

Dokumentace javového kódu

Table of Contents

Dokumentace javových programů	1
Dokumentace javových programů	2
Pravidla komentování	2
Komentování uvnitř metod?	2
Typy komentářů	2
Dokumentační	3
Připomenutí z Pythonu	3
Nástroj javadoc	3
Značky nástroje javadoc	3
Co dokumentujeme	4
Dokumentace kódu uvnitř metod	4
Dokumentace metod	4
Dokumentace atributů	5
Dokumentace třídy	5
Komentář třídy	5
Komentář metody I	5
Komentář metody II	5
Kde uvádíme dokumentační komentáře	6
Problémy dokumentování	6

Dokumentace javových programů

- Dokumentace je *nezbytnou součástí* javových programů.
- Existuje mnoho druhů dokumentací dle účelu:

instalační

pokyny pro nasazení produktu

systemová

konfigurace, správa produktu

uživatelská

pro užívání produktu

vývojářská (programátorská)

pro údržbu, rozšiřování a znovupoužití

Dokumentace javových programů

- Zde se budeme věnovat především dokumentaci *programátorské*.
- To znamená pro ty, kteří budou náš kód využívat ve svých programech, rozšiřovat jej, udržovat jej.
- Komentáře říkají nejen *jak je kód psán a co dělá*, ale také a zejména *proč se to tak dělá*.
- Programátorské dokumentaci se říká *dokumentace API, javadoc*.
- Nejlepším příkladem je přímo [dokumentace Java Core API](#)

Pravidla komentování

Při psaní dokumentace dodržujeme tato pravidla:

- Jedná se hlavně o dokumentaci *pro ostatní*, tudíž prioritně dokumentujeme to, co je pro použití ostatními
- Dokumentujeme především celé třídy a jejich **public** a **protected** metody.
- Ostatní věci dle potřeby, například těžce pochopitelné nebo nelogické pasáže kódu.
- Dokumentaci píšeme *přímo do zdrojového kódu ve speciálních dokumentačních komentářích*.

Komentování uvnitř metod?

- Dvnitř metod nepíšeme většinou *žádné* komentáře, ale můžeme (obtížné části).
- Ideální ovšem je, když *uvnitř metod* žádný komentář být nemusí, neboť účel a funkčnost je z kódu zřejmá :-)
- Ve výuce (ale často i v praxi) budou komentáře ve vašem projektu vynucovány nástrojem *checkstyle*.

Typy komentářů

Řádkové

od značky `//` do konce řádku, nepromítnou se do dokumentace

```
// inline comment, no javadoc generated
```

Blokové

mohou být na libovolném počtu řádků, nepromítnou se do dokumentace

```
/* block comment,  
no javadoc generated */
```

Dokumentační

Libovolný počet řádků, promítnou se do dokumentace:

```
/**  
 * Documentary comment, javadoc generated  
 */
```

- Řádkové a blokové komentáře by měly být pouze *dočasné*.
- Ve výsledném kódu by měly být komentáře pouze *dokumentační*.

Připomenutí z Pythonu

```
# Simple comment, not going to documentation  
def my_function():  
    '''docstring comment of this function -  
    going to help and doc'''  
    return None  
  
print("Using __doc__:")  
print(my_function.__doc__)  
  
print("Using help:")  
help(my_function)
```

- Namísto *docstringů* používáme v Javě dokumentační komentáře `/** zde je */`

Nástroj javadoc

- Slouží ke strojovému generování dokumentace z dokumentačních komentářů a ze samotné struktury programu
- Dokumentace se vygeneruje do sady HTML souborů, takže to bude vypadat jako v [Java Core API](#).
- Používá speciální značky se znakem zavináč, např. `@author`, v dokumentačních komentářích

Značky nástroje javadoc

Javadoc používá značky vkládané do dokumentačních komentářů; hlavními jsou:

