

INTERFACES



DEFINICIÓN Y CREACIÓN

- Una interfaz es un conjunto de métodos abstractos públicos.
- Su objetivo es definir el formato de ciertos métodos, que posteriormente las clases se encargarán de implementar.
- Se crean en archivos .java, utilizando la palabra *interface*:

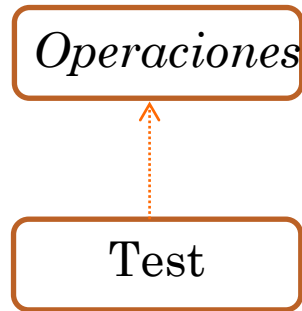
```
public interface Operaciones{  
    void girar(int grados);  
    int invertir();  
}
```

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



IMPLEMENTACIÓN DE UNA INTERFAZ

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



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

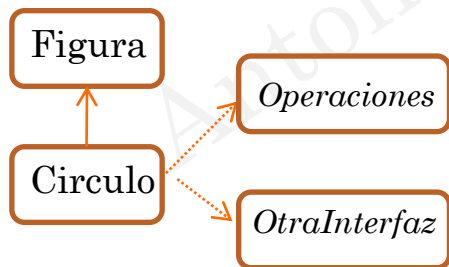
```
public class Test implements Operaciones{  
    ➤ public void girar(int grados){  
        :  
    }  
    ➤ public int invertir(){  
        :  
    }  
}
```

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 {  
    :  
}
```

EJEMPLOS INTERFACES JAVA ESTÁNDAR

➤ Colecciones:

- **List.** Interfaz que implementan colecciones tipo lista, como ArrayList
- **Set.** Interfaz que implementan los conjuntos
- **Map.** Interfaz que implementan las tablas
- **Collection.** Interfaz base para trabajar con colecciones. Heredada por List y Set

➤ SQL:

- **Connection, ResultSet, etc.**



INTERFACES Y POLIMORFISMO

➤ El polimorfismo puede aplicarse también con interfaces:

```
public interface Operaciones{  
    void girar(int grados);  
    int invertir();  
}  
public class Test implements Operaciones {  
    //implementación de girar e invertir  
}  
public class Circulo extends Figura implements  
Operaciones{  
    :  
}
```



```
usoPolimorfismo(new Test(..));  
usoPolimorfismo(new Circulo(..));  
:  
void usoPolimorfismo(Operaciones op){  
    op.girar(200);  
    op.invertir();  
}
```

Opera sobre un objeto de cualquier clase que implemente la interfaz



POLIMORFISMO INTERFACES JAVA ESTÁNDAR

➤ Clase Collections:

```
sort(List<T> list)
```

```
reverse(List<?> list)
```

➤ Clase Arrays:

```
List<T> asList(T...e)
```



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



OTROS ELEMENTOS EN UNA INTERFAZ

➤ Una interfaz puede incluir, además de métodos abstractos, constantes públicas y estáticas:

```
public interface Operaciones{  
    int data=100;  
    void girar(int grados);  
    int invertir();  
}
```

Por defecto es public static final. Pueden omitirse estas palabras reservadas

➤ Y más cosas desde Java 8, que estudiaremos en la próxima lección...

