
template<typename T>  
class vector {  
private:  
    T* ptr;  
    int capacity;  
    int sz;  
public:  
    // Constructors  
    vector();  
    vector(int cap);  
    ~vector();  
  
    // utilities methods  
    /* operator[] */  
    T& operator[](int n);  
  
    /* Expand extra memory */  
    void reserve(int newalloc);  
  
    // input, output  
    void push(const T& element);  
    void print();  
  
};  
#endif  
-----
```

Yêu cầu

Hiện thực lớp template vector, và viết chương trình chính để kiểm thử lớp vector này.

Bài 2 Viết chương trình quản lý hệ thống đăng ký của một phòng ban của trường Cao đẳng/Đại học. Biết rằng sơ đồ quan hệ giữa các lớp trong hệ thống đăng ký như Hình 2.1.



Hình 2.1 Sơ đồ Quan hệ giữa các lớp trong hệ thống đăng ký

Trong Hình 2.1 có 5 lớp. Mỗi đối tượng Sinh Viên (student) tạo 1 đối tượng chuỗi (tên sinh viên) và tạo một đối tượng Lịch biểu (schedule), tạo một mảng các tên Khóa học (string). Mỗi đối tượng Khóa học (course) tạo 1 đối tượng chuỗi (tên khóa học) và tạo 1 đối tượng bảng phân công (roster), tạo ra 1 mảng tên Sinh viên (string). Lớp Đăng ký (registrar) chỉ sử dụng các đối tượng Sinh Viên và Khóa học.

Giả sử một Sinh viên có thể tham gia tối đa 5 khóa học, và mỗi khóa học có thể có 40 sinh viên.

Biết rằng :

- + Thông tin Sinh viên gồm: Mã sinh viên (C-string), tên sinh viên (C-string), ngày sinh, địa chỉ (C-string).
- + Thông tin Khóa học gồm: Mã khóa học (C-string), tên khóa học (C-string), lịch học (ngày mở).

Yêu cầu :

1. Đọc danh sách khóa học
2. Sinh viên đăng ký khóa học
3. Ghi danh sách sinh viên lên file
4. Đọc danh sách sinh viên từ file, và xuất ra màn hình
5. In danh sách sinh viên của một khóa học
6. In thời khóa biểu của sinh viên (danh sách khóa học sinh viên đã đăng ký)

Hướng dẫn

Hướng dẫn

Bài 2: Hệ thống Đăng ký Khóa học sử dụng quan hệ ‘Association’ và ‘Dependency’ trong thiết kế sơ đồ quan hệ.

Relationship among classes

• INHERITANCE

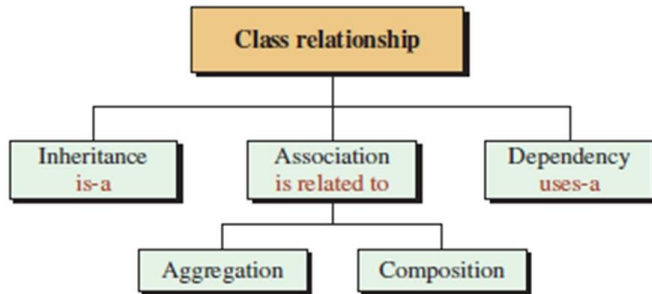
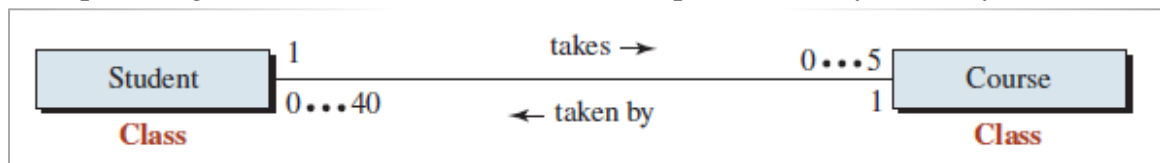


Figure 3.1 Relationship among classes

- Inheritance
- Association
- Not all relationships between classes can be defined as inheritance.

21

Mỗi quan hệ giữa ‘Student’ và ‘Course’ là mối quan hệ ‘many-to-many’.



Hình 2.3 Mối quan hệ giữa Sinh_viên và Khóa_học

Phân tích Quan hệ Student – Course : ‘many-to-many’;

Relationship among classes

• INHERITANCE

- Design a simple registration system

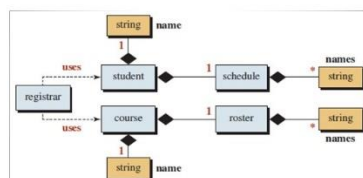


Figure 4.1 Relationship among classes in registration project

• UML – Unified Modeling Language

- Student Schedule : Array -of-courseNames

Student schedule							
	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
Morning	ML	OOP	OOP	x	PTech	DL	x
Afternoon	DL	x	x	OOP	OOP	ML	x

- Array-of-courseNames = {"Mon":{"Machine Learning", "Deep Learning"}, "Tues":{"OOP", "x"}, "Wed":{"OOP", "x"},...}

- Course Roster : Array-of-studentNames

- Array-of-studentNames = {"1": "Le Van Chi", "2": "Nguyen Van Ba", "3": "Dang Van Bi", "4": "Vo Van Ngan",...}

29

Hướng dẫn – Chia file : *Mỗi lớp – class là một file tách biệt*

Separating Class Definition Implementation

- Separating Class Definition Implementation

```
#ifndef ClassName_H
#define ClassName_H

A class header for the class named ClassName

#endif
```

- Example

- Định nghĩa lớp Circle - file **Circle.h**

```
#ifndef CIRCLE_H
#define CIRCLE_H
#include <iostream>
using namespace std;
class Circle { /* code here */ };
#endif
```

- Hiện thực lớp Circle – File **Circle.cpp**

```
#include "Circle.h"

/* code here */
```

- Cài đặt main() function – File **app.cpp**
(Testing program)

```
#include "Circle.h"

int main() { /* code here */ }
```

31