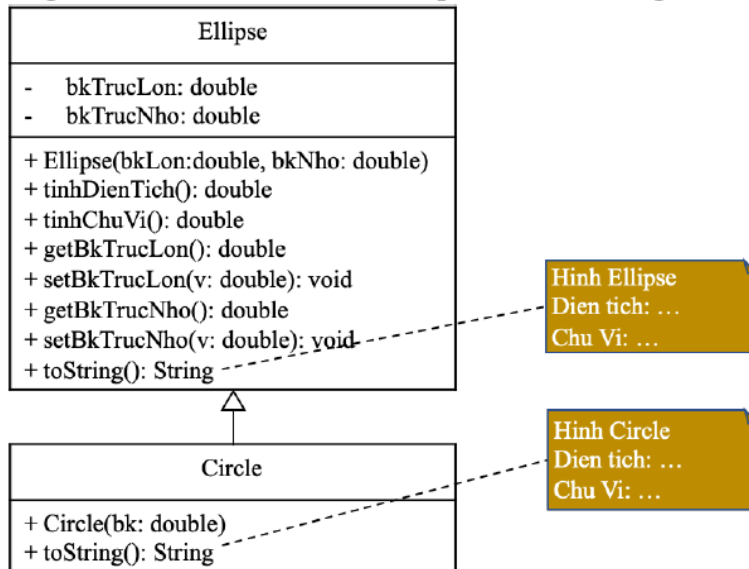


BÀI TẬP THỰC HÀNH 3

BT1: Viết chương trình hiện thực hóa sơ đồ lớp bên dưới, trong đó:



Lớp `Ellipse` có 2 thuộc tính: bán kính trục lớn, bán kính trục nhỏ, và các phương thức:

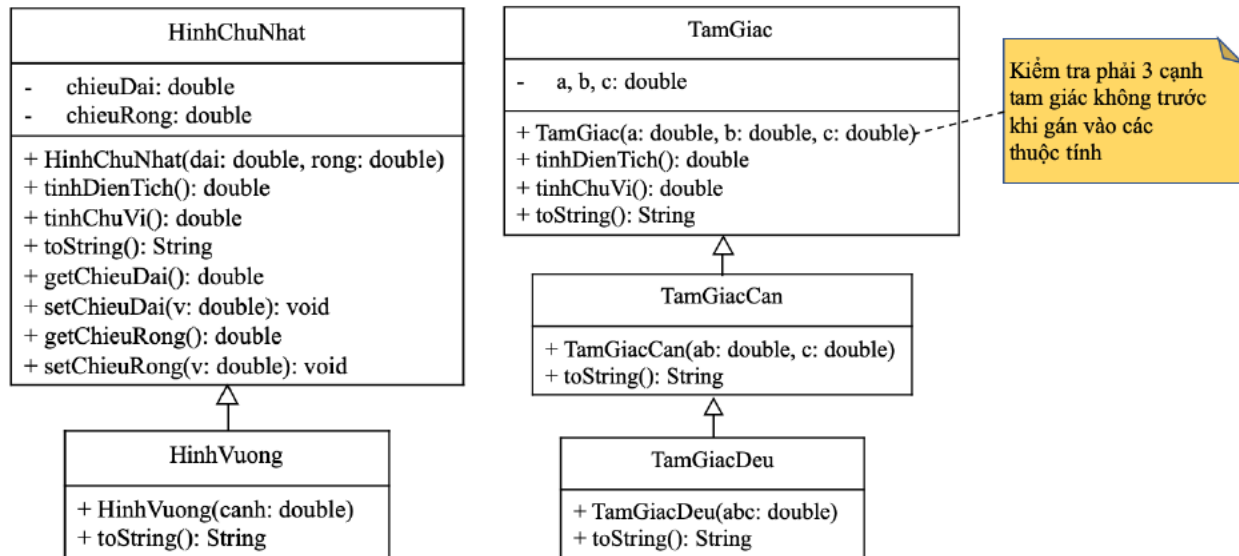
- Phương thức khởi tạo 2 tham số khởi gán cho bán kính hai trục.
- Các phương thức getter và setter của các thuộc tính.
- Ghi đè phương thức `toString()` trả về chuỗi đại diện cho một `Ellipse` có định dạng như phần ghi chú trong hình.
- Phương thức tính diện tích và chu vi của hình `Ellipse`. Biết công thức tính như sau:
 - Diện tích = $PI \times bkTrucLon \times bkTrucNho$
 - Chu vi = $2 \times PI \times \sqrt{\frac{bkTrucLon^2 + bkTrucNho^2}{2}}$

