

# Đề thi thực hành cuối kỳ

Môn: PP. LTHDT

Lớp: 19CTT4

## Phần I – Online

**Bài 1** - Chương trình quản lý danh sách hoá đơn tiền điện của khách hàng sử dụng phương pháp lập trình hướng đối tượng (*thừa kế và đa hình*). Thông tin bao gồm các loại khách hàng:

- Khách hàng Việt Nam: mã khách hàng (*kiểu chuỗi có 10 kí tự, ví dụ: "1234567890"*), họ tên (*kiểu chuỗi*), ngày ra hoá đơn (*kiểu chuỗi, ví dụ: "10/03/2021"*), đối tượng khách hàng (*sinh hoạt, kinh doanh, sản xuất*): số lượng (số KW tiêu thụ), đơn giá, định mức.

Biết rằng:

- Đối tượng *sinh hoạt* có định mức 200 kw
- Đối tượng *kinh doanh* có định mức 300 kw
- Đối tượng *sản xuất* có định mức 500 kw

Thành tiền được tính như sau:

- Nếu số lượng  $\leq$  định mức thì: thành tiền = số lượng \* đơn giá
- Ngược lại thì: thành tiền = đơn giá \* định mức + số lượng KW vượt định mức \* đơn giá \* 2.5.

(Số lượng KW vượt định mức = số lượng – định mức);

- Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá.

Thành tiền = số lượng \* đơn giá.

Áp dụng kế thừa và đa hình, khai báo lớp khách hàng Việt Nam (*class KHVietNam*) kế thừa lớp Khách hàng (*class KhachHang*), và lớp KHNuocNgoai thừa kế lớp KhachHang. Cài đặt các hàm sau:

1. Cài đặt các hàm constructor/destructor cần thiết cho mỗi lớp.
2. Nhập thông tin khách hàng.
3. Xuất thông tin khách hàng.
4. Tính thành tiền.

Xây dựng lớp quản lý danh sách các khách hàng (*class QLKhachHang*) sử dụng mảng động, cài đặt các hàm sau:

5. Nhập danh sách các hoá đơn khách hàng.
6. Tính tổng số lượng cho từng loại khách hàng.
7. Tính trung bình thành tiền của khách hàng Việt Nam.
8. Xuất ra các hoá đơn trong tháng 03 năm 2021 (của cả 2 loại khách hàng).
9. Giả sử danh sách khách hàng đã được sắp xếp theo mã khách hàng hãy thêm một khách hàng mới vào danh sách đã cho, mà vẫn duy trì thứ tự sau khi thêm.

## Phần II – offline

**Bài 2** - 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 (Constructors), 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. Giả sử danh sách sinh viên được sắp xếp tăng dần theo mã số sinh viên, 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.

**Bài 3** – Cho chương trình sau:

```
/* *****  
 * Class Object, Inheritance, Polymorphism  
 * ***** */  
#include<iostream>  
using namespace std;  
  
class Base {  
public:  
    Base() {};  
    virtual ~Base() {};  
    virtual void mfunc() { };  
};  
  
class Drived_1 : public Base {  
public:  
    Drived_1() {};  
    ~Drived_1(){};  
    // Override member function mfunc()  
    virtual void mfunc() {};  
};  
  
class Drived_2 : public Base {  
public:  
    Drived_2() {};  
    ~Drived_2(){};  
    // Override member function mfunc()  
    virtual void mfunc() {};
```

```

};

int main() {

    Base b;          //(1)
    Base* pb = new Base();    //(2)
    Drived_1 D1;      //(3)
    Drived_2 D2;      //(4)
    Drived_1* pD1 = new Drived_1();    //(5)
    Base* pbD1 = new Drived_1();    //(6)

    Drived_2 pD2 = new Base(); //(7)

    Base** roster = new Base*[3];    //(8)
    for (int i = 0; i < 3; i++) {    //(9)
        roster[i] = NULL;    //(10)
    }
    roster[0] = new Drived_1();    //(11)
    roster[1] = new Drived_2();    //(12)
    roster[2] = new Drived_1();    //(13)

    for (int i = 0; i < 3; i++) {    //(14)
        Drived_1* ptr = dynamic_cast<Drived_1*>(roster[i]);    //(15)
        ptr->mfunc();    //(16)
    }
}

```

### Yêu cầu:

1. Hãy giải thích ngắn gọn các phát biểu từ (1) đến (6) làm gì?
2. Phát biểu (7) đúng/Sai, Tại sao?
3. Giải thích sự khác nhau giữa hai phát biểu (2) và (6)
4. Hãy giải thích các phát biểu từ (8) đến (15) làm gì?
5. Phát biểu (16): Thì đối tượng ptr sẽ gọi hàm mfunc của lớp nào trong các lớp Base, Drived\_1, Drived\_2, giải thích tại sao?

*Note: Sử dụng kiến thức về Lớp và đối tượng, Thừa kế và Đa hình để trả lời các yêu cầu trên.*

-----\*\*\*-----