

Sobrecarga en constructores

Una clase puede definir varios constructores (un objeto puede instanciarse de varias formas posibles). Para cada nueva instancia el constructor que se ejecute dependerá de la coincidencia en la cantidad y tipos de parámetros. Si no se encuentra ninguna coincidencia se producirá un error en tiempo de compilación.

Las características en tener en consideración en los constructores son:

1. Los constructores tienen el mismo nombre que la clase
2. Se permite la sobrecarga del constructor
3. Los constructores no son métodos.
4. Los constructores no tienen un tipo de retorno, ni siquiera "void".

En el siguiente ejemplo, se escriben 4 tipos de constructores diferentes para la clase persona.

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
    private String numeroCel;  
  
    public Persona() {  
        nombre = "Juan";  
        edad = 25;  
        numeroCel = "00-000-0000";  
    }  
  
    public Persona(String nombre) {  
        this.nombre = nombre;  
        edad = 25;  
        numeroCel = "00-000-0000";  
    }  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
        numeroCel = "00-000-0000";  
    }  
  
    public Persona(String nombre, int edad, String numeroCel) {  
        this.nombre = nombre;  
        this.edad = edad;
```

```
    this.numeroCel = numeroCel;
}
}
```

Desde un constructor puede invocarse explícitamente a otro constructor utilizando la palabra reservada **this**. **this** es una referencia al propio objeto. Cuando **this** es seguido por paréntesis se entiende que se está invocando al constructor. Los datos que se ingresen dentro de sus paréntesis buscarán la coincidencia más específica. En el caso que no encuentre un constructor con los valores asignados a **this**, arrojará un error en tiempo de compilación.

```
public class Persona {

    private String nombre;
    private int edad;
    private String numeroCel;

    public Persona() { //Constructor 1
        this("Juan", 25, "00-000-0000"); //Llamando a constructor 4
    }

    public Persona(String nombre) { //Constructor 2
        this(nombre, 25); //Llamando a constructor 3
    }

    public Persona(String nombre, int edad) { //Constructor 3
        this.nombre = nombre;
        this.edad = edad;
        numeroCel = "00-000-0000";
    }

    public Persona(String nombre, int edad, String numeroCel) { //Constructor 4
        this.nombre = nombre;
        this.edad = edad;
        this.numeroCel = numeroCel;
    }
}
```

Sobrecarga en métodos

Una clase puede definir varios métodos con el mismo nombre, bajo ciertas condiciones.

1. La cantidad de parámetros es diferente.
2. Si la cantidad de parámetros es igual, algún tipo de dato deberá ser diferente.
3. Si la cantidad de parámetros y tipos son iguales, algún parámetro tendrá un orden diferente.

Obs. 1: Un método sobrecargado puede tener diferente tipo de retorno.

Obs. 2: El tipo de dato de retorno, no se toma como criterio de diferencia.

En el siguiente ejemplo, se tiene la clase Numero con su método max sobrecargado.

```
public class Numero {  
    int max(int a, int b) {  
        return a > b ? a : b;  
    }  
  
    int max(int a, int b, int c) { //Diferentes números de parámetros  
        if (a > b && a > c)  
            return a;  
        if (b > a && b > c)  
            return b;  
        return c;  
    }  
  
    int max(int... v) { //Número variado de argumentos  
        int maximo = v[0];  
        for (int i = 1; i < v.length; i++)  
            if (v[i] > maximo)  
                maximo = v[i];  
        return maximo;  
    }  
}
```

```
}
```

```
int max(int a, float b) { //Diferente tipo de parámetro
```

```
    return (int) ((a > b) ? a : b);
```

```
}
```

```
int max(float a, int b) { //Diferente orden de parámetro
```

```
    return (int) ((a > b) ? a : b);
```

```
}
```

```
}
```