

Seminar 5. Introducere in Javadoc

Introducere

Timp de lucru estimat: 5 minute

Acest document reprezinta Seminarul 5 pentru cursul de introducere in programare Java. El contine aspecte teoretice despre Javadoc, instrumentul standard de documentare in Java, exemple practice si exercitii pentru exersarea utilizarii Javadoc.

1 Partea Teoretica

Timp de lucru estimat: 60 minute

1.1 Ce este Javadoc?

Javadoc este un instrument standard folosit pentru generarea automata a documentatiei API din comentarii special formate in codul sursa Java. Comentariile Javadoc sunt plasate deasupra claselor, constructorilor, metodelor si campurilor, si sunt procesate pentru a produce documentatie in format HTML.

1.2 Structura comentariilor Javadoc

Comentariile Javadoc au o structura specifica si sunt delimitate de `/**` si `*/`.

Exemplu:

```
/**
 * Aceasta este o clasa de exemplu.
 */
public class Exemplu {
    /**
     * Aceasta este o metoda de exemplu.
     *
     * @param param Un parametru de tip String.
     * @return Un mesaj de salut.
     */
    public String metodaExemplu(String param) {
        return "Salut, " + param + "!";
    }
}
```

1.3 Elemente de baza ale Javadoc

1.3.1 Descrierea sumara si detaliata

Comentariul Javadoc incepe cu o descriere sumara, urmata optional de o descriere detaliata.

```
/**
 * Descriere sumara.
 *
 * Descriere detaliata care poate contine mai multe
 * linii.
 */
```

1.3.2 Etichete Javadoc (Tags)

Etichetele Javadoc sunt utilizate pentru a furniza informatii suplimentare. Acestea sunt plasate dupa descrierea sumara si detaliata.

Etichete comune:

- `@author` - indica autorul clasei sau al metodei.
- `@version` - specifica versiunea clasei.
- `@param` - descrie un parametru al unei metode.
- `@return` - descrie valoarea returnata de o metoda.
- `@throws` sau `@exception` - indica exceptiile aruncate de o metoda.
- `@see` - ofera referinte catre alte clase sau metode.

Exemplu:

```
/**
 * Calculeaza suma a doua numere.
 *
 * @param a Primul numar.
 * @param b Al doilea numar.
 * @return Suma celor doua numere.
 */
public int suma(int a, int b) {
    return a + b;
}
```

1.4 Generarea documentatiei cu Javadoc

Pentru a genera documentatia folosind Javadoc, se foloseste comanda:

```
javadoc NumeFisier.java
```

Aceasta va crea un set de fisiere HTML care contin documentatia generata din comentariile Javadoc din codul sursa.

1.5 Practici bune in utilizarea Javadoc

- Scrieti comentarii Javadoc pentru toate clasele publice, metodele si campurile.
- Folositi un limbaj clar si concis in descrieri.
- Documentati parametrii si valorile returnate.
- Indicati exceptiile aruncate de metode.
- Folositi etichete @see pentru a crea legaturi catre clase sau metode relevante.

2 Partea Practica

Timp de lucru estimat: 60 minute

2.1 Exemple

2.1.1 Exemplul 1: Documentarea unei clase simple

Cod Java cu comentarii Javadoc:

```
/**
 * Clasa reprezinta un punct in spatiul
 *   bidimensional.
 *
 * @author
 * @version 1.0
 */
public class Punct {
    private double x;
    private double y;

    /**
     * Constructor implicit care initializeaza
     *   punctul la origine.
     */
    public Punct() {
        this.x = 0;
        this.y = 0;
    }

    /**
     * Constructor cu parametri.
     *
     * @param x Coordonata pe axa X.
     * @param y Coordonata pe axa Y.
     */
    public Punct(double x, double y) {
```

```

        this.x = x;
        this.y = y;
    }

    /**
     * Calculeaza distanta pana la un alt punct.
     *
     * @param altPunct Punctul pana la care se
     *         calculeaza distanta.
     * @return Distanța dintre cele doua puncte.
     */
    public double distantaLa(Punct altPunct) {
        double dx = this.x - altPunct.x;
        double dy = this.y - altPunct.y;
        return Math.sqrt(dx * dx + dy * dy);
    }
}

```

2.1.2 Exemplul 2: Documentarea unei metode care arunca exceptii

Cod Java cu comentarii Javadoc:

```

    /**
     * Clasa care ofera metode pentru operatii
     * matematice avansate.
     */
    public class MatematicaAvansata {
        /**
         * Calculeaza radicalul unui numar.
         *
         * @param numar Numarul pentru care se
         *         calculeaza radicalul.
         * @return Radicalul numarului.
         * @throws IllegalArgumentException Daca
         *         numarul este negativ.
         */
        public double radical(double numar) {
            if (numar < 0) {
                throw new
                    IllegalArgumentException("Numarul
                        trebuie sa fie
                        pozitiv.");
            }
            return Math.sqrt(numar);
        }
    }
}

```

2.2 Exerciții Propuse

1. Documentați clasa `ContBancar` din seminarul anterior folosind comentarii Javadoc. Asigurați-vă că toate metodele și constructorii sunt documentați corespunzător.

2. Creați o clasă `CalculatorGeometric` care conține metode pentru calculul ariei și perimetrului unor forme geometrice (cerc, pătrat, dreptunghi). Adăugați comentarii Javadoc pentru toate metodele.

3. Scrieți o clasă `ConversieTemperatura` care conține metode pentru conversia temperaturilor între Celsius, Fahrenheit și Kelvin. Documentați clasa și metodele folosind Javadoc.

4. Documentați o clasă `Vehicul` care are metode pentru pornire, oprire și accelerare. Incluzeti detalii despre parametri, valori returnate și excepții aruncate.

5. Creați o clasă `GestionareStudenti` care permite adăugarea, eliminarea și căutarea studenților într-o listă. Folosiți Javadoc pentru a documenta fiecare metodă și explicați funcționalitatea acesteia.

2.3 Instrucțiuni pentru Rezolvare

Pentru fiecare exercițiu:

- Scrieți codul Java al clasei, dacă nu există deja.
- Adăugați comentarii Javadoc deasupra claselor, constructorilor și metodelor.
- Folosiți etichetele Javadoc corespunzătoare (`@param`, `@return`, `@throws` etc.).
- Generați documentația folosind comanda `javadoc` și verificați rezultatul în fișierele HTML generate.