

## TIPO ENUMERADO

- ▶ Tipo de dato especial
- ▶ Indica que una variable tendrá como valor uno de entre un conjunto cerrado.
- ▶ Por ejemplo Direccion (Norte, Sur, Este, Oeste).

```
public enum Direccion {  
    NORTE, SUR, ESTE, OESTE  
}  
  
public enum Dia{  
    LUNES, MARTES,  
    MIERCOLES, JUEVES,  
    VIERNES, SABADO,  
    DOMINGO  
}
```

Ejemplo:

```
public enum DiaSemana {  
    LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO  
}
```

Para crear una enumeración podéis usar el asistente de eclipse, parecido al de crear las clases.

Una vez que la he creado, puedo crear diferentes métodos que reciban un tipo enum

```
//  
public static void main(String[] args) {  
  
    saludaSegunDia(DiaSemana.LUNES);  
    saludaSegunDia(DiaSemana.MIERCOLES);  
    saludaSegunDia(DiaSemana.VIERNES);  
    saludaSegunDia(DiaSemana.SABADO);  
  
}  
  
public static void saludaSegunDia(DiaSemana d) {  
    switch (d) {  
        case LUNES:  
            System.out.println("Los lunes son duros :(");  
            break;  
  
        case VIERNES:  
            System.out.println("Los viernes son mejores :)");  
            break;  
  
        case SABADO: case DOMINGO:  
            System.out.println("Los fines son los mejores dias :)))");  
            break;  
    }  
}
```

Para java una enumeración es como una clase, puede incluir métodos, atributos, es como un conjunto de valores, pero podríamos verlo como un conjunto de instancias u objetos que tengan sus métodos y atributos.

## TIPO ENUMERADO

- ▶ Son más potentes que en otros lenguajes.
- ▶ Para Java son tipos de clases.
- ▶ Pueden incluir métodos y otros atributos.
- ▶ El compilador añade métodos especiales (*values*).
- ▶ Podemos pensar en que tenemos un conjunto cerrado de instancias de una clase.

Mira esta enumeración

```
1 public enum Planeta {  
2     MERCURIO (3.303e+23, 2.4397e6),  
3     VENUS (4.869e+24, 6.0518e6),  
4     TIERRA (5.976e+24, 6.37814e6),  
5     MARTE (6.421e+23, 3.3972e6),  
6     JUPITER (1.9e+27, 7.1492e7),  
7     SATURNO (5.688e+26, 6.0268e7),  
8     URANO (8.686e+25, 2.5559e7),  
9     NEPTUNO (1.024e+26, 2.4746e7);  
10  
11     private final double masa; //en kg  
12     private final double radio; //en metros  
13 }
```

Cada elemento tiene dos atributos, podemos ponerle final, porque eso sí, son inmutables, no van a variar de valor. Podemos tener métodos getter and setter para obtener la masa y radio de cada planeta. U otros métodos:

```
1 //Constante universal G (m3 kg-1 s-2)  
2 public static final double G = 6.67300E-11;  
3  
4 public double gravedadSuperficial() {  
5     return G * masa / (radio * radio);  
6 }  
7  
8 public double pesoEnLaSuperficie(double otraMasa) {  
9     return otraMasa * gravedadSuperficial();  
10 }
```

Puedo trabajar con los elementos como si fueran instancias de una clase, y con value, recorro todos los valores de la enumeración.

```
1  
2 public static void main(String[] args) {  
3  
4     double pesoEnLaTierra = 75.0;  
5     double masaEnLaTierra = pesoEnLaTierra/ Planeta.TIERRA.gravedadSuperficial();  
6     for (Planeta p : Planeta.values())  
7         System.out.printf("Tu peso en %s es %f\n", p, p.pesoEnLaSuperficie(masaEnLaTierra));  
8  
9 }
```

Eso sí, no podemos desde el main crear nuevas instancias, si se descubriera un planeta más, debemos modificar la enumeración, no podemos añadir otro elemento de otra forma.