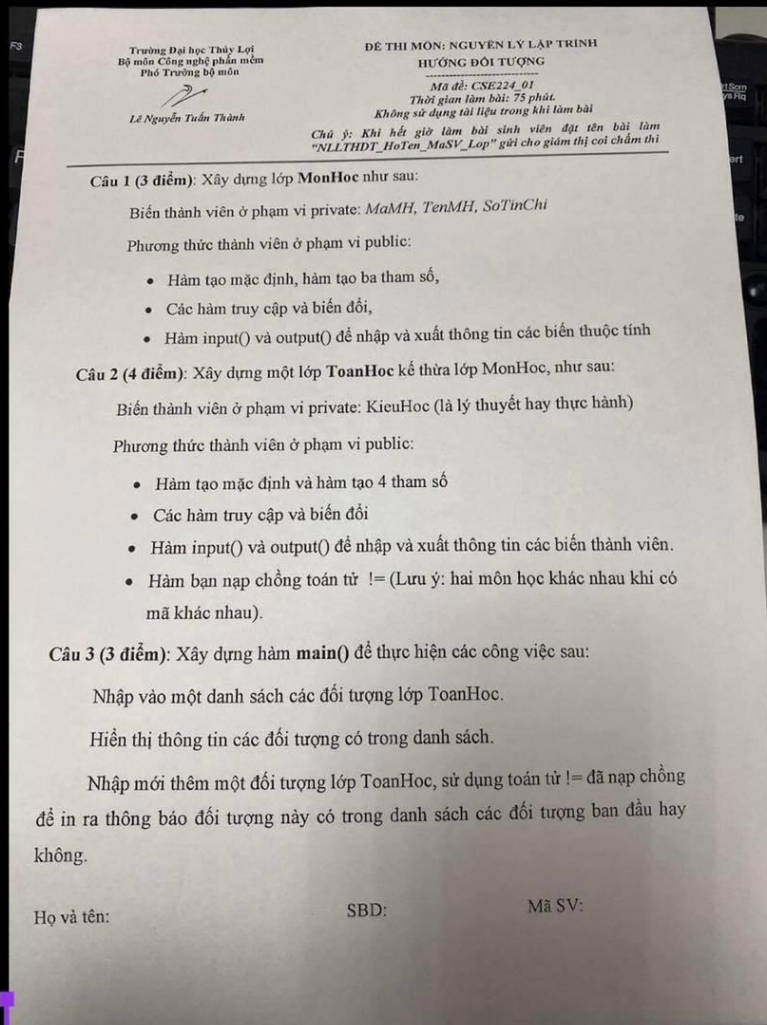


14:05

84%



```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
class MonHoc{
```

```
protected:
```

```
    string MaMH, TenMH;
```



(7 điểm) Xây dựng class CPoint gồm

- Hai thuộc tính, không cho phép truy cập trực tiếp từ bên ngoài class là:
Tung độ x và hoành độ y. **(1 điểm)**
- Và các phương thức sau:
 - o Constructor không tham số, có tham số của CPoint để khởi tạo các thuộc tính. **(1 điểm)**
 - o Nhập một điểm từ bàn phím. **(1 điểm)**
 - o Xuất thông tin của một điểm ra màn hình. **(1 điểm)**
 - o Di chuyển điểm sang phải 1 đơn vị **(1 điểm)**
 - o Khoảng cách giữa hai điểm dùng hàm friend **(1 điểm)**
 - o Kiểm tra 3 điểm thẳng hàng. (Giả sử a, b, c lần lượt là khoảng cách từ A->B, B->C, A->C. Điều kiện để 3 điểm thẳng hàng là $a+b=c$ hoặc $a+c=b$ hoặc $b+c=a$) **(1 điểm)**

(3 điểm) Xây dựng chương trình chính để thực hiện các công việc sau:

