



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

HDTH - BÀI TẬP TUẦN 1 & 2


HKII 2022-2023

Phần I – Kiến thức nền

1. Viết và sử dụng hàm (function) trong chương trình, mỗi hàm thực hiện một chức năng của chương trình.
2. Các kiểu cấu trúc dữ liệu:
 - Struct
 - Arrays
 - Linked list
 - Stacks
 - Queue
 - Tree
 - Hash table
 - Graph

Ví dụ: Định nghĩa cấu trúc dữ liệu 'Student' sử dụng cấu trúc 'struct'

```
/* Định nghĩa cấu trúc Student */  
struct Student {  
    char stdName[30];  
    char stdId[9];  
    char ngay[11];        // "01/01/2000"  
    float Marks[3];       // baitap, giuaky, cuoiky  
    float stdGpa;  
};
```

-  Sử dụng cấu trúc mảng động để cài đặt danh sách sinh viên
- ```
Student** list-of-student;
```

3. Quản lý tập tin – Đọc/Ghi tập tin

-----

## Phần II – Lập trình hướng đối tượng

### Bài tập thực hành class and objects

**BÀI 1** Hiện thực lớp Stack sử dụng cấu trúc mảng động.

Biết rằng lớp Stack định nghĩa như sau:

```
#ifndef STACK_H
#define STACK_H
#include <iostream>
#include <cassert>
using namespace std;

class Stack {
private:
 int* ptr;
 int capacity;
 int size;
public:
 // Constructor
 Stack(int capacity);
 // destructor
 ~Stack();
 // Thêm phần tử vào stack, thêm vào cuối
 void push(const int& element);
 // Lấy phần tử khỏi stack, Lấy ra ở cuối theo LIFO
 int pop();
};
#endif
```

-----

## Bài 2 Cài đặt các lớp sau

1. Cài đặt lớp danh sách liên kết
2. Cài đặt lớp ngăn xếp sử dụng danh sách liên kết (đã cài đặt ở Câu 1)

Biết rằng

- Lớp danh sách liên kết định nghĩa như sau

```
#ifndef LIST_H
#define LIST_H
#include <iostream>
#include <cassert>
#include <string>
using namespace std;

// Định nghĩa Node
struct Node {
 string data;
 Node* next;
};
// Định nghĩa lớp list
class List {
private:
 Node* Head;
 int count;
 Node* makeNode(const string& val);
public:
 List();
 ~List();
 void insert(int pos, const string& val);
 void erase(int pos);
 string get(int pos) const;
 void print() const;
 int size() const;
};
#endif
```

- Lớp stack định nghĩa sử dụng lớp List như sau:

```
#ifndef STACK_H
#define STACK_H
#include "list.h"
// Định nghĩa lớp Stack sử dụng đối tượng list
class Stack {
private:
```

```
List list;
public:
 void push(const string& val);
 void pop();
 string top() const;
 int size() const;
};
#endif

```

**Bài 3** - Viết chương trình quản lý sinh viên, với thông tin sinh viên gồm: Họ và tên (C-string), mã số (C-string), ngày sinh (C-string), Marks là mảng gồm 3 cột điểm: Bài tập, thi giữa kỳ, và thi cuối kỳ. Biết rằng công thức tính điểm trung bình như sau:

$$\text{stdAvg} = \text{baitap} * 0.25 + \text{giuaky} * 0.25 + \text{cuoiky} * 0.5$$

### Yêu cầu

1. Cài đặt lớp sinh viên, viết các hàm dựng (Constructor), hàm hủy (Destructor), hàm sao chép, hàm gán bằng.
2. Xây dựng lớp quản lý danh sách sinh viên (*QLSinhVien*) sử dụng mảng động/danh sách liên kết.
3. Viết hàm ghi danh sách sinh viên lên tập tin.
4. Viết hàm đọc danh sách sinh viên từ tập tin.
5. Viết hàm tìm các sinh viên có điểm trung bình nhỏ hơn điểm trung bình của lớp, ghi danh sách các sinh viên này vào tập tin.
6. Viết hàm thêm một sinh viên mới vào danh sách và cập nhật lại vào tập tin danh sách sinh viên.
7. Viết hàm xếp loại sinh viên.
8. Viết hàm xuất danh sách sinh viên có thông tin xếp loại.
9. Viết hàm tìm tất cả các sinh viên có cùng ngày sinh nhật trong ngày hiện tại.

**Note:** Bài 3, bài tập lớn, Sinh viên có thể làm theo nhóm, mỗi nhóm 2 sv. Chương trình hoàn thành đầy đủ theo yêu cầu ( $\geq 75\%$ ), có hệ thống menu để chạy các chức năng của chương trình. Sinh viên báo cáo Bài 3 trên lớp vào các tuần 2, và tuần 3.

-----