

# Jazyk Java 1

## Seminář 11

Daniel Bazala



Katedra informatiky  
Univerzita Palackého v Olomouci

# JavaDoc

- nástroj, který umí prohledat zdrojové kódy v Javě a na základě speciálních komentářů vygenerovat dokumentaci ve formátu HTML
- integrovaná ve většině IDE, možnost náhledů
- komentáře mají speciální syntaxi
- můžeme využít HTML a CSS

## ■ tagy:

```
@param jake parametry metoda akceptuje  
@return co metoda vraci  
@throws jaké vyjimky může vyvolat  
@see méně používané, význam "see_also"  
@author jméno autora  
@version verze třídy, rozhraní nebo enumu
```

## ■ př.

```
/**
 * Returns an Image object that can be painted on the screen.
 * The url argument must specify an absolute url.
 *
 * @param url an absolute URL
 * @param name the location of the image
 * @return the image at the specified URL
 * @see Image
 */
public Image getImage(URL url, String name) {
    try {
        return getImage(new URL(url, name));
    } catch (MalformedURLException e) {
        return null;
    }
}
```

- javadoc program pro vygenerování HTML dokumentace
- součástí JDK
- `javadoc company.example -d output`

# JUnit

- JUnit testovací framework pro Javu
- <https://junit.org>
- unit testy jako základní technika zachycení chyb
- přísně vzato testy nemohou dokázat korektnost kódu, ale pouze jeho nesprávné fungování



- ověření správných výsledků pomocí assertů

```
assertTrue/assertFalse
assertNull/assertNotNull
assertSame/assertNotSame (==)
assertEquals
assertArrayEquals
assertIterableEquals (deep)
fail (error)
assertThrows (exception)
assertThat
assertAll
```

- mějte třídu, kterou chceme otestovat:

```
public class Calculator {
    private double result = 0.0;

    public void sum(double sum) {
        result += sum;
    }

    public void divide(double divide) {
        result /= divide;
    }

    public double getResult() {
        return result;
    }
}
```

## ■ testy:

```
public class TestCalculator {
    @Test
    public void sum() {
        Calculator calc = new Calculator();
        calc.sum(4.0);
        calc.sum(7.0);
        assertEquals(11.0, calc.getResult());
    }

    @Test
    public void divByZero() {
        Calculator calc = new Calculator();
        calc.sum(4.0);
        calc.divide(0.0);
        assertTrue(Double.POSITIVE_INFINITY == calc.getResult());
    }
}
```

# Úkol

# Úkol seminář 11

- <http://marcus.webly3d.net/ukol11>