`@author`

specifikuje autora

@param

popisuje jeden parametr metody

@return

popisuje co metoda vrací

@throws

popisuje informace o výjimce, kterou metoda propouští ("vyhazuje")



V Javě 23 je už možné psát komentáře stylem Markdown, jak jsme zvyklí nejen z Github a další dokumentace. [Markdown in Documentation Comments](#)

Co dokumentujeme

- Dokumentujeme především *celé třídy*, zejména veřejné
- Jejich `public` a `protected` metody
- Lze dokumentovat i *celý balík* a to sepsáním souboru `package-info.html` v příslušném balíku
- V pořádných knihovnách (API) dokumentace k balíkům je

Dokumentace kódu uvnitř metod

- Řádkové nebo blokové komentáře vysvětlující smysl kódu.
- Jsou vidět pouze při detailním studiu zdrojového kódu metody.
- Pokud váš kód vyžaduje vysvětlující komentář, je něco špatně. Refaktorujte kód do čitelnější podoby a/nebo rozdělte metodu na menší metody (s `private` nebo `protected` viditelností).

Dokumentace metod

- U veřejných metod je dokumentace, zejména pak kontraktu (parametry, návratová hodnota) nutností.
- Dokumentace popisuje kontrakt pro použití metody, tj. vysvětluje význam a omezení vstupních argumentů a výstupních hodnot, chování metody apod.
- Veřejné metody, včetně konstruktorů, je nutné poctivě popsat dokumentačními komentáři, aby bylo použití metod jasné bez nutnosti studia zdrojového kódu.
- U neveřejných metod není strukturovaná dokumentace tak kritická, protože se používají převážně pouze v rámci dané třídy (případně podtříd).
- Při správném (popisném) pojmenování metod a parametrů tak nemusí být dokumentace nutná. Nicméně se rovněž doporučuje.
- Gettery a settery se nemusí dokumentovat, protože jejich význam je zcela jasný.

Dokumentace atributů

- Atributy jsou důležité datové položky, kterou souvisí stavem a životním cyklem objektu. A to i přesto, že jsou většinou privátní.
- Jejich dokumentace je proto velmi žádoucí.
- Atributy které ovlivňují použití nebo chování objektu by měly být popsány strukturovanými dokumentačními komentáři.
- Ostatní stačí řádkovými nebo blokovými.

Dokumentace třídy

- Účel každé třídy by měl být jasně popsán strukturovanými dokumentačními komentáři, aby bylo zřejmé její použití bez nutnosti studovat konkrétní metody nebo atributy.

Komentář třídy

```
/**
 * This class represents a point in two dimensional space.
 *
 * @author Petr Novotný
 **/
public class Vertex2D { ... }
```

Komentář metody I

Zkrácený oficiální komentář metody `toString()`:

```
/**
 * Returns a string representation of the object. In general, the
 * {@code toString} method returns a string that
 * "textually represents" this object. The result should ...
 *
 * @return a string representation of the object.
 */
public String toString() { ... }
```

Část `{@code toString}` značí formátování, vypíše to `toString` neproporcionálním písmem.

Komentář metody II

```
/**
```

```
* Returns the smaller of two int values.
* If the arguments have the same value, the result is that same value.
*
* @param a an argument.
* @param b another argument.
* @return the smaller of {@code a} and {@code b}.
*/
public int min(int a, int b) {
    return (a <= b) ? a : b;
}
```

Kde uvádíme dokumentační komentáře

1. Před hlavičkou *třídy* — pak komentuje třídu jako celek.
2. Před hlavičkou *metody* — pak komentuje příslušnou metodu.

- [How to Write Doc Comments for the Javadoc Tool](#)

Problémy dokumentování

Komentáře lžou

změním kód a zapomenu upravit komentář

```
/**
 * Calculates velocity.
 */
System.out.println(triangle.getArea());
```

Zbytečné komentáře

```
private int size; // creates size
```

Ideální je psát kód a názvy tříd a metod tak, aby se komentáře ani nemusely číst