



1. Métodos de Ordenación
2. Métodos de Estáticos

Métodos de Ordenación y Estáticos

Métodos de Ordenación

MAP

```
CREATE or replace TYPE figura_t AS OBJECT(  
  coordenadaX NUMBER,  
  coordenadaY NUMBER,  
  altura NUMBER,  
  anchura NUMBER,  
  MAP MEMBER FUNCTION calculaArea RETURN NUMBER,  
  MEMBER PROCEDURE mueveFigura(x1 NUMBER, y1 NUMBER));
```

```
CREATE OR REPLACE TYPE BODY figura_t AS  
  MAP MEMBER FUNCTION calculaArea RETURN NUMBER IS  
    BEGIN  
      return altura*anchura;  
    END;  
  MEMBER PROCEDURE mueveFigura(x1 NUMBER, y1 NUMBER) IS  
    BEGIN  
      coordenadaX:=x1;  
      coordenadaY:=y1;  
    END;  
END;
```

MAP

```
set serveroutput on;
DECLARE
    figura1 figura_t;
    figura2 figura_t;
BEGIN
    figura1:=figura_t(0,0,50,100);
    figura2:=figura_t(0,0,100,100);
    IF (figura1>figura2) THEN dbms_output.put_line('La figura 1 es mayor que la figura 2');
    ELSE dbms_output.put_line('La figura 2 es mayor que la figura 1');
    END IF;
END;
```

ORDER

```
CREATE or replace TYPE figura_t AS OBJECT(  
  coordenadaX NUMBER,  
  coordenadaY NUMBER,  
  altura NUMBER,  
  anchura NUMBER,  
  MEMBER FUNCTION calculaArea RETURN NUMBER,  
  ORDER MEMBER FUNCTION compararArea (f figura_t) RETURN NUMBER,  
  MEMBER PROCEDURE mueveFigura(x1 NUMBER, y1 NUMBER));
```

```
CREATE OR REPLACE TYPE BODY figura_t AS  
  MEMBER FUNCTION calculaArea RETURN NUMBER IS  
    BEGIN  
      return altura*anchura;  
    END;  
  ORDER MEMBER FUNCTION compararArea (f figura_t) RETURN NUMBER IS  
    BEGIN  
      return calculaArea-f.calculaArea;  
    END;  
  MEMBER PROCEDURE mueveFigura(x1 NUMBER, y1 NUMBER) IS  
    BEGIN  
      coordenadaX:=x1;  
      coordenadaY:=y1;  
    END;  
END;
```

ORDER

```
set serveroutput on;
DECLARE
    figural figura_t;
    figura2 figura_t;
BEGIN
    figural:=figura_t(0,0,50,100);
    figura2:=figura_t(0,0,100,100);
    dbms_output.put_line('La figura 1 es mayor que la figura 2 en '||figural.compararArea(figura2));
END;
```

Resumen

- Un método MAP, mapea el objeto en un valor escalar y puede ordenar múltiples objetos por su posición en el eje escalar.
- Un método ORDER directamente compara valores de dos objetos particulares.
- Cuando sea necesario ordenar un gran número de objetos, mejor usar el método MAP

Resumen

- En un objeto, se puede declarar un método MAP o un método ORDER, pero no ambos.
- Si un tipo de objeto no tiene definido ningún método de comparación solamente permite comparaciones de igualdad.

Métodos de Estáticos

```
CREATE OR REPLACE TYPE solid_typ AS OBJECT(  
  len INTEGER,  
  wth INTEGER,  
  hgt INTEGER,  
  STATIC PROCEDURE new_solid_typ(l INTEGER, w INTEGER, h INTEGER)  
);  
  
CREATE TABLE SOLID_TABLE OF SOLID_TYP;  
CREATE OR REPLACE TYPE BODY solid_typ AS  
  STATIC PROCEDURE new_solid_typ(l INTEGER, w INTEGER, h INTEGER) IS  
    sql_insert VARCHAR2(100);  
  BEGIN  
    sql_insert:='INSERT INTO SOLID_TABLE VALUES (solid_typ('||l||','||w||','||h||'))';  
    EXECUTE IMMEDIATE sql_insert;  
  END;  
END;  
  
BEGIN  
  solid_typ.new_solid_typ(20, 20, 20);  
END;
```

“Donde no puedas amar, no te demores”

FRIDA KAHLO