Lớp `Circle` kế thừa lớp `Ellipse` có các phương thức sau:

- Phương thức khởi tạo 1 tham số để khởi gán bán kính hình tròn.
- Ghi đè phương thức `toString()` trả về chuỗi đại diện cho một `Circle` có định dạng như phần ghi chú trong hình.

Viết phương thức `main()` để kiểm tra các chức năng trong các lớp.

BT2: Tương tự bài 1, hiện thực hóa 2 sơ đồ lớp bên dưới. Biết công thức tính diện tích của tam giác khi biết 3 cạnh a, b, c là: $S = \sqrt{p(p-a)(p-b)(p-c)}$. Trong đó, $p = \frac{a+b+c}{2}$ là nửa chu vi.



BT3: Một công ty chuyên sản xuất sách và băng đĩa phục vụ học tập trong nhà trường phổ thông. Mỗi sản phẩm đều có mã sản phẩm, tên sản phẩm, mô tả sản phẩm, nhà sản xuất, giá bán. Ngoài ra:

- Sách có thêm thông tin số trang.
- Băng đĩa có thêm thông tin thời lượng.

Xây dựng các lớp sản phẩm trên. Sau đó, xây dựng lớp quản lý danh sách sản phẩm tại công ty có các chức năng sau:

- Nhập thêm sản phẩm vào danh sách.
- Xóa sản phẩm khỏi danh sách.
- Xuất thông tin sản phẩm (bao gồm đầy đủ các thuộc tính của từng loại sản phẩm, mỗi thuộc tính hiển thị trên một dòng).
- Tìm kiếm thông tin sản phẩm thông qua tên, mô tả, loại sản phẩm.
- Sắp xếp danh sách theo thứ tự giá bán giảm dần.

Viết phương thức `main()` chứa menu sau để người dùng chọn các chức năng thực hiện:

1. Thêm sản phẩm.
2. Xóa sản phẩm theo mã sản phẩm.
3. Cập nhật tên sản phẩm, mô tả sản phẩm, hoặc giá bán dựa trên mã sản phẩm.
4. Tìm kiếm sản phẩm theo mã sản phẩm, tên sản phẩm, hoặc khoảng giá bán.
5. Sắp xếp các sản phẩm giảm dần theo giá.
6. Xem danh sách sản phẩm.
7. Thoát.

BT4: Viết chương trình quản lý thông tin tiền lương nhân viên làm việc thời vụ của công ty XYZ, biết một nhân viên bao gồm các thông tin số mã số, chứng minh nhân dân, họ tên, quê

quán, giới tính, số điện thoại liên lạc. Trong đó, mã số được tạo tự động theo cú pháp sau: NV[xxxx], với xxxx là số thứ tự tăng dần từ 1 đến 9999. VD: NV0123.

Nhân viên bình thường được tính lương bằng $\langle \text{lương} \rangle * \langle \text{số ngày công} \rangle$. Tuy nhiên nhân viên thuộc một trong 3 bộ phận A, B hoặc C được tính như sau:

- Bộ phận A: $1.2 * \langle \text{lương} \rangle * \langle \text{số ngày công} \rangle$
- Bộ phận B: $1.5 * \langle \text{lương} \rangle * \langle \text{số ngày công} \rangle$
- Bộ phận C: $2.5 * \langle \text{lương} \rangle * \langle \text{số ngày công} \rangle$

Ngoài ra, mã số của nhân viên thuộc 3 bộ phận này cũng có điểm khác. Ví dụ:

- Bộ phận A: NVA0001
- Bộ phận B: NVB0002
- Bộ phận C: NVC0003

Trong đó, $\langle \text{lương} \rangle$ là mức lương chuẩn cho mỗi ngày làm, quy định công ty là 100.000 VNĐ/ngày và nhân viên sẽ được điểm danh mỗi ngày.

