

Tên học phần: **Phương pháp Lập trình Hướng đối tượng**

Thời gian làm bài: **110 phút**

Câu 1 (05 điểm)

```
01 #include <iostream>
02 #include <sstream>
03 #include <string>
04 #include <typeinfo>
05 using namespace std;
06 const double PI = 3.1415928;
07 class Shape
08 {
09 public:
10     Shape(){ cerr<<"construct Shape"<<endl; }
11     virtual string Description() {
12         stringstream ss;
13         ss<<"Shape(area=" << this->Area() << ")";
14         return ss.str();
15     }
16     virtual float Area() = 0;
17     virtual void Scale(float scaleFactor) = 0;
18     const type_info& InterfaceType() {
19         return typeid(Shape);
20     }
21     const type_info& ImplementationType() {
22         return typeid(*this);
23     }
24     virtual ~Shape(){
25         cerr<<"destruct Shape"<<endl;
26     }
27 };
28
29 class Circle : public Shape
30 {
31 private:
32     float radius;
33 public:
34     Circle(float r) :Shape(), radius(r) {
35         cerr << "construct Circle" << endl;
36     }
37     string Description() {
38         stringstream ss;
39         ss << "Circle(r=" << this->radius << ")";
40         return ss.str();
41     }
42     float Area() { return PI*radius*radius; }
43     void Scale(float scaleFactor) {
44         this->radius *= scaleFactor;
45     }
46     const type_info& InterfaceType() {
47         return typeid(Circle);
48     }
49     ~Circle(){ cerr<<"destruct Circle"<<endl; }
50 };
51
52 class Ellipse : public Shape
53 {
54 private:
55     float majorR, minorR;
56 public:
57     Ellipse(float a, float b)
58         :Shape(), majorR(a), minorR(b)
59     {
60         cerr << "construct Ellipse" << endl;
61     }
62     string Description() {
63         stringstream ss;
64         ss << "Ellipse(a=" << this->majorR
65            << ", b=" << this->minorR << ")";
66         return ss.str();
67     }
68     float Area(){ return PI*majorR*minorR; }
69     void Scale(float scaleFactor) {
70         this->majorR *= scaleFactor;
71         this->minorR *= scaleFactor;
72     }
73     void Scale(float sa, float sb) {
74         this->majorR *= sa;
75         this->minorR *= sb;
76     }
77     const type_info& InterfaceType() {
78         return typeid(Ellipse);
79     }
80     ~Ellipse(){
81         cerr<<"destruct Ellipse"<<endl;
82     }
83 };
```

```

83 int main()
84 {
85     Ellipse ellipse(2.0/3, 6);
86     Shape *shape = &ellipse;
87     shape->Scale(1.0/2);
88     //shape->Scale(3, 1.0/3);
89
90     cout << "ellipse: " << endl;
91     cout << ellipse.Description() << endl;
92     cout << ellipse.Area() << endl;
93     cout << ellipse.InterfaceType().name() << endl;
94     cout << ellipse.ImplementationType().name() << endl;
95
96     cout << "shape = &ellipse: " << endl;
97     cout << shape->Description() << endl;
98     cout << shape->Area() << endl;
99     cout << shape->InterfaceType().name() << endl;
100    cout << shape->ImplementationType().name() << endl;
101
102    //Shape ashape;
103    //cout << ashape.Description() << endl;
104
105    return 0; //no error
106}

```

Chú thích về chương trình C++ bên trên:

- cerr và cout là 2 luồng xuất chuẩn đều **xuất ra màn hình console**.
- typeid là toán tử tính kiểu của một biểu thức, trả về đối tượng type_info.
VD: `class X; typeid(X).name()=="X";`
`X *a; typeid(a).name()=="X*";`

Hãy trả lời những câu hỏi liên quan tới chương trình bên trên:

- Hãy viết kết quả xuất ra màn hình sau khi chương trình chạy xong.
- Trong chương trình có nhiều hàm có cùng tên, chúng có quan hệ gì với nhau (hay không có quan hệ gì hết)? Hãy giải thích quan hệ giữa 2 hàm trong những cặp trùng tên sau: (Nêu tên quan hệ nếu có, nêu ý nghĩa và ứng dụng của quan hệ đó.)
 - Shape::Description() và Circle::Description()
 - Ellipse::Scale(float) và Ellipse::Scale(float, float)
 - Shape::InterfaceType() và Circle::InterfaceType()
 - Circle::InterfaceType() và Ellipse::InterfaceType()
- Nếu bỏ dấu comment (//) của các dòng 88, 102 và 103 ra thì chương trình gặp lỗi gì?
- [*] Phải thay thế vào chỗ của các dòng bị comment (88, 102 và 103) những dòng code như thế nào để thực hiện được ý đồ của các dòng code đó:
 - Ở dòng 88 muốn gọi phương thức Ellipse::Scale(float, float) thông qua con trỏ shape (chứ không phải thông qua biến đối tượng ellipse).
 - Ở dòng 102 và 103 muốn gọi được phương thức Shape::Description().

* Lưu ý: Những câu đánh dấu sao [*] là không bắt buộc (tính điểm cộng).