- Tạo ra một object (P1) thuộc class CPoint được khởi tạo bởi các giá trị tùy ý và một object khác (P2) nhập từ bàn phím. **(1 điểm)**
- Tính khoảng cách giữa P1 và P2. Di chuyển P2 sang phải 1 đơn vị và xuất thông tin của P2 ra màn hình. **(1 điểm)**
- Tạo ra một object khác (P3), kiểm tra xem 3 điểm: P1, P2(sau khi đã di chuyển), P3 có thẳng hàng hay không. **(1 điểm)**

1/Viết 1 lớp **Vehicles** có một biến thành viên kiểu nguyên chỉ tốc độ tiêu chuẩn của phương tiện , các hàm thành viên bao gồm:

-hàm tạo mặc định và hàm tạo 1 tham số

-Phương thức ảo thuần túy maxSpeed() trả về tốc độ tối đa của phương tiện giao thông (3đ)

2/Viết một lớp **Ferrari** kế thừa lớp **Vehicles** với các biến thành viên: màu sắc kiểu string và hệ số tốc độ kiểu double. Các hàm thành viên bao gồm:

-Hàm tạo mặc định và hàm tạo 3 tham số

-Ghi đè phương thức ảo thuần túy của lớp cha để tính tốc độ tối đa của xe Ferrari = hệ số tốc độ*tốc độ tiêu chuẩn. Lưu ý cần khai báo phương thức là const.

-Hàm input() để nhập thông số của Ferrari và hàm output() để xuất thông tin của Ferrari cũng như tốc độ tối đa ra màn hình

-Nạp chồng toán tử > như là hàm thành viên của lớp để so sánh tốc độ tối đa của hai xe Ferrari (T1>T2: T1 có tốc độ tối đa lớn hơn T2 hay không với T1 và T2 là các đối tượng thuộc lớp Ferrari (4đ)

3/Viết hàm main() để nhập vào một danh sách các đối tượng Ferrari. Sử dụng toán tử > đã được nạp chồng để tìm và in ra đối tượng Ferrari có tốc độ tối đa lớn nhất trong danh sách đã nhập (3đ)

Lê Nguyễn Tuấn Thành

ĐỀ THI MÔN: NGUYÊN LÝ LẬP TRÌNH
HƯỚNG ĐỐI TƯỢNG

Mã đề: CSE224_04

Thời gian làm bài: 75 phút.

Không sử dụng tài liệu trong khi làm bài

Chú ý: Khi hết giờ làm bài sinh viên đặt tên bài làm
"NLLTHDT_HoTen_MaSV_Lop" gửi cho giám thị coi chấm thi

Câu 1 (3 điểm): Xây dựng lớp **Vehicles** có một biến thành viên kiểu nguyên chỉ tốc độ tiêu chuẩn của phương tiện, các phương thức thành viên bao gồm:

- Hàm tạo mặc định và hàm tạo 1 tham số
- Phương thức ảo thuần `maxSpeed()`: trả về tốc độ tối đa của phương tiện giao thông.

Câu 2 (4 điểm): Xây dựng lớp **Ferrari** kế thừa lớp **Vehicles** với các biến thành viên: màu sắc kiểu string và hệ số tốc độ kiểu double. Các phương thức thành viên bao gồm:

- Hàm tạo mặc định và hàm tạo 3 tham số
- Ghi đè phương thức ảo thuần túy của lớp cha để tính tốc độ tối đa của xe Ferrari = hệ số tốc độ * tốc độ tiêu chuẩn. Lưu ý cần khai báo phương thức là `const`.
- Hàm `input()` để nhập thông số của Ferrari và hàm `output()` để xuất thông tin của Ferrari cũng như tốc độ tối đa ra màn hình.
- Nạp chồng toán tử `>` như là hàm thành viên của lớp để so sánh tốc độ tối đa của hai xe Ferrari ($T1 > T2$: $T1$ có tốc độ tối đa lớn hơn $T2$ hay không? với $T1$ và $T2$ là các đối tượng thuộc lớp Ferrari).

Câu 3 (3 điểm): Xây dựng hàm `main()` để nhập vào một danh sách các đối tượng Ferrari. Sử dụng toán tử `>` đã được nạp chồng để tìm và in ra đối tượng Ferrari có tốc độ tối đa lớn nhất trong danh sách đã nhập.