Chương trình bao gồm các chức năng sau:

- Thêm và xóa nhân viên mới của một bộ phận trong danh sách nhân viên.
- Hiện thị danh sách nhân viên, mỗi nhân viên hiện thị các thông tin: họ tên, số chứng minh nhân dân, giới tính, quê quán, số điện thoại, bộ phận.
- Tra cứu nhân viên theo số chứng minh nhân dân, họ tên hoặc quê quán.
- Tính tiền lương cho các nhân viên. Trong phương thức này, người dùng sẽ nhập số ngày công của mỗi nhân viên và hiện thị mã số, họ tên, số ngày công và tiền lương của từng nhân viên.

Viết phương thức `main()` chứa menu sau để người dùng chọn chức năng thực hiện:

1. Thêm nhân viên
2. Xóa nhân viên
3. Tra cứu nhân viên
4. Hiện thị danh sách nhân viên
5. Tính lương nhân viên
6. Thoát

Sau một thời gian phát triển chương trình, công ty XYZ yêu cầu mở thêm bộ phận D với cách tính lương như sau: $\langle \text{lương} \rangle * \langle \text{số ngày làm} \rangle + \langle \text{phụ cấp} \rangle$, tùy vào năng lực

của mỗi nhân viên lúc phỏng vấn sẽ có mức lương phụ cấp khác nhau.

Hãy cập nhật chương trình đã viết nhưng hạn chế tối đa sửa mã nguồn để thêm bộ phận D, cũng như quản lý tiền lương cho các nhân viên thuộc bộ phận D.

BT5: Xây dựng lớp `VanDongVien` thể hiện thông tin một vận động viên thể thao. Lớp gồm có các thuộc tính private sau: họ tên (ví dụ: Dang Van Lam), môn thi đấu (ví dụ: Bong Da).

Xây dựng lớp `VDVBongDa` kế thừa từ `VanDongVien` với việc bổ sung thêm các thuộc tính là: vị trí đá bóng (VD: `Hau Ve`), số áo (VD: `12`), CLB chủ quản (VD: `CLB Ha Noi`).

Xây dựng lớp `VDVBoi` kế thừa từ `VanDongVien` với việc bổ sung thêm các thuộc tính là: môn sở trường (VD: `100m tu do`), CLB chủ quản (VD: `TpHCM`)

Viết lớp `Test` với hàm `main()` thực hiện các yêu cầu sau:

- Khai báo một đối tượng `ArrayList` tên `dsVDV` chứa các đối tượng thuộc lớp `VanDongVien` để quản lý danh sách các vận động viên tham dự Seagames.
- Mở tập tin `VanDongVien.txt` để đọc và tạo các đối tượng phù hợp để thêm vào `dsVDV`.
- Xây dựng menu cho phép người dùng thực hiện các chức năng sau:
 1. In ra danh sách các vận động viên.
 2. Nhập tên một môn thể thao, in ra danh sách các vận động viên của môn đó.
 3. Nhập vị trí thi đấu trong môn bóng đá, in ra các cầu thủ đá ở vị trí này.
 4. Nhập tên CLB, in ra các cầu thủ thuộc CLB này.
 5. Nhập cự ly và kiểu bơi. In ra các VDV bơi lội thi đấu ở cự ly này.
 6. Kết thúc.

BT6: Xây dựng lớp `CongDan` để thể hiện các thông tin của một công dân, với các thuộc tính private như: họ tên, số định danh cá nhân, ngày sinh, tháng sinh, năm sinh, nơi sinh, nghề nghiệp. Lớp có các phương thức sau:

- Phương thức khởi tạo có tham số khởi gán cho các thuộc tính của công dân.
- Phương thức tính tuổi công dân.
- Ghi đè phương thức `toString()` để trả về chuỗi chứa các thông tin của công dân (mỗi thuộc tính in trên một dòng).

