Tipos enumerados

Tipos enumerados

- **Tipo enumerado**. Un tipo especial que permite representar variables que tienen un conjunto finito de valores.
- Ejemplos:
- Caras de una moneda: águila / sol, cara / cruz.
- Valores de monedas: 50c, $1, $2, $5, $10, $20.
- Días de la semana: lunes, martes, miércoles, jueves, viernes, sábado, domingo.
- Planetas: Mercurio, Venus, Tierra, Marte, Júpiter, Saturno, Urano, Neptuno.

Tipos enumerados

- Sintaxis:

```
public enum Nombre {
    VALOR, VALOR, ..., VALOR
}
```

- Un tipo enumerado es un tipo especial de clase.
- Puede ir en su propio archivo .java.
- Se acostumbra que el nombre del tipo enumerado comience con mayúscula.
- Se acostumbra que los valores se escriban en mayúscula.

Ejemplo

```
public enum Season { SPRING, SUMMER, FALL, WINTER }
Season season;
season = Season.SUMMER;
if (season == Season.WINTER) {
    …
}
```

Métodos enum

int compareTo(E)	Todos los tipos enum son comparables por orden de declaración
boolean equals(o)	No se requiere, se puede usar ==
String name()	Equivalente a toString
int ordinal()	Regresa un número por orden de declaración (primero es 0, segundo es 1, …)
static E valueOf(s)	Convierte un string en un valor enum
static E[] values()	Regresa un arreglo con los valores de la enumeración

Ejemplo

```
public class EnumDemo {
   // Define two enum types -- remember that the definitions
   // go OUTSIDE the main() routine!
   enum Day { SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY,
SATURDAY }
   enum Month { JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC }

   public static void main(String[] args)
   {
       Day tgif;     // Declare a variable of type Day.
       Month libra;  // Declare a variable of type Month.
       tgif = Day.FRIDAY;    // Assign a value of type Day to tgif.
       libra = Month.OCT;    // Assign a value of type Month to libra.
```

Ejemplo

```
        System.out.print(“My sign is libra, since I was born in ”);
        System.out.println(libra);   // Output value will be:  OCT
        System.out.print(“That's the ”);
        System.out.print( libra.ordinal() );
        System.out.println(“-th month of the year.”);
        System.out.println(“   (Counting from 0, of course!)”);
        System.out.print(“Isn't it nice to get to ”);
        System.out.println(tgif);   // Output value will be:  FRIDAY
        System.out.println( tgif + “ is the ” + tgif.ordinal() + “-th day of the week.”);
    }
}
```

Tipos enumerados más complejos

- Un tipo enumerado puede tener campos, métodos y constructores.

Ejemplo

```
public enum Coin {
    PENNY(1), NICKEL(5), DIME(10), QUARTER(25);
    private final int cents;
    private Coin(int cents) {          // Constructor
        this.cents = cents;
    }
    public int getCents()  { return cents; }
    public int perDollar() { return 100 / cents; }
    public String toString() {       // "NICKEL (5c)"
        return super.toString() + " (" + cents + "c)";
      }
}
```

Enumerados con instrucciones switch

```
enum Season { SPRING, SUMMER, FALL, WINTER }
Season season = Season.SUMMER;
```

Enums con instrucciones switch

```
switch ( currentSeason ) {
    case WINTER:    // ( NOT Season.WINTER ! )
        System.out.println("December, January, February");
        break;
     case SPRING:
        System.out.println("March, April, May");
        break;
    case SUMMER:
        System.out.println("June, July, August");
        break;
    case FALL:
        System.out.println("September, October, November");
        break;
}
```