Họ tên:

Bùi Đức

SBD:

Mã SV:

Lê Nguyễn Tuấn Thành

ĐỀ THI MÔN: NGUYÊN LÝ LẬP TRÌNH
HƯỚNG ĐỐI TƯỢNG

Mã đề: CSE224_04

Thời gian làm bài: 75 phút.

Không sử dụng tài liệu trong khi làm bài

Chú ý: Khi hết giờ làm bài sinh viên đặt tên bài làm
"NLLTHDT_HoTen_MaSV_Lop" gửi cho giám thị coi chấm thi

Câu 1 (3 điểm): Xây dựng lớp **Vehicles** có một biến thành viên kiểu nguyên chỉ tốc độ tiêu chuẩn của phương tiện, các phương thức thành viên bao gồm:

- Hàm tạo mặc định và hàm tạo 1 tham số
- Phương thức ảo thuần `maxSpeed()`: trả về tốc độ tối đa của phương tiện giao thông.

Câu 2 (4 điểm): Xây dựng lớp **Ferrari** kế thừa lớp **Vehicles** với các biến thành viên: màu sắc kiểu string và hệ số tốc độ kiểu double. Các phương thức thành viên bao gồm:

- Hàm tạo mặc định và hàm tạo 3 tham số
- Ghi đè phương thức ảo thuần túy của lớp cha để tính tốc độ tối đa của xe Ferrari = hệ số tốc độ * tốc độ tiêu chuẩn. Lưu ý cần khai báo phương thức là `const`.
- Hàm `input()` để nhập thông số của Ferrari và hàm `output()` để xuất thông tin của Ferrari cũng như tốc độ tối đa ra màn hình.
- Nạp chồng toán tử `>` như là hàm thành viên của lớp để so sánh tốc độ tối đa của hai xe Ferrari ($T1 > T2$: $T1$ có tốc độ tối đa lớn hơn $T2$ hay không? với $T1$ và $T2$ là các đối tượng thuộc lớp Ferrari).

Câu 3 (3 điểm): Xây dựng hàm `main()` để nhập vào một danh sách các đối tượng Ferrari. Sử dụng toán tử `>` đã được nạp chồng để tìm và in ra đối tượng Ferrari có tốc độ tối đa lớn nhất trong danh sách đã nhập.

Họ tên:

Bùi Đức

SBD:

Mã SV:

Một đại lý bán các sản phẩm với hai chế độ bán hàng sỉ và lẻ. Bạn hãy xây dựng các lớp sau

- Lớp SANPHAM gồm: (3 điểm)
 - o Thuộc tính private: tên, nhà sản xuất
 - o Phương thức: hàm tạo 2 tham số, các hàm truy cập và biến đổi các biến thành viên, hàm nhập và hiển thị thông tin của đối tượng.
- Lớp BANLE kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: giá bán lẻ, số lượng (<10)
 - o Phương thức: hàm tạo 4 tham số, hàm nhập (chú ý kiểm tra số lượng) và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá bán lẻ * số lượng
- Lớp BANSI kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: giá sỉ, số lượng (>100)
 - o Phương thức: hàm tạo 4 tham số, hàm nhập (chú ý kiểm tra số lượng) và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá sỉ * số lượng

Chương trình chính (2 điểm):

- Nhập danh sách N đối tượng SANPHAM. Chú ý: tùy thuộc đối tượng là BANLE hay BANSI mà sử dụng phù hợp các phương thức (1 điểm)
- Hiển thị thông tin của danh sách N đối tượng vừa nhập (0.5 điểm)
- Hiển thị thông tin sản phẩm có thanh toán ít nhất trong danh sách trên (0.5 điểm) (nếu có nhiều sản phẩm thỏa mãn, in thông tin của sản phẩm xuất hiện đầu tiên).

Một trường học trung học phổ thông có viên chức là cán bộ hành chính và giáo viên.

