BÀI 4: Interface trong ngôn ngữ lập trình Java và Unified Modeling Language (UML)

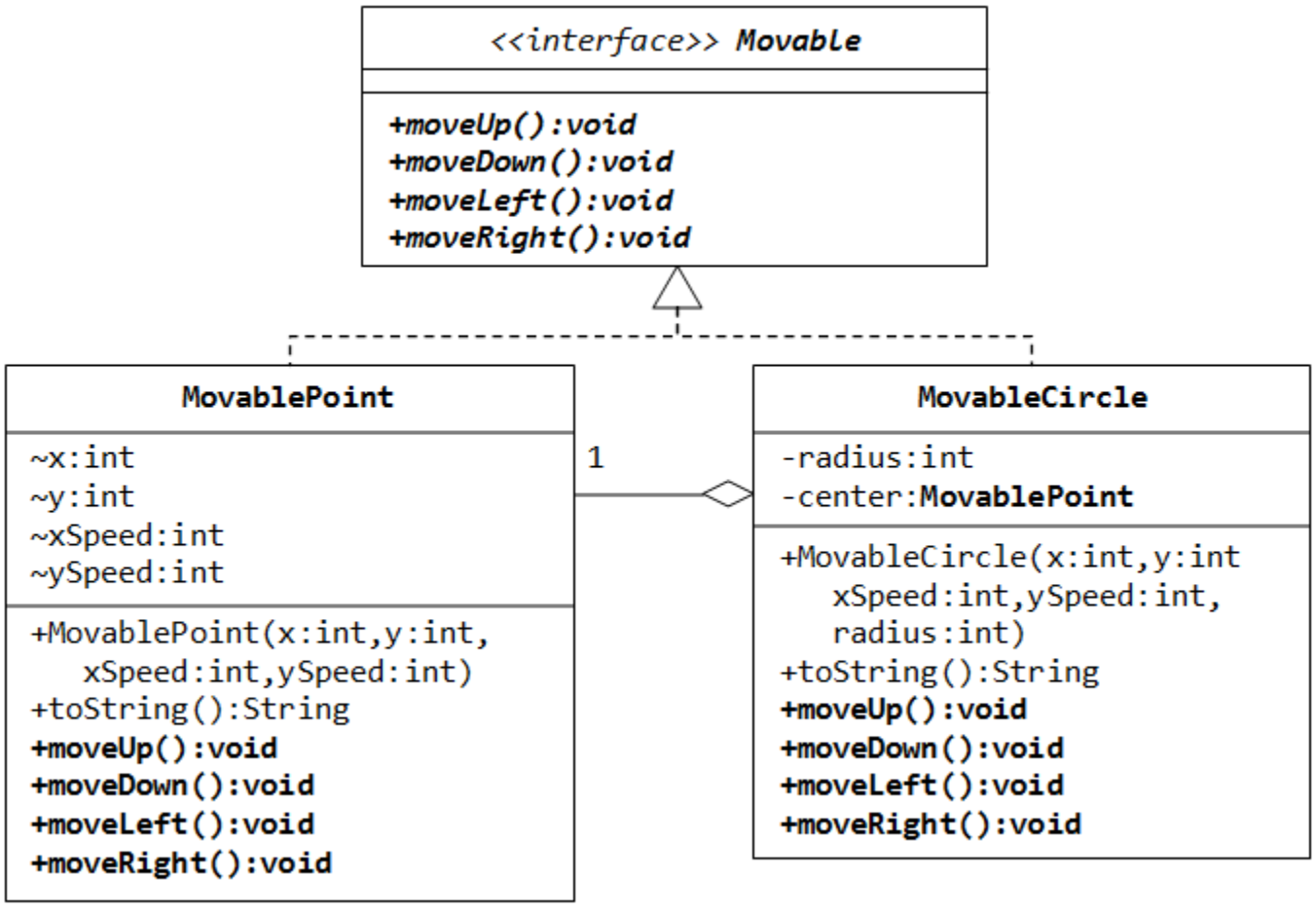
1.MỤC ĐÍCH

Thông qua bài thực hành này, sinh viên có thể hiểu được:

- Khái niệm cơ bản về Interface trong lập trình hướng đối tượng, cách khai báo một Interface, cách sử dụng Interface, tại sao sử dụng Interface.
- Khái niệm về Unified Modeling Language (UML), tại sao sử dụng Interface, phân tích và thiết kế chương trình sử dụng các biểu đồ UML.

2.BÀI TẬP

Bài 1 (+0.5): Giả sử ta có những đối tượng với những cách di chuyển giống nhau: Move Up, Down, Left và Right, Việc di chuyển phụ thuộc vào mỗi đối tượng. Hãy viết 2 class riêng biệt, MovablePoint và MovableCircle – implement Movable Interface và chương trình kiểm tra.



