Métodos

De forma habitual, en programación orientada a objetos, cuando se define una clase también se establecen los métodos que modelan su comportamiento. En el contexto que nos ocupa, los métodos son procedimientos y funciones que se especifican después de los atributos del objeto. Pueden ser de varios tipos; MEMBER, CONSTRUCTOR Y STATIC.

```
CREATE OR REPLACE TYPE RECTANGULO AS OBJECT
(
  BASE   NUMBER,
  ALTURA NUMBER,
  AREA   NUMBER,
  STATIC PROCEDURE PROC1 (ANCHO INTEGER, ALTO INTEGER),
  MEMBER PROCEDURE PROC2 (ANCHO INTEGER, ALTO INTEGER),
  CONSTRUCTOR FUNCTION RECTANGULO (BASE NUMBER, ALTURA NUMBER)
                       RETURN SELF AS RESULT
);
```

Una vez que se ha creado el tipo con la especificación de atributos y métodos, se puede establecer el cuerpo del tipo mediante `CREATE OR REPLACE TYPE BODY`

```
CREATE OR REPLACE TYPE BODY nombre_tipo AS
    [STATIC | MEMBER] PROCEDURE nombreProcedure [(parametro1, parametro2,...)]
IS
    Declaraciones;
BEGIN
    Instrucciones;
END;


CREATE OR REPLACE TYPE BODY nombre_tipo AS
    [STATIC | MEMBER | CONSTRUCTOR] FUNCTION nombrefuncion [(parametro1,
parametro2,...)] RETURN tipo_valor_retorno
IS
    Declaraciones;
BEGIN
    Instrucciones;
END;
```

MEMBER

Son los métodos más habituales, propios de cada objeto. Pueden ser Procedimientos o Funciones.

CONSTRUCTOR

Su cometido es inicializar los objetos. Sus argumentos son los atributos del objeto y devuelve el objeto inicializado. Oracle define un constructor por defecto para cada objeto, si bien, este constructor se puede sobreescribir o definir constructores adicionales que incorporen otras funcionalidades (restricciones, valores por defecto, etc.). En la cláusula RETURN deben establecer la

expresión **RETURN SELF AS RESULT**.

STATIC

Al igual que los métodos de clase de la POO, estos métodos son independientes de las instancias de un objeto. Engloban las operaciones que no pertenecen a un objeto concreto sino al tipo.

```
CREATE OR REPLACE TYPE BODY RECTANGULO AS

    MEMBER PROCEDURE PROC2 (ANCHO INTEGER, ALTO INTEGER) IS
      BEGIN
        SELF.ALTURA := ALTO; --SE PUEDE ACCEDER A LOS ATRIBUTOS DEL TIPO
        SELF.BASE := ANCHO;
        AREA := ALTURA*BASE;
        INSERT INTO TABLAREC VALUES(AREA);
        DBMS_OUTPUT.PUT_LINE('FILA INSERTADA');
        COMMIT;
      END;
    END;

    CONSTRUCTOR FUNCTION RECTANGULO (BASE NUMBER, ALTURA NUMBER) RETURN SELF AS
RESULT IS
      BEGIN
        SELF.BASE := BASE;
        SELF.ALTURA := ALTURA;
        SELF.AREA := BASE * ALTURA;
        RETURN;
    END;

    STATIC PROCEDURE PROC1 (ANCHO INTEGER, ALTO INTEGER) IS
      BEGIN
        INSERT INTO TABLAREC VALUES(ANCHO*ALTO);
        --ALTURA := ALTO; --ERROR NO SE PUEDE ACCEDER A LOS ATRIBUTOS DEL TIPO
        DBMS_OUTPUT.PUT_LINE('FILA INSERTADA');
        COMMIT;
    END;
```

Sobrecarga

Como ya se ha comentado, mediante el mecanismo de herencia se da la posibilidad de extender objetos a partir de objetos ya existentes. Así pues, los objetos hijos pueden definir sus propios métodos o modificar los de sus objetos padres. La cláusula *OVERRIDING* permite redefinir métodos. Hay que situarla antes del tipo de método (STATIC, MEMBER, CONSTRUCTOR). De forma similar a otros lenguajes que soportan POO, es posible utilizar el mismo nombre para métodos distintos; variando sus parámetros formales en tipo, orden o cantidad.