(2.5 điểm) Xây dựng lớp VIENCHUC

- Mỗi viên chức có các thuộc tính: Mã viên chức, Họ tên, Năm sinh, HSL (Hệ số lương).

- Xây dựng các hàm tạo, hàm nhập, hàm xuất các thông tin của viên chức, hàm ảo tính lương viên chức.

(2.5 điểm) Xây dựng lớp CBHANHCHINH kế thừa từ lớp VIENCHUC

- Mỗi cán bộ hành chính quản lý các thuộc tính: chức danh (như kế toán, thủ quỹ, giáo vụ, bảo vệ,...), phụ cấp ăn trưa.

- Xây dựng các hàm tạo, hàm nhập, hàm xuất các thông tin của cán bộ hành chính.

- Xây dựng hàm tính lương của cán bộ hành chính theo công thức:

$$\text{HSL} * 2000000 + \text{phụ cấp ăn trưa}.$$

(2.5 điểm) Xây dựng lớp GIAOVIEN kế thừa từ lớp VIENCHUC

- Mỗi giáo viên quản lý các thông tin: môn dạy (như toán, văn, vật lý, hóa học,...), trình độ chuyên môn (như đại học, thạc sĩ,...), thâm niên (số năm đã làm giáo viên).

- Xây dựng hàm tạo, hàm nhập, hàm xuất các thông tin của giáo viên.

- Xây dựng hàm tính lương của giáo viên theo công thức:

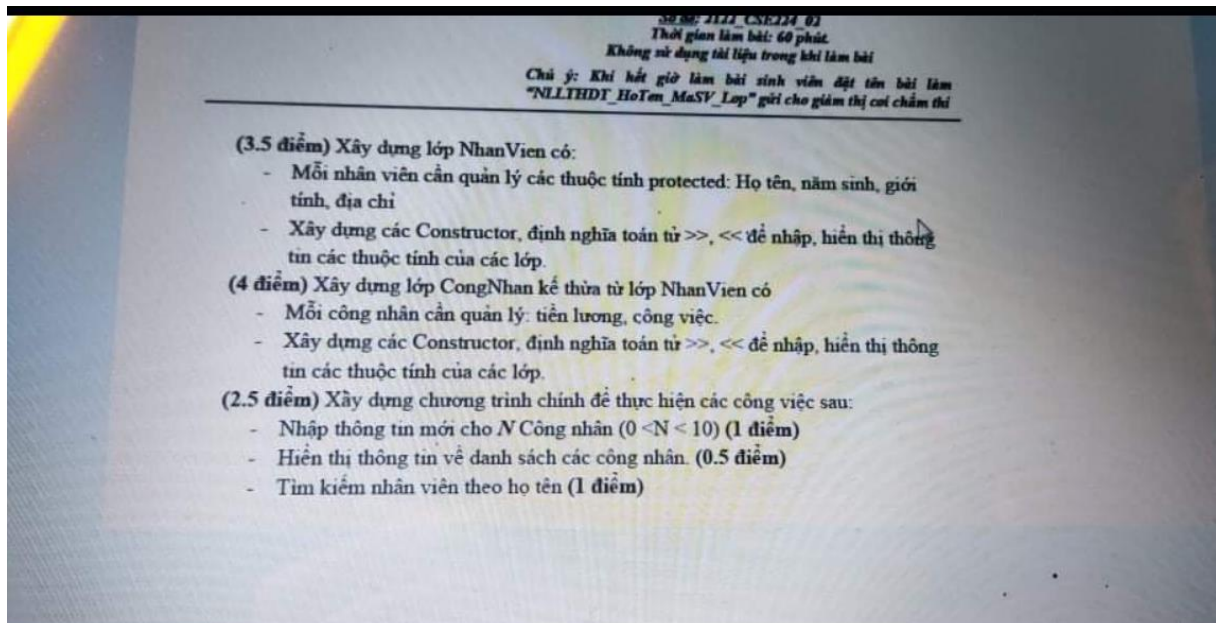
$$\text{HSL} * 2000000 * (130 + \text{thâm niên}) / 100$$

