



Đề thi môn Kỹ Thuật Lập Trình, niên khóa 2010-2011

Thời gian : 100 phút

Tài liệu mở

Phần câu hỏi (2 điểm)

Sinh viên chỉ cần trả lời đúng 4 câu trong phần này

Câu 1.

Đâu là lý do tốt nhất cho việc sử dụng ngôn ngữ lập trình hướng đối tượng :

- a) Có thể định nghĩa các kiểu dữ liệu riêng
- c) Một chương trình hướng đối tượng có thể được dạy để tự sửa lỗi
- c) Dễ dàng định hình một chương trình hướng đối tượng
- d) Có thể định nghĩa các kiểu dữ liệu riêng và dễ dàng định hình một chương trình hướng đối tượng

Câu 2.

Ba tính năng chính của ngôn ngữ lập trình hướng đối tượng ?

- a) Đóng gói dữ liệu (*data encapsulation*), thừa kế (*inheritance*) và xử lý ngoại lệ (*exception handling*)
- b) Thừa kế (*inheritance*), đa hình (*polymorphism*) và xử lý ngoại lệ (*exception handling*)
- c) Đóng gói dữ liệu (*data encapsulation*), thừa kế (*inheritance*) và đa hình (*polymorphism*)
- d) Quá tải (*overloading*), thừa kế (*inheritance*) và đa hình (*polymorphism*)

Câu 3.

Hàm tạo (*constructor*) là

- a) một hàm được gọi khi một thực thể của lớp được khởi tạo.
- b) một hàm được gọi khi một thực thể của lớp bị xóa.
- c) một hàm đặc biệt để thay đổi giá trị của bộ nhớ được cấp phát động.
- d) một hàm được gọi để thay đổi giá trị của một biến.

Câu 4.

Cho đoạn mã sau đây:

```
int recur(int n, int a, int d){  
    if(n==1)  
        return a;  
    else  
        return d + recur(n-1, d, a-1);  
    return 0 ;  
}
```

Giá trị trả về là bao nhiêu khi gọi **recur(3,5,7)**.



- a) 11
- b) 13
- c) 15
- d) 17
- e) 19

Câu 5.

Một lớp dẫn xuất (*derived class*)

- a) thừa kế dữ liệu và phương thức thành viên (*data members and member functions*) từ lớp cơ sở (*base class*)
- b) thừa kế các hàm tạo (*constructors*) và các hàm hủy (*destructors*)
- c) đối tượng có thể truy cập vào những thành viên được bảo vệ (*protected members*) cùng với dấu chấm (*dot operator*).
- d) thừa kế dữ liệu và phương thức thành viên (*data members and member functions*) từ lớp cơ sở (*base class*) cũng như là thừa kế các hàm tạo (*constructors*) và các hàm hủy (*destructors*)

Câu 6.

Giả sử rằng lớp **X** không có quá tải toán tử cho phép gán (*overloaded assignment operator*).

Điều gì xảy ra khi một phép gán **a=b** được thực hiện, với **a** và **b** là hai đối tượng thuộc lớp **X**?

- a) Toán tử gán tự động (*automatic assignment operator*) sẽ được sử dụng
- b) Hàm tạo bản sao (*copy constructor*) sẽ được dùng
- c) Xuất hiện lỗi biên dịch (*compiler error*)
- d) Xuất hiện lỗi thực thi (*run-time error*)

Phần lập trình (8 điểm)

Để hỗ trợ hiện thực một hệ thống quản lý khách hàng (*CRM – customer relation management*), chúng ta cần xây dựng một class **Customer** và một class **CRM** với mục đích như sau:

- **Customer**: lưu trữ thông tin, những thuộc tính và phương thức cần thiết để thao tác với các thực thể khách hàng như sau:
 - o Khách hàng có thể là cá nhân hoặc đại diện cho một cơ quan, doanh nghiệp nào đó.
 - o Trường hợp khách hàng mang tính chất cá nhân, cần lưu những thông tin cá nhân sau: mã khách hàng, họ và tên, và các thông tin bổ sung như tên viết tắt (biệt danh), địa chỉ, ngày sinh (bao gồm ngày tháng và năm), mã dịch vụ (kiểu số



- nguyên, ở đây chúng ta không cần quan tâm đến tên cụ thể của dịch vụ), số lần sử dụng dịch vụ, chi phí và thời gian (ngày, giờ) mỗi lần sử dụng dịch vụ
- Trường hợp khách hàng đại diện cho doanh nghiệp, ngoài thông tin cá nhân cần lưu thêm thông tin về doanh nghiệp như: mã số thuế, tên doanh nghiệp, địa chỉ, điện thoại, fax, email, địa chỉ website doanh nghiệp
 - Loại thẻ của khách hàng (thẻ vàng, thẻ bạc, thẻ thông thường)
- **CRM**: quản lý danh sách tất cả khách hàng (nghĩa là chứa một mảng các đối tượng thuộc lớp **Customer**)