Biết code của Movable Interface là:

```
public interface Movable {  // saved as "Movable.java"
```

```
   public void moveUp();
   ......
}
```

Với lớp MovablePoint, khai báo các thuộc tính với phạm vi chỉ cho truy cập trong package với dấu "~" . Còn với lớp MovableCircle, phải sử dụng lớp MovablePoint để hiển thị tâm của nó.

```
public class MovablePoint implements Movable { // saved as "MovablePoint.java"
   // instance variables
   int x, y, xSpeed, ySpeed;     // package access

   // Constructor
   public MovablePoint(int x, int y, int xSpeed, int ySpeed) {
      this.x = x;
      ......
   }
   ......

   // Implement abstract methods declared in the interface Movable
   @Override
   public void moveUp() {
      y += ySpeed;   // y-axis pointing up for 2D graphics
   }
   ......
}
public class MovableCircle implements Movable { // saved as "MovableCircle.java"
   // instance variables
   private MovablePoint center;   // can use center.x, center.y directly
                                  //  because they are package accessible
   private int radius;

   // Constructor
   public MovableCircle(int x, int y, int xSpeed, int ySpeed, int radius) {
      // Call the MovablePoint's constructor to allocate the center instance.
      center = new MovablePoint(x, y, xSpeed, ySpeed);
      ......
   }
   ......

   // Implement abstract methods declared in the interface Movable
   @Override
   public void moveUp() {
      center.y += center.ySpeed;
   }
   ...
```

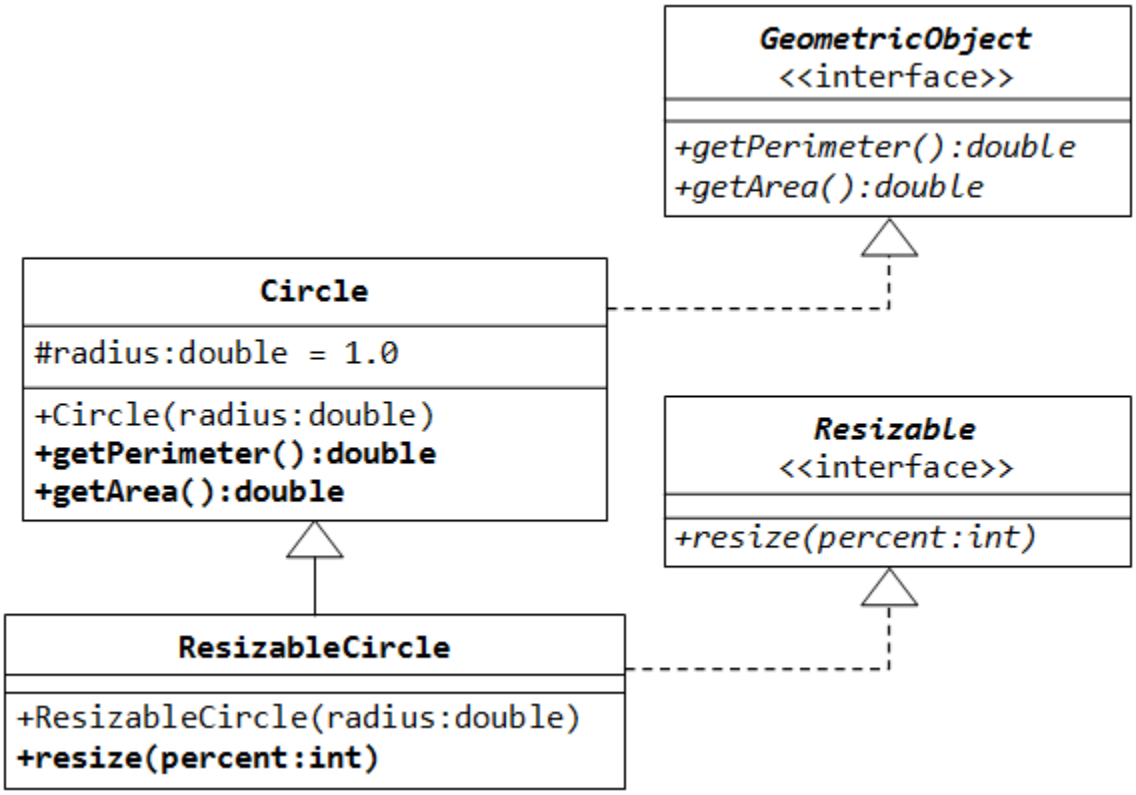
Viết chương trình test như thế này:

```
Movable m1 = new MovablePoint(5, 6, 10, 15);     // upcast
System.out.println(m1);
m1.moveLeft();
```

```
System.out.println(m1);

Movable m2 = new MovableCircle(1, 2, 3, 4, 20);  // upcast
System.out.println(m2);
m2.moveRight();
System.out.println(m2);
```

Bài 2 (+0.5): Viết chương trình kiểm tra `TestResizableCircle`



Biết

```
public interface GeometricObject {
   public double getPerimeter();
   ......
}

public class Circle implements GeometricObject {
   // Private variable
   ......

   // Constructor
   ......

   // Implement methods defined in the interface GeometricObject
   @Override
   public double getPerimeter() { ...... }
```

```
   ......
}

public interface Resizable {
   public double resize(...);
}
public class ResizableCircle extends Circle implements Resizeable {

   // Constructor
   public ResizableCircle(double radius) {
      super(...);
   }

   // Implement methods defined in the interface Resizable
   @Override
   public double resize(int percent) { ...... }
}
```