


Interfaces

Definición

- Una interfaz es un conjunto de métodos abstractos
 - Su objetivo es definir el formato de ciertos métodos, que posteriormente las clases se encargarán de implementar
 - También puede incluir constantes, que serán públicas y estáticas
- 

Creación de una interfaz

- Una interfaz se crea, al igual que las clases, en archivos .java
- Se define con la palabra reservada *interface*

Como los métodos solo pueden ser públicos y abstractos, se pueden omitir las palabras *abstract* y *public*

```
public interface Operaciones{  
    int k=10;  
    void girar(int grados);  
    int invertir();  
}
```

En el caso de las constantes, se omiten las palabras *public*, *final* y *static*

Implementación de una interfaz

- Una clase que implementa una interfaz está obligada a sobrescribir (implementar) todos los métodos de la misma.

Operaciones



Test

Al implementar los métodos en la clase, es obligatorio indicar el modificador *public*.

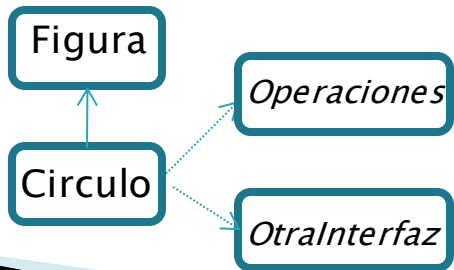
```
public class Test implements Operaciones{  
    public void girar(int grados){  
        :  
    }  
    public int invertir(){  
        :  
    }  
}
```

Flexibilidad de las interfaces

➤ Una clase puede implementar varias interfaces:

```
public class Test implements Operaciones, OtraInterfaz{  
    //implementación de todos los métodos de Operaciones  
    //y OtraInterfaz  
}
```

➤ Una clase puede heredar otra clase y, a su vez, implementar una o varias interfaces:



```
public class Circulo extends Figura implements  
Operaciones, OtraInterfaz {  
    :  
}
```

Herencia múltiple en interfaces

➤ Una interfaz puede heredar una o varias interfaces:

```
public interface Operaciones{  
    void girar(int grados);  
    int invertir();  
}  
public interface Inter1{  
    int miMetodo();  
}  
public interface InterFin extends Operaciones, Inter1{  
    void nuevoMetodo();  
}
```

```
public class Prueba implements InterFin{  
    public void girar(int grados){...}  
    public int invertir(){...}  
    public int miMetodo(){...}  
    public void nuevoMetodo(){...}  
}
```

La clase está obligada a implementar los métodos de la interfaz que implementa y los de las interfaces que esta hereda

Revisión conceptos



Si I1 e I2 son interfaces y C1 es una clase, indica cuales de las siguientes definiciones son correctas:

- a. class C2 implements I1, I2 {...}
- b. class C3 extends C1 implements I1, I2{...}
- c. interface I3 implements I1, I2{...}
- d. interface I4 extends I1 implements C1{...}
- e. interface I5 extends I1, I2

Respuesta

- a. Correcta, una clase puede implementar varias interfaces
- b. Correcta, una clase puede heredar otra e implementar varias interfaces
- c. Incorrecta, una interfaz no implementa, sino que hereda otras interfaces
- d. Incorrecta, una interfaz no puede implementar ni heredar clases
- e. Correcta, una interfaz puede heredar varias interfaces

Polimorfismo con interfaces

- Mediante una interfaz, se puede hacer referencia a un objeto que la implementa.
- Con esta referencia se podría llamar a las implementaciones de los métodos declarados en la de la interfaz, pudiendo llevar a cabo el polimorfismo a través de las interfaces.

```
interface Test{  
    int CONV=8.75;  
    void metodo1(int x);  
    int metodo2(String s);  
}  
class ClasePrueba implements Test{  
    public void metodo1(int x){...}  
    public int metodo2(String s){...}  
    public void metodoPropio(){...}  
}
```



```
Test ts=new ClasePrueba();  
ts.metodo1(10);  
ts.metodo2("hello");  
ts.metodoPropio();//Error de compilación
```

```
Test.CONV;  
ClasePrueba.CONV;
```



A las constantes se les puede llamar con la interfaz o con las clases que la implementan