

# Clase 11: Implementación de clases

Una clase es un pedazo de código, que define a los objetos que tienen su mismo nombre. Por ejemplo, la clase Fraccion es el código Java donde se define todo acerca de los objetos Fracción.

Lo primero que define la clase, es el nombre de los objetos. Luego define su representación interna (todas las variables que son necesarias para describir al objeto) y finalmente define que es lo que hacen los objetos (los métodos).

## 1 Representación Interna

Casi todas las clases tienen una representación interna, es decir, un conjunto de variables que la describen. Estas variables pueden ser primitivas, como int, boolean y double, o referencias a otros objetos.

### 1.1 Ejemplos

#### 1.1.1 R.I. de clase Fecha

Para describir una fecha, se pueden usar

- 3 ints (mes, año, día)
- 1 int (ej: 20041231)
- 1 String “2004-12-31”
- 1 String “20041231”

#### 1.1.2 R.I. de clase Fracción

En el caso de una fracción, lo más sencillo es representar el denominador y numerador con un entero

#### 1.1.3 R.I. de clase Complejo

Un double para la parte real y otro para la parte imaginaria

#### 1.1.4 R.I. de clase Mesa

Un string para el color, un int para la cantidad de patas, un double para su altura, etc

## 2 Una clase sencilla: Fecha<sup>1</sup>

La forma más sencilla de una clase abarca sólo su representación interna:

```
class Fecha {  
    int dia;  
    int mes;  
    int año;  
}
```

El siguiente código usa la clase Fecha

```
1 Fecha f=new Fecha();  
2 f.dia=15;  
3 f.mes=8;  
4 f.año=2004;  
5 con.println(“La fecha es “+f.dia+”/”+f.mes+”/”+f.año);
```

es decir, uno puede acceder a la R.I. de la clase poniendo el nombre de la variable seguido de un punto y el nombre de la variable de R.I. Solo con esto, ya se pueden crear objetos que sirven para guardar datos agregados.

## 3 La implementación de Fecha

Luego de definir la R.I., se escribe en el código las cosas que pueden hacer los objetos definidos por la clase. Esto se lleva a cabo mediante métodos.

### 3.1 Constructor

El primer método que uno debería definir es el constructor. Este método se llama igual que la clase, no tiene tipo y se llama automáticamente cuando se hace new.

```
1 class Fecha {  
2     int dia;  
3     int mes;  
4     int año;  
5     public Fecha (int d, int m, int a) {  
6         this.dia=d;  
7         this.mes=m;  
8         this.año=a;  
9     }  
10 }
```

---

<sup>1</sup>Para este ejemplo, vamos a suponer que todos los meses tienen 30 días.

Un programa que usa lo anterior es

```
Fecha f;  
f=new Fecha(15,10,1985);
```

### 3.2 this

En la línea 6, aparece mencionado this. Si te fijas, tiene la misma sintaxis de referencia.variable que se uso anteriormente, solo que en vez de poner como referencia el nombre de una variable, se pone **this**.

this siempre hace referencia al objeto que invoca al método. Por ejemplo, si se hace

```
Fecha f,g;  
f=new Fecha(1,12,2004);  
g=new Fecha(7,6,2003);
```

la primera vez que se ejecuta la línea 6, el valor de this es la referencia al objeto Fecha 1/12/2004, y la segunda vez que se ejecuta, es la referencia al objeto Fecha 7/6/2003.

### 3.3 Métodos que devuelven información

Los objetos Fecha tienen un método que devuelve una representación textual de su valor. A este método le llamaremos toString:

```
class Fecha {  
    int dia;  
    int mes;  
    int año;  
    public Fecha (int d, int m, int a) {  
        this.dia=d;  
        this.mes=m;  
        this.año=a;  
    }  
    public String toString() {  
        return this.dia+"/"+this.mes+"/"+this.año;  
    }  
}
```

Un ejemplo de uso:

```
Fecha f=new Fecha(14,10,1977);  
con.println("El valor de f es "+f.toString());
```

### 3.4 La clase completa

La definición de toda la clase se lleva a cabo agregando métodos, como se vió en la clase 6. La única diferencia, es cada método, además de poder acceder a las variables que se le entregan como parámetro, puede acceder a la representación interna:

```
class Fecha {  
    int dia;  
    int mes;  
    int año;  
    public Fecha (int d, int m, int a) {  
        this.dia=d;  
        this.mes=m;  
        this.año=a;  
    }  
    public String toString() {  
        return this.dia+"/"+this.mes+"/"+this.año;  
    }  
    public void sumaDias(int dias) {  
        this.dia+=dias;  
        if (this.dia>30) {  
            this.dia-=30;  
            this.mes++;  
        }  
        if (this.mes>12) {  
            this.mes=1;  
            this.ano++;  
        }  
    }  
    public Fecha mañana() {  
        Fecha f=new Fecha(this.dia, this.mes, this.año);  
        f.sumaDias(1);  
        return f;  
    }  
}
```