Xây dựng lớp `SinhVien` kế thừa từ lớp `CongDan`. Trong đó, thuộc tính nghề nghiệp luôn có giá trị là “Sinh viên”. Ngoài ra, lớp có thêm 2 thuộc tính khác là ngành học và trường đại học.

Xây dựng lớp `HocSinh` kế thừa từ lớp `CongDan`. Trong đó, thuộc tính nghề nghiệp luôn có giá trị là “Học sinh” và vùng số định danh cá nhân sẽ được lưu mã số học sinh. Ngoài ra, lớp có thêm các thuộc tính khác là điểm Văn, điểm Toán, Ngoại ngữ, và xếp loại hạnh kiểm.

Xây dựng lớp `Test` với hàm `main()` thực hiện các yêu cầu sau:

- Khai báo một đối tượng `ArrayList` tên `dsCongDan` chứa các đối tượng thuộc lớp `CongDan` để quản lý danh sách công dân.
- Mở tập tin `SV-HS-CD.txt` để đọc, sau đó khai báo các đối tượng phù hợp (`CongDan/SinhVien/HocSinh`) rồi thêm vào `dsCongDan`. Lưu ý: Cần tách ngày

sinh (gồm ngày-tháng-năm) thành 3 vùng riêng biệt. Với dòng nào là học sinh, không cần lưu ngày sinh, tháng sinh, năm sinh, và nơi sinh.

- Xây dựng menu cho phép người dùng thực hiện các chức năng sau:
 1. Nhập một số nguyên từ 1 đến 3. Sau đó, in ra danh sách tương ứng, với `CongDan` là 1, `SinhVien` là 2, `HocSinh` là 3.
 2. Nhập một chữ cái, in ra các đối tượng có **tên** bắt đầu bằng chữ cái đó.
 3. Nhập một tháng từ 1 đến 12, in ra các đối tượng sinh trong tháng đó.
 4. Nhập tên một tỉnh thành, in ra các đối tượng có nơi sinh tương ứng.
 5. Kết thúc.

BT7: Một hệ thống ngân hàng gồm 2 loại tài khoản chính là tài khoản có kỳ hạn và tài khoản không kỳ hạn. Một tài khoản bao gồm các thông tin số tài khoản, tên tài khoản, điện thoại, email, số tiền tài khoản, ngày tạo tài khoản, trạng thái tài khoản. Riêng tài khoản có kỳ hạn có thêm thông tin kỳ hạn và ngày đáo hạn. Trong đó số tài khoản là chuỗi có 6 số tính từ 00001 và tăng dần khi tạo tài khoản mới. Thông tin kỳ hạn cho phép được lấy từ enum sau:

```
enum KyHan {  
    MOT_TUAN,  
    MOT_THANG,  
    MOT_NAM,  
}
```

Viết chương trình thực hiện các chức năng:

- Xem thông tin tài khoản gồm thông tin số tài khoản, tên tài khoản, số tiền và loại tài khoản, nếu là tài khoản có kỳ hạn thì hiển thị thêm thông tin kỳ hạn và ngày đáo hạn.
- Tra cứu tài khoản theo số tài khoản và tên tài khoản.
- Tính tiền lãi khi đến ngày đáo hạn, tiền lãi được cộng trực tiếp số tiền trong tài khoản.

Loại tài khoản		Lãi suất
Không kỳ hạn		0.1%/năm
Có kỳ hạn	1 tuần	0.5%/năm
	1 tháng	4.5%/năm
	1 năm	6.8%/năm

- Nộp tiền vào tài khoản được nạp ngay đối với tài khoản không kỳ hạn, nhưng tài khoản có kỳ hạn chỉ được nạp vào trong ngày đáo hạn.
- Rút tiền thì được rút ngay đối với tài khoản không kỳ hạn, nhưng tài khoản có kỳ hạn chỉ được rút trong ngày đáo hạn, nếu người dùng vẫn rút trong ngày đáo hạn thì tài khoản đó sẽ bị tắt toán (trở thành tài khoản không kỳ hạn) và không có tiền lãi trong kỳ hạn hiện tại.
- Cập nhật ngày đáo hạn đối với tài khoản có kỳ hạn.