

Genéricos

Definición

- Permiten que una clase puede operar con cualquier tipo de objeto Java (no pueden ser primitivos)
- En la definición de la clase se emplea una letra para hacer referencia de forma genérica al tipo:

```
class MyClass<T>{...}
```

- Al crear un objeto de la misma, se debe especificar el tipo concreto con el que se va a trabajar:

```
MyClass<String> mc=new MyClass<>();  
var n=new MyClass<String>();
```

Ejemplo

➤ Clase para encapsulación de cualquier tipo Java:

```
public class Bean<T> {  
    private T dato;  
    public Bean(T dato) {  
        this.dato=dato;  
    }  
    public void setDato(T dato) {  
        this.dato=dato;  
    }  
    public T getDato() {  
        return dato;  
    }  
}
```



```
Bean<String> b1=new Bean<>("hello");  
System.out.println(b1.getDato());  
Bean<Integer> b2=new Bean<>(30);  
System.out.println(b2.getDato());
```

Tipo genérico como parámetro

- A la hora de definir un parámetro de tipo genérico, se debe emplear el operador comodín (?)

```
public void print(Bean<?> bean){  
    System.out.println(bean.getDato());  
}
```

- El método podrá ser llamado con un objeto Bean de cualquier tipo:

```
Bean<String> b1=new Bean<>("hello");  
Bean<Integer> b2=new Bean<>(30);  
print(b1);  
print(b2);
```

Restricciones de tipo

- Se puede definir una clase que solo admita objetos de un determinado subtipo o supertipo:

cualquier subclase de Number


```
class MyClass<T extends Number>{  
    ...  
}
```

cualquier superclase de JButton

```
class MyClass<T super JButton>{  
    ...  
}
```

- También al definir parámetros de tipo genérico:

```
public void print(Bean<? extends Number> bean){  
    System.out.println(bean.getDato());  
}
```



```
print(new Bean<String>("hello")); //error compilación  
print(new Bean<Integer>(30)); //ok
```

Métodos genéricos

- Una clase no genérica puede incluir métodos que reciban como parámetro un tipo genérico.
- Se incluirá la expresión `<T>` en la definición del método:

```
public class MetodosGenericos {  
    public <T> String tipo(T dato) {  
        return dato.getClass().getName();  
    }  
}
```



```
MetodosGenericos mg=new MetodosGenericos();  
System.out.println(mg.tipo("hello")); //String  
System.out.println(mg.tipo(50)); //Integer
```