



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

Jueves, 21/09/2023

Hora de inicio:

07:10

Introducción a la programación y computación 1 [F]

José Eduardo Morales García

Polimorfismo

- El polimorfismo es una característica importante de la programación orientada a objetos que permite que un objeto pueda ser tratado como si fuera de un tipo diferente al suyo. En Java, el polimorfismo se logra a través del uso de clases y métodos abstractos, interfaces y clases que implementan esas interfaces.

Sobrecarga de métodos

- ▶ La sobrecarga de métodos en Java es una característica que permite definir varios métodos con el mismo nombre en la misma clase, siempre y cuando tengan diferentes parámetros. Esto significa que una clase puede tener dos o más métodos con el mismo nombre, pero cada uno con diferentes tipos de parámetros.

EJEMPLO

```
public class EjemploSobrecarga {

    public int sumar(int a, int b) {
        return a + b;
    }

    public double sumar(double a, double b) {
        return a + b;
    }

    public String sumar(String a, String b) {
        return a + b;
    }

    public static void main(String[] args) {
        EjemploSobrecarga ejemplo = new EjemploSobrecarga();
        System.out.println(ejemplo.sumar(2, 3)); // imprime 5
        System.out.println(ejemplo.sumar(2.5, 3.5)); // imprime 6.0
        System.out.println(ejemplo.sumar("Hola ", "Mundo")); // imprime "Hola Mundo"
    }
}
```

Virtualización

- El polimorfismo en Java se refiere a la capacidad de un objeto para ser tratado como un objeto de su superclase o interfaz. La virtualización es una técnica que permite simular o emular recursos o comportamientos físicos en un entorno virtual. En Java, el polimorfismo y la virtualización se utilizan juntos para crear objetos que se comportan de manera diferente según su tipo real.

EJEMPLO

```
public abstract class Animal {  
    public abstract void hacerSonido();  
}
```

```
public class Gato extends Animal {  
    @Override  
    public void hacerSonido() {  
        System.out.println("Miau!");  
    }  
}
```

```
public class Perro extends Animal {  
    @Override  
    public void hacerSonido() {  
        System.out.println("Guau!");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Animal animal1 = new Perro();  
        Animal animal2 = new Gato();  
  
        animal1.hacerSonido(); // imprime "Guau!"  
        animal2.hacerSonido(); // imprime "Miau!"  
    }  
}
```

Construcciones abstractas

- ▶ En programación, una construcción abstracta se refiere a un elemento de código que no tiene una implementación concreta y se utiliza para definir una estructura o comportamiento genérico que luego puede ser implementado por clases concretas. Las construcciones abstractas son muy utilizadas en lenguajes de programación orientados a objetos, como Java, para definir clases abstractas e interfaces.

Clase abstracta

- Una clase abstracta en Java es una clase que no se puede instanciar directamente, sino que se utiliza como una superclase para otras clases. Las clases abstractas se utilizan para definir una estructura común para un conjunto de subclases relacionadas.

EJEMPLO

```
public class Perro extends Animal {  
    public Perro(String nombre) {  
        super(nombre);  
    }  
  
    @Override  
    public void hacerSonido() {  
        System.out.println("Guau!");  
    }  
}
```

```
public abstract class Animal {  
    protected String nombre;  
  
    public Animal(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public abstract void hacerSonido();  
  
    public String getNombre() {  
        return nombre;  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Animal perro = new Perro("Fido");  
        Animal gato = new Gato("Pelusa");  
  
        System.out.println(perro.getNombre() + " hace:");  
        perro.hacerSonido();  
  
        System.out.println(gato.getNombre() + " hace:");  
        gato.hacerSonido();  
    }  
}
```

```
public class Gato extends Animal {  
    public Gato(String nombre) {  
        super(nombre);  
    }  
  
    @Override  
    public void hacerSonido() {  
        System.out.println("Miau!");  
    }  
}
```

Interface

- ▶ En Java, una interfaz es una colección de métodos abstractos y constantes que pueden ser implementados por cualquier clase que implemente la interfaz. Las interfaces se utilizan para definir un contrato que una clase debe cumplir para poder utilizarse en cierto contexto.

EJEMPLO

```
public interface Forma {  
    double getArea();  
  
    double getPerimetro();  
}
```



Dudas y Preguntas



Otras actividades

- ▶ Curso de Python intermedio o avanzado (con parte práctica)
- ▶ Coecys

Parte práctica