

Interface

Mục tiêu

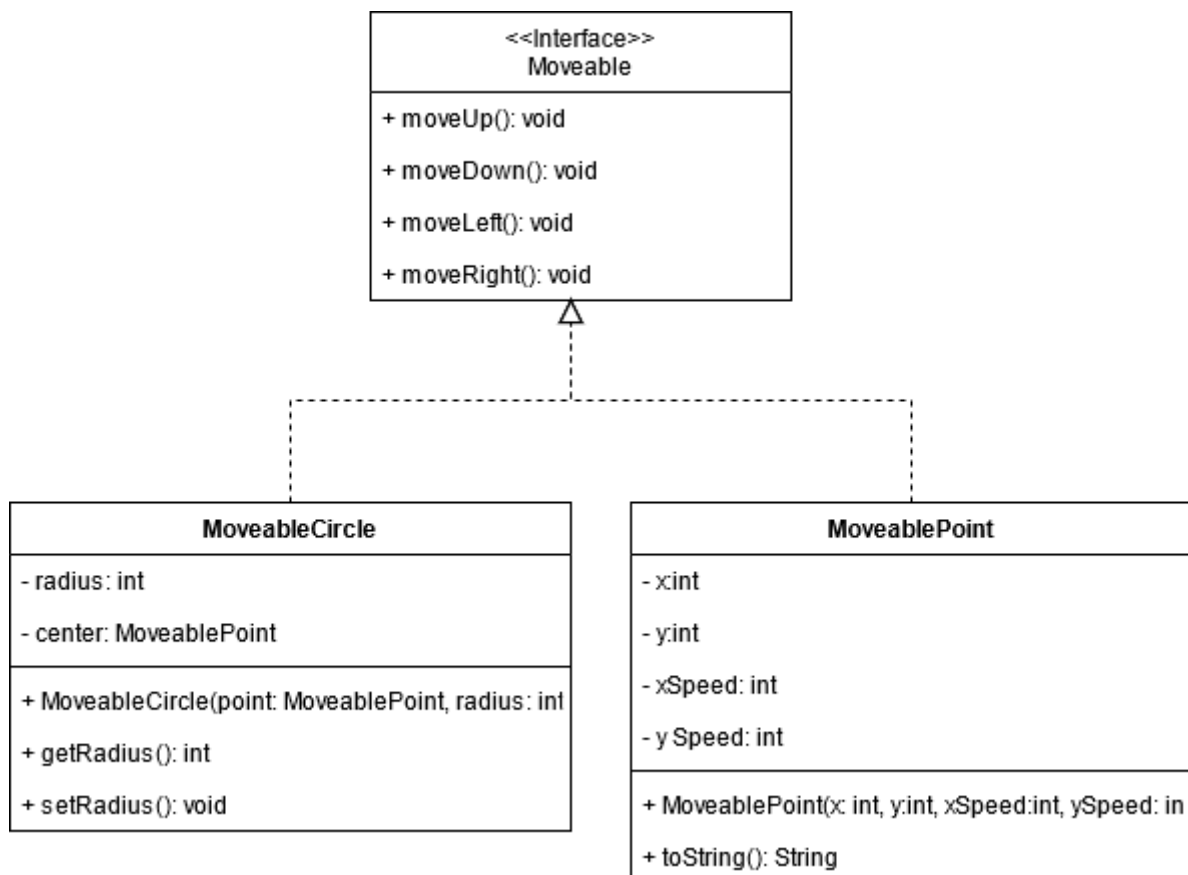
Kết thúc bài thực hành này, sinh viên có khả năng:

- ✓ Biết cách xác định và định nghĩa các hành vi phổ quát (giống nhau) của các đối tượng bằng cách sử dụng Interface
- ✓ Biết cách định nghĩa Interface và định nghĩa các lớp hiện thực Interface
- ✓ Biết cách xác định được kiểu dữ liệu của đối tượng ở thời điểm thực thi
- ✓ Phân biệt được sự khác nhau giữa Abstract Class và Interface và biết cách vận dụng vào những tình huống cụ thể
- ✓ Biết cách vận dụng khả năng đa thừa kế trong Interface

Dùng phương pháp lập trình hướng đối tượng, các em hãy giải quyết các bài tập sau đây:

Bài 1: (+2 điểm)

Hãy xây dựng interface Moveable và các lớp hiện thực interface này là MoveablePoint và MoveableCircle theo sơ đồ cây thừa kế như sau:



Mô tả:

❖ Interface Moveable

- Thể hiện khả năng di chuyển của một lớp đối tượng
- Chứa các phương thức thuần ảo: moveUp, moveDown, moveLeft, moveRight có nghĩa là di chuyển một đối tượng đi lên, đi xuống, sang trái, và sang phải

❖ Class MoveableCircle

- Biểu diễn các đối tượng hình tròn có thể di chuyển được
- Đặc trưng bởi các thuộc tính riêng: radius (bán kính), center (tâm hình tròn). Trong đó center là một đối tượng thuộc kiểu MoveablePoint, thể hiện mối quan hệ has-a (một MoveableCircle có tâm là một MoveablePoint)
- Các phương thức riêng của lớp:
 - o MoveableCircle(): khởi tạo giá trị cho các thuộc tính
 - o getRadius(): phương thức getter của thuộc tính radius
 - o setRadius(): phương thức setter của thuộc tính radius

❖ Class MoveablePoint

- Biểu diễn các đối tượng điểm có thể di chuyển được
- Đặc trưng bởi các thuộc tính: x, y, xSpeed (khoảng cách theo phương ngang mà mỗi lần điểm có thể di chuyển được), ySpeed (khoảng cách theo phương đứng mà mỗi lần điểm có thể di chuyển được)
- Các phương thức riêng của lớp:
 - o MoveablePoint(): khởi tạo giá trị cho các thuộc tính
 - o toString(): thể hiện thông tin về đối tượng theo dạng
 - o MoveablePoint[x = ?, y = ?, xSpeed = ?, ySpeed = ?]

Bài 2: (+2 điểm)

Xây dựng một lớp Octagon (hình bát giác) kế thừa từ lớp GeometricObject (hình đa giác) được cho như bên dưới và hiện thực các interface Comparable và Cloneable do Giảng viên cung cấp.

Mô tả lớp trừu tượng GeometricObject:

<i>GeometricObject</i>
color: String # filled: boolean # dateCreated: java.util.Date
+ GeometricObject() + GeometricObject(color: String, filled: boolean) + getColor(): String + setColor(color: String): void + isFilled(): boolean + setFilled(filled: boolean): void + getDateCreated(): java.util.Date + toString(): String (GeometricObject[color=red, filled = true, dateCreated = "09/05/2020 7:36:47"]) + getArea(): double + getPerimeter(): double

Giả sử rằng 8 cạnh của hình bát giác đều bằng nhau, khi đó diện tích của hình được tính theo công thức:

$$\text{Diện tích} = (2 + 4\sqrt{2}) * \text{cạnh} * \text{cạnh}$$

Sau khi xây dựng các lớp Octagon và GeometricObject hiện thực các interface Comparable và Cloneable nói trên, hãy tạo ra trong chương trình một đối tượng Octagon với độ dài cạnh bằng 5, tính chu vi và diện tích của hình. Sau đó, tạo ra một đối tượng Octagon mới bằng

cách sử dụng phương thức `clone()` và so sánh hai đối tượng này bằng phương thức `compareTo()`

Ví dụ: hai hình ngũ giác có độ dài cạnh bằng nhau nhưng là hai hình khác nhau

