

# Asertos

## Programación I

Grado en Ingeniería Informática  
JCRdP y JDGD

# Introducción

- ▶ Según el Diccionario de la lengua española: un aserto es "la afirmación de la certeza de algo"
- ▶ En el caso de la programación podemos afirmar la certeza de ciertas expresiones si el programa funciona correctamente
- ▶ Muchos lenguajes de programación suministran instrucciones con formato:
- ▶ `assert Exp`

donde `Exp` es una expresión lógica que se evalúa en ejecución y en caso de que resulte falsa se detiene la ejecución

# Ejemplos de asertos

- Si tenemos un método para calcular la raíz cuadrada de un número entero podemos asegurar:

```
r = raíz(num);  
assert r > 0;  
assert r*r == num;
```

# Ejemplos de asertos

- Si tenemos un método de ordenación de un array, podemos asegurar que cada elemento es mayor o igual que el anterior:

```
sort(v);  
for(int i=1; i< v.length;i++)  
    assert v[i-1] <= v[i];
```

# Asertos en Java

- ▶ Formato: **assert** Exp1[:Exp2]  
Exp1 es una expresión lógica, se evalúa al ejecutar el assert, si da falso se lanza la excepción `AssertionError`, si existe Exp2 se le pasa al constructor como información adicional
- ▶ La excepción siempre lleva la información de desde dónde se ha lanzado
- ▶ La evaluación de los asertos puede ser muy costosa. Para activarla usar la opción de línea de comando del intérprete "-ea"

# No usar asertos en Java ...

- ▶ Para el chequeo de los argumentos de un método público

Los parámetros de métodos públicos se deben comprobar sin asertos y lanzando excepciones como `IllegalArgumentException`, `IndexOutOfBoundsException` o `NullPointerException`

- ▶ Para hacer alguna tarea requerida por la aplicación

```
assert students.remove(maria);
```

si se desactivan los asertos el programa fallará

# Precondiciones, postcondiciones e invariantes de clase

- ▶ **Precondición:** condiciones que deben cumplir los parámetros pasados al método
- ▶ **Postcondición:** condiciones que debe cumplir el resultado de la ejecución de un método. Con frecuencia se debe disponer de los datos iniciales para evaluar si se cumple
- ▶ **Invariante de clase:** condiciones del estado del objeto que deben cumplirse siempre, antes y después de la ejecución de un método

# Precondiciones y asertos

- ▶ Las precondiciones en los métodos públicos no deben comprobarse con asertos.
- ▶ Sí es adecuado usar asertos para comprobar las precondiciones de los métodos privados, ya que en caso de que fallen es un problema interno de la clase



# Postcondiciones y asertos

- Es adecuado usar asertos para comprobar las postcondiciones de los métodos públicos y privados, ya que si fallan siempre será un problema interno del método

# Invariantes de clase y asertos

- Los invariantes de clase se establecen creando un método privado que devuelve un lógico y que se debe llamar al finalizar cada método y comprobarlo con un assert