

Conceptos avanzados de POO en Java

Pedro O. Pérez M., PhD.

Desarrollo de videojuegos con Java
Tecnológico de Monterrey

pperezm@tec.mx

10-2023

Conceptos avanzados de POO en Java

- Interfaces

- Clases abstractas

- Clases internas

Examen Rápido 3

- ▶ Java incorpora una construcción del lenguaje, mediante la declaración `interface`, que permite enunciar un conjunto de constantes y de cabeceras de métodos abstractos; éstos deben implementarse en las clases y constituyen la interfaz de la clase. En cierto modo, es una forma de declarar que todos los métodos de una clase son públicos y abstractos, con ello se especifica el comportamiento común de todas las clases que implementen la interfaz; su declaración es similar a la de una clase; en la cabecera utiliza la palabra reservada `interface` en vez de `class`.

- ▶ La interfaz especifica el comportamiento común que tiene un conjunto de clases, el cual se realiza en cada una de ellas y se conoce como implementación de interfaz; utiliza una sintaxis similar a la derivación o extensión de una clase, con la palabra reservada `implements` en lugar de `extends`.
- ▶ La clase que implementa la interfaz tiene que especificar el código (la implementación) de cada uno de sus métodos; de no hacerlo, la clase se convierte en abstracta y debe declararse `abstract`; esto es una forma de obligar a que cada método de la interfaz se implemente.

Listing 1: Ejemplo de interfaz

```
public interface GeometricFigure {  
    public double area();  
}  
  
public class Square implements GeometricFigure {  
    private double side;  
  
    ...  
  
    public double area() {  
        return (side * side);  
    }  
}  
  
public class Circle implements GeometricFigure {  
    private double radius;  
  
    ...  
  
    public double area() {  
        return (radius * radius) * Math.PI;  
    }  
}
```

- ▶ Java no permite que una clase derive de dos o más clases, es decir, no permite la herencia múltiple; sin embargo, una clase sí puede implementar más de una interfaz y tener el comportamiento común de varias de ellas.
- ▶ Las interfaces no son clases porque especifican un comportamiento mediante métodos para la clase que las implementa; por ello, una clase puede heredar de su clase base y a la vez implementar una interfaz.

Listing 2: Múltiples interfaces

```
public class ClassA implements Interface1, Interface2, ... Interfacen {  
    }
```

Clases abstractas

- ▶ Las clases abstractas declaran métodos y variables instancia, y normalmente tienen métodos abstractos; si una clase tiene un método abstracto debe declararse abstracta; una característica importante de estas clases es que de ellas no se pueden definir objetos, es decir, no se puede instanciar de una clase abstracta; el compilador devuelve un error siempre que se intenta crear un objeto de dichas clases.

Listing 3: Clases abstractas

```
public abstract class Person {  
    private String lastName;  
  
    public boolean identification(String a, String c){...}  
}
```

- ▶ Una clase interna es la que se declara dentro de otra clase; se puede decir que es anidada.
- ▶ Las clases internas declaran atributos y métodos de igual forma que las externas o de nivel superior, con la peculiaridad de que sus métodos pueden acceder a los atributos de su clase externa.

Listing 4: Clases internas

```
public class Student {  
    private int age;  
    String name;  
    Direction direction;  
  
    public Student(String name, int age, String street, String city,  
        String postalCode) {  
        this.name = name;  
        this.age = age;  
        direction = new Direction(street, city, postalCode);  
    }  
  
    class Direction {  
        String street, city, code;  
  
        Direction(String street, String city, String code) {  
            this.street = street;  
            this.city = city;  
            this.code = code;  
        }  
    }  
    ...  
}
```

Un objeto de la clase Direction puede hacer referencia a los miembros de Student; los objetos de la clase interna disponen de una referencia implícita al objetos que los contiene.

Examen Rápido 3

(Conceptos avanzados de POO en Java)