(2.5 điểm) Xây dựng chương trình chính để thực hiện các công việc sau:

- Nhập thông tin mới cho N viên chức (Hỏi người dùng muốn nhập cho cán bộ hành chính hay giáo viên và nhập đúng thông tin cho đối tượng đó). **(1.0 điểm)**

- Hiện thị thông tin về danh sách các viên chức. **(0.5 điểm)**

- Tìm viên chức có lương cao nhất trường. **(1.0 điểm)**



Một cửa hàng tiện ích bán các sản phẩm là đồ hộp hoặc rau quả. Bạn hãy xây dựng các lớp sau

- Lớp SANPHAM gồm: (3 điểm)
 - o Thuộc tính private: tên, giá sản phẩm
 - o Phương thức: hàm tạo 2 tham số, các hàm truy cập và biến đổi các biến thành viên, hàm nhập và hiển thị thông tin của đối tượng.
- Lớp DOHOP kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: số lượng sản phẩm
 - o Phương thức: hàm tạo 3 tham số, hàm nhập và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá * số lượng sản phẩm
- Lớp RAUQUA kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: cân nặng rau quả
 - o Phương thức: hàm tạo 3 tham số, hàm nhập và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá * cân nặng rau quả

Chương trình chính (2 điểm):

- Nhập danh sách N đối tượng SANPHAM. Chú ý: tùy thuộc đối tượng là DOHOP hay RAUQUA mà sử dụng phù hợp các phương thức (1 điểm)
- Hiển thị thông tin của danh sách N đối tượng vừa nhập (0.5 điểm)
- Hiển thị thông tin sản phẩm phải thanh toán nhiều nhất trong danh sách trên (0.5 điểm)
(nếu có nhiều sản phẩm thoả mãn, in thông tin của sản phẩm xuất hiện đầu tiên).

Một cửa hàng tiện ích bán các sản phẩm là đồ hộp hoặc rau quả. Bạn hãy xây dựng các lớp sau

- Lớp SANPHAM gồm: (3 điểm)
 - o Thuộc tính private: tên, giá sản phẩm
 - o Phương thức: hàm tạo 2 tham số, các hàm truy cập và biến đổi các biến thành viên, hàm nhập và hiển thị thông tin của đối tượng.
- Lớp DOHOP kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: số lượng sản phẩm
 - o Phương thức: hàm tạo 3 tham số, hàm nhập và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá * số lượng sản phẩm
- Lớp RAUQUA kế thừa public từ lớp SANPHAM có thêm: (2.5 điểm)
 - o Thuộc tính private: cân nặng rau quả
 - o Phương thức: hàm tạo 3 tham số, hàm nhập và hiển thị thông tin của đối tượng, hàm thanh toán tính theo công thức: giá * cân nặng rau quả

Chương trình chính (2 điểm):

- Nhập danh sách N đối tượng SANPHAM. Chú ý: tùy thuộc đối tượng là DOHOP hay RAUQUA mà sử dụng phù hợp các phương thức (1 điểm)
- Hiển thị thông tin của danh sách N đối tượng vừa nhập (0.5 điểm)
- Hiển thị thông tin sản phẩm phải thanh toán nhiều nhất trong danh sách trên (0.5 điểm)
(nếu có nhiều sản phẩm thoả mãn, in thông tin của sản phẩm xuất hiện đầu tiên).

Kiểm tra giữa kỳ

Thời gian: 60 phút

Xây dựng lớp **Point** gồm:

- Thuộc tính: x, y kiểu nguyên biểu diễn hoành độ và tung độ của điểm

Phương thức:

- Hàm tạo không tham số
- Hàm tạo hai tham số
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- CheoChinh : hàm kiểm tra Point có thuộc đường chéo chính hay không (1 điểm thuộc đường chéo chính khi và chỉ khi tung độ bằng hoành độ).