Câu 2 (05 điểm)

Bệnh viện Nhân dân 115 là một trong những bệnh viện lớn tại TP.HCM. Hiện bệnh viện đang có nhu cầu xây dựng hệ thống quản lý hoạt động khám chữa bệnh của các bệnh nhân. Mỗi bệnh nhân được cấp 1 mã số (MSBN) để tiện việc quản lý danh sách các bệnh nhân. Bệnh viện cho phép khám điều trị nội trú và ngoại trú. Nếu cùng 1 người có cả hoạt động khám ngoại trú lẫn điều trị nội trú thì sẽ được cấp 2 mã số khác nhau và được quản lý riêng như 2 người khác nhau. Viện phí của từng bệnh nhân được tính như sau:

Với bệnh nhân ngoại trú: Bệnh viện thu một mức phí nhất định cho mỗi lần khám tùy theo mỗi loại bệnh lý.

Với bệnh nhân nội trú: Tiền khám chữa bệnh = số ngày nằm viện * (phí khám bệnh mỗi ngày + đơn giá phòng).

Có 3 loại phòng cho bệnh nhân điều trị nội trú chọn lựa với đơn giá:

- Phòng loại C, đơn giá 600.000 đ/ngày.
- Phòng loại B, đơn giá 900.000 đ/ngày.
- Phòng loại A, đơn giá 1.400.000 đ/ngày.

Mỗi hoạt động khám chữa bệnh điều được ghi nhận lại dưới dạng nhật ký (log hoạt động) theo tháng. Tùy theo từng loại hoạt động, mỗi dòng log sẽ có các thông tin tương ứng như sau:

- BN ngoại trú khám bệnh: Ngày, MSBN, "KB", phí khám chữa bệnh.
- BN nội trú nhập viện: Ngày, MSBN, "NV", phí khám chữa bệnh mỗi ngày, loại phòng.
- BN nội trú xuất viện: Ngày, MSBN, "XV".
- Tổng kết viện phí (cuối cùng trong log): Ngày, -1, "TKVP".

Trong đó:

- Ngày là một số nguyên từ 1 tới 31 thể hiện ngày trong tháng.
- MSBN là một số tự nhiên, được đánh số liên tiếp từ 1.
- "KB", "NV", "XV", "TKVP" là các chuỗi ký tự thể hiện loại hoạt động.

Ví dụ một log hoạt động 8 ngày đầu tiên của tháng 12:

(Giả sử chưa có bệnh nhân nào trong danh sách.)

Nội dung tập tin log hoạt động	Giải thích ý nghĩa & Viện phí tới ngày hôm đó (VPTL)
1 001 KB 20000	new BN[MSBN=001] khám bệnh => VPTL = 20kđ
1 002 NV 100000 B	new BN[MSBN=002] nhập viện vào phòng loại B, mỗi ngày khám chữa bệnh hết 100kđ
6 003 KB 50000	new BN[MSBN=003] khám bệnh => VPTL = 50kđ
6 001 KB 30000	BN[MSBN=001] khám bệnh => VPTL = 20kđ + 30kđ = 50kđ
6 004 NV 300000 C	new BN[MSBN=004] nhập viện vào phòng loại C, mỗi ngày khám chữa bệnh hết 300kđ
7 004 XV	BN[MSBN=004] xuất viện => VPTL = $I \cdot (300k + 600kđ) = 900kđ$
8 -1 TKVP	<p>Tổng kết viện phí:</p> <p>+ BN[MSBN=001] => VPTL = 50kđ</p> <p>+ BN[MSBN=002] đã nằm 7 ngày ở phòng loại B => VPTL = $7 \cdot (100kđ + 900kđ) = 7000kđ$</p> <p>+ BN[MSBN=003] => VPTL = 50kđ</p> <p>+ BN[MSBN=004] => VPTL = 900kđ</p> <p>=> Tổng viện phí = 50kđ + 7000kđ + 50kđ + 900kđ = 8000kđ</p>

- a) Hãy áp dụng các tính chất hướng đối tượng đã học để vẽ sơ đồ lớp (cả chi tiết các thành phần trong lớp lẫn quan hệ giữa các lớp) cho chương trình thực hiện các việc sau:
- Đọc dữ liệu các bệnh nhân từ tập tin nhật ký hoạt động để lập danh sách bệnh nhân (thay cho việc nhập liệu từ bàn phím như thông thường).
 - In ra màn hình bảng thống kê viện phí của từng bệnh nhân và tổng viện phí bệnh viện thu vào trong tháng.
 - Cho biết bệnh viện có bao nhiêu bệnh nhân nội trú và bao nhiêu bệnh nhân ngoại trú.
- b) Viết chương trình cài đặt cho thiết kế ở câu a. (Cài đặt ngay trong khai báo lớp, không cần tách riêng khai báo lớp với định nghĩa hàm. Được phép sử dụng bộ thư viện chuẩn STL: string, vector, v.v...)
- c) [*] Để việc điều trị hiệu quả hơn, các bác sĩ cần biết được lịch sử điều trị của từng bệnh nhân. Hãy nâng cấp hệ thống để bổ sung thêm hoạt động Lập hồ sơ bệnh án cho bệnh nhân. Hồ sơ bệnh án của mỗi bệnh nhân sẽ ghi lại mọi hoạt động của bệnh nhân đó trong tháng để có thể tra cứu lại các hoạt động khi cần thiết. Bạn sẽ thay đổi hay bổ sung thêm cho các lớp đối tượng như thế nào? Hãy giải thích và minh họa thêm trên sơ đồ lớp ở câu a.

* Lưu ý: Những câu đánh dấu sao [*] là không bắt buộc (tính điểm cộng).