
Tema 15:

Aserciones



Aserciones

- Definición
- Una *aserción* es una condición lógica insertada en el código Java, de ideas o condiciones que se asumen son ciertas. El sistema se encarga de comprobarlas y avisar mediante una excepción en caso de que no se cumplan.
- Generalmente es para verificar valores de las variables en cierto punto del programa.



Uso

- Se aconseja su uso durante el desarrollo y las pruebas, se pueden “eliminar” en la implantación del sistema.
- “Reglas No”
 - ✓ Usarse para detectar errores en los datos de entrada al programa.
 - ✓ Usarse en la entrada de métodos públicos.

... Uso

- “Reglas Si”

- ✓ Usarse en la entrada de métodos privados.
- ✓ Usarse en la salida de métodos públicos o privados.
- ✓ Usarse para verificar cómo se suponen están las variables y estructuras de datos internas.
- ✓ Usarse en la sentencia **default** de la estructura **switch** cuando todos los casos correctos están explícitos (cuando **else** no debería ocurrir).
- ✓ Usarse en el último **else** de construcciones **if ... else if ... switch** cuando todos los casos correctos están explícitos (cuando el último **else** no debería ejecutarse jamás).
- ✓ Usarse en ciclos largos.

Sintaxis

- Sintaxis 1:

assert expresion;

- Ejemplo

double m= 100.0;

double n= m/2;

assert m == n*2; //cond. asumida *true*.

- Sintaxis 2:

assert expresion1:expresion2;

- Ejemplo:

double m= 100.0;

double n= m/2;

assert m == n*2 : "Java no sabe dividir " +
m + " entre 2: " + n;

Ejemplo

- `if (! (estado_ideal) ) throw new Error("Fallo en esta zona del programa");`
- Equivale a :
`assert estado_ideal;`
- `for (int i = 0; i < 10; i++) {
 if (v[i] > 1000) return i; }
assert false;`

En el caso de que el ciclo termine sin haber ejecutado el `return`, se evaluará la expresión del `assert` (en este caso *false*) por lo que el sistema lanza una excepción (*AssertionError*).

Ejemplos de uso

- En métodos públicos y no públicos.
- **public** Book reservaCopia(String titulo,Miembro miembro) {
 if (! esValidoTitulo(titulo)) throw new
 IllegalArgumentException("Mal titulo:" + titulo);

```
Book libro =getDisponibleCopia(titulo);  
reservar(libro, miembro);  
assert libroEnStrock(libro);
```

```
    return libro;
```

```
}
```

- **private** Book reservaCopia(String titulo,Miembro miembro) {
 assert esValidoTitulo(titulo);

```
Book libro =getDisponibleCopia(titulo);  
reservar(libro, miembro);  
assert libroEnStrock(libro);
```

```
    return libro;
```

```
}
```

Ejemplos de uso

- En if's

```
if (i % 3 == 0) { ... }
```

```
else if (i % 3 == 1) { ... }
```

```
    else { assert i % 3 == 2 : i; ... }
```

- Con while

```
    while (saldo > 0)
```

```
    { retirar(1000); }
```

```
    assert (saldo <= 0);
```


Ejemplos de uso

- En switch

```
switch (op) {  
    case AND: sumar();  
              break;  
    case OR:  restar();  
              break;  
    case NOT: negar();  
              break;  
    default: assert false;  
}  
...
```

Activación y desactivación

- Las aserciones por default están deshabilitadas.
- Indicarlo al compilador :
`javac -source 1.4 MiPrograma.java`
- Para activar:
`java -enableassertions MiPrograma`
`java -ea MiPrograma //otra opción`
`java -ea:class MiClase`
- Para desactivar:
`java -disableassertions MiPrograma`
`java -da MiPrograma //otra opción`

Ejemplo1

```
class AssertDemo {
    static int val = 3;

    static int getnum() {
        return val--;
    }

    public static void main(String args[])
    {
        int n;
        for(int i=0; i < 10; i++) {
            n = getnum();
            assert n > 0; // fallará cuando n es 0
            System.out.println("n es " + n);
        }
    }
}
```

Ejemplo2

```
class TestAserciones {  
    private static int busca (int n, final int[] v) {  
        int resultado= busca (n, v, 0, v.length);  
        assert resultado >= -1 && resultado < v.length;  
        return resultado; }  
  
    private static int busca (int n, final int[] v, int a, int z) {  
        assert a >= -1 && a <= v.length;  
        assert z >= -1 && z <= v.length;  
        if (a < z) {  
            int m= (a+z)/2;  
            if (n == v[m])  
                return m;  
            if (n < v[m])  
                return busca(n, v, a, m-1);  
            if (n > v[m])  
                return busca(n, v, m+1, z);  
            assert false;  
        } return -1;  
    }  
}
```

Ejemplo2

```
public static void main (String[] args) {  
    int[] v= {1, 2, 3, 5, 7, 11, 13, 17, 19};  
    System.out.print("Primos: ");  
    for (int i= 0; i < v.length; i++)  
        System.out.print(v[i] + " ");  
    System.out.println();  
    System.out.println("17: " + busca(17, v));  
    System.out.println("15: " + busca(15, v));  
    System.out.println("-1: " + busca(-1, v));  
}  
} //Fin de TestAserciones.
```