Xây dựng lớp **Pixel** kế thừa từ lớp Point bao gồm thêm:

- Thuộc tính: TenMau

Phương thức:

- Hàm tạo không tham số
- Hàm tạo ba tham số
- Nạp chồng toán tử nhập
- Nạp chồng toán tử xuất
- KiemTra: hàm kiểm tra Pixel thuộc đường chéo chính và có màu "Xanh" hay không?

Chương trình chính: Nhập vào từ bàn phím n Pixel (n nhập từ bàn phím). Hiển thị thông tin các Pixel thuộc đường chéo chính và có màu xanh.



Trường Đại học Thủy Lợi
Bộ môn Công nghệ phần mềm
Phó Trưởng bộ môn

Lê Nguyễn Tuấn Thành

ĐỀ THI MÔN: NGUYÊN LÝ LẬP TRÌNH
HƯỚNG ĐỐI TƯỢNG

Số đề: CSE224_03

Thời gian làm bài: 75 phút.

Không sử dụng tài liệu trong khi làm bài

Chú ý: Khi hết giờ làm bài sinh viên đặt tên bài làm
"SBD_HoTen_MaSV_Lop" gửi cho giám thị coi chấm thi

Câu 1 (2.5 điểm): Xây dựng lớp `VeMayBay` (Vé máy bay) như sau:
Thuộc tính: `TenChuyen`, `NgayBay`, `GiaVe`
Phương thức công khai:

- Phương thức khởi tạo không tham số và phương thức khởi tạo 3 tham số
- `Nhap()`: phương thức nhập giá trị cho các thuộc tính
- `Xuat()`: phương thức in giá trị các thuộc tính ra màn hình
- `getGiaVe()`: hàm trả về giá vé
- `setGiaVe()`: phương thức thay đổi giá vé

Câu 2 (2.0 điểm): Xây dựng lớp `Nguoi` (Người) như sau:
Thuộc tính: `HoTen`, `GioiTinh`, `Tuoi`

Phương thức công khai:

- Phương thức khởi tạo không tham số và phương thức khởi tạo 3 tham số
- `Nhap()`: phương thức nhập giá trị cho các thuộc tính
- `Xuat()`: phương thức in giá trị các thuộc tính ra màn hình

Câu 3 (3.5 điểm): Xây dựng lớp `HanhKhach` (Hành khách) kế thừa lớp `Nguoi`, và bổ sung thêm:

Thuộc tính: `VeMayBay*`, `SoLuong` (mỗi hành khách được mua nhiều vé)

Phương thức công khai:

- Phương thức khởi tạo không tham số và phương thức khởi tạo 3 tham số
- `Nhap()`: định nghĩa lại phương thức `Nhap()` của lớp `Nguoi`
- `Xuat()`: định nghĩa lại phương thức `Xuat()` của lớp `Nguoi`
- `tinhTongTien()`: trả về tổng số tiền phải trả của hành khách

Câu 4 (2.0 điểm): Xây dựng chương trình chính như sau:

Nhập vào một danh sách N hành khách (với N nhập từ bàn phím, $0 < N < 10$).

Hiển thị danh sách hành khách và số tiền phải trả tương ứng của mỗi khách hàng

Câu 1 (2.5 điểm): Xây dựng lớp NhanVien (Nhân Viên), như sau:

- Mỗi nhân viên có 4 thuộc tính trong khu vực được bảo vệ (protected) gồm: *họ tên, năm sinh, giới tính, hơng*
- Cài đặt các hàm tạo: hàm tạo không tham số và hàm tạo với 4 tham số
- Định nghĩa toán tử >>, << để nhập, hiển thị giá trị cho các thuộc tính của lớp

Câu 2 (2.5 điểm): Xây dựng lớp LapTrinhVien (Lập trình viên) kế thừa từ lớp NhanVien, như sau:

- Mỗi lập trình viên có thêm 1 thuộc tính riêng tư: *ngôn ngữ lập trình*
- Xây dựng các hàm tạo: hàm tạo không tham số và hàm tạo với 5 tham số
- Định nghĩa toán tử >>, << để nhập, hiển thị giá trị các thuộc tính của lớp