Sau đây, sinh viên cần khai báo các thuộc tính và phương thức của các lớp **CRM** và **Customer** bằng ngôn ngữ C++ thỏa mãn các yêu cầu sau:

- a) (1 điểm) Hãy khai báo các thuộc tính cần thiết như đặc tả bên trên của hai lớp **CRM** và **Customer**.

Giả sử, chúng ta đã có sẵn đối tượng **DateTime** như sau :

```
class DateTime{int hour, minute, second, day, month, year;}
```

và khai báo kiểu liệt kê Card cho sẵn như sau :

```
enum Card {GodenCard, SilverCard, NormalCard }
```

- b) (0,5 điểm) Xây dựng những hàm tạo của **Customer** để có thể tạo và nạp thông tin chi tiết nhanh chóng: loại khách hàng (cá nhân hay doanh nghiệp, tên, ngày sinh).
- c) (1 điểm) Trong class **CRM**, viết một phương thức hỗ trợ thêm một khách hàng mới, hãy dùng vòng lặp **do..while** để kiểm tra xem đã có chứa thông tin của khách hàng này chưa; nếu có sẵn thông tin thì không cần thêm vào, trường hợp ngược lại thì thêm vào cuối danh sách.
- d) (1 điểm) Xây dựng một phương thức trong class **CRM**, cho phép tìm kiếm một đối tượng theo tên của khách hàng và trả về đối tượng **Customer** lưu trữ thông tin tương ứng (trả về **NULL** nếu không tìm thấy)
- Hãy sử dụng vòng lặp **while(){ ...}** để hiện thực
 - Hãy sử dụng kỹ thuật gọi đệ quy để hiện thực
- e) (0,5 điểm) Xây dựng những phương thức trong **Customer** cho phép truy xuất và sửa chữa thông tin của khách hàng: thay đổi (nâng cấp) thẻ của một khách hàng (thẻ vàng, thẻ bạc



hay thông thường) và làm thế nào để xác định một khách hàng là thuộc dạng cá nhân hay là doanh nghiệp.

- f) (0,5 điểm) Viết một phương thức trong **Customer** trả về cùng lúc số lượng dịch vụ sử dụng và tổng số tiền dịch vụ thu được từ khách hàng.
- g) (0,5 điểm) Xây dựng những phương thức trong **CRM**, cho phép chỉnh sửa thông tin khách hàng có mã là 'ID' như là: họ tên, tên viết tắt, địa chỉ, ngày sinh.
- h) (1 điểm) Viết phương thức **void useService(int serviceCode, int servicePrice)** cho phép nhập vào thông tin của dịch vụ sau mỗi lần sử dụng cho cả hai lớp đối với một khách hàng. Lưu ý rằng cần nhập thông tin về thời gian sử dụng dịch vụ, ở đây chúng ta đơn giản hóa bằng cách lưu trữ thời điểm lưu trữ thông tin. Giả sử trong thư viện, ta có sẵn hàm **void time(DateTime &x)** sẽ gán thời điểm gọi hàm vào biến **x**.
- i) (1 điểm) Hãy dùng từ khóa **static** để khai báo thêm các thuộc tính cần thiết trong lớp **Customer** và sau đó viết các phương thức hỗ trợ việc tính thống kê trong lớp **CRM** như :
 - o số lượng khách riêng lẻ (cá nhân),
 - o số lượng khách là doanh nhân,
 - o tổng số lượng khách hàng,
 - o tỷ lệ khách doanh nhân trên tổng số,
- j) (0,5 điểm) Hãy sử dụng kỹ thuật gọi đệ quy để xây dựng phương thức trong **CRM** để hiển thị thông tin khách hàng theo thứ tự trong bảng chữ cái.
 - o Giả sử cho sẵn hàm so sánh giữa hai chuỗi **compare(char* s1, char* s2)**: trả về giá trị **"true"** nếu như **s1** đứng trước **s2** theo thứ tự trong bảng chữ cái, và trả về **"false"** trong trường hợp ngược lại.
- k) (0,5 điểm) Viết hàm **main**, tạo một lớp **CRM** và nhập vào khoảng 5 khách hàng cá nhân và 5 doanh nhân; sau đó, xuất ra các thông tin thống kê cho sẵn.

-HẾT-