Câu 3 (2.5 điểm): Xây dựng lớp KiemThuVien (Kiểm thử viên) kế thừa từ lớp NhanVien, như sau:

- Mỗi kiểm thử viên có thêm 1 thuộc tính riêng tư: *chuyên môn* (kiểm thức giao diện hay kiểm thử chức năng)
- Xây dựng các hàm tạo: hàm tạo không tham số và hàm tạo với 5 tham số
- Định nghĩa toán tử >>, << để nhập, hiển thị thông tin các thuộc tính của lớp

Câu 4 (2.5 điểm): Xây dựng chương trình chính để thực hiện các công việc sau:

- Nhập thông tin cho N nhân viên ($0 < N < 10$) từ bàn phím. Khi nhập thông tin, chương trình sẽ hỏi người dùng muốn nhập thông tin cho lập trình viên hay kiểm thử viên và sau đó nhập đúng thông tin cho đối tượng của lớp đó.
- Hiển thị thông tin danh sách các nhân viên vừa nhập.
- Tìm kiếm nhân viên theo họ tên.

Câu 1.(2đ) Viết một class **HinhKhoi** với các biến thành viên ở phạm vi **private**: **MaHK**, **TenHK**; các hàm thành viên ở phạm vi **public**: hàm tạo mặc định, hàm tạo hai tham số, các hàm truy cập và biến đổi biến thành viên, hàm **input()** và hàm **output()** để nhập và xuất thông tin. Lớp còn có hai hàm thành viên có tên là **tinhChuVi()** và **tinhDienTich()** để trả về chu vi và diện tích của hình khối, tuy nhiên định nghĩa hiện tại của các hàm này chỉ bao gồm một thông điệp lỗi và câu lệnh thoát khỏi chương trình.

Câu 2.(2đ) Viết một class **HinhChuNhat** kế thừa lớp **HinhKhoi** với các biến thành viên ở phạm vi **private**: **chieuRong**, **chieuDai**; các hàm thành viên ở phạm vi **public**: hàm tạo mặc định, hàm tạo bốn tham số, các hàm truy cập và biến đổi biến thành viên, hàm **input()** và **output()** để nhập và xuất thông tin. Định nghĩa lại hai hàm **tinhChuVi()** và **tinhDienTich()** của lớp cha để chúng trả về chu vi và diện tích của hình chữ nhật.

Câu 3.(2đ) Viết một class **HinhTron** được kế thừa từ lớp **HinhKhoi** với biến thành viên ở phạm vi **private**: **banKinh**; các hàm thành viên ở phạm vi **public**: hàm tạo mặc định, hàm tạo ba tham số, các hàm truy cập và biến đổi biến thành viên, hàm **input()** và **output()** để nhập và xuất thông tin. Định nghĩa lại hai hàm **tinhChuVi()** và **tinhDienTich()** của lớp cha để chúng trả về chu vi và diện tích của hình tròn **HinhTron**.

Câu 4.(1.5đ) Viết các hàm bạn của class **HinhTron** gồm: hàm nạp chồng toán tử **>** để kiểm tra xem $ht1 > ht2$ hay không?; hàm nạp chồng toán tử **=** để kiểm tra xem $ht1 = ht2$ hay không? (với $ht1$, $ht2$ là hai đối tượng của lớp **HinhTron**, và $ht1 > ht2$ nếu diện tích $ht1$ lớn hơn diện tích $ht2$, $ht1 = ht2$ nếu diện tích $ht1$ bằng diện tích $ht2$).

Câu 5.(2.5đ) Viết chương trình cho phép:

- Khai báo hai đối tượng hình chữ nhật bằng hàm tạo mặc định và hàm tạo bốn tham số. Hiển thị thông tin của các hình chữ nhật vừa nhập.
- Nhập thông tin của một số hình tròn và lưu trữ vào một danh sách (mảng hoặc vector,...). Hiển thị thông tin của các hình tròn vừa nhập. Sử dụng các toán tử đã nạp chồng để tìm và hiển thị thông tin của các hình tròn có diện tích lớn nhất.