

Vježbe 7 - stvaranje dokumentacije u *Java*-i

Matej Mihelčić

Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu

matmih@math.hr

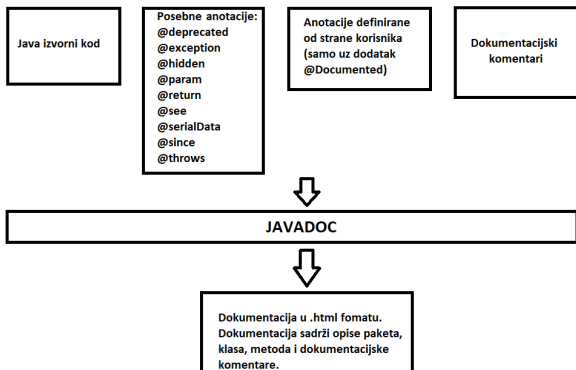
7. prosinca, 2022.



Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

Kreiranje dokumentacije je **aktivnost** koja se izvodi prije, tijekom i/ili nakon programiranja određenih funkcionalnosti programa. Glavni cilj pisanja dokumentacije je **opisati funkcionalnost i strukturu** programskog rješenja i **olakšati** njegovo **korištenje** drugim programerima ili korisnicima.

Programska dokumentacija za jezik *Java* se može automatski generirati korištenjem alata *Javadoc*.



Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

Dokumentacijski komentari se pišu u izvorni kod kao i programski komentari. Kao i kod programskih komentara, dokumentacijski komentari opisuju funkcionalnost nekog dijela koda, ali se prenose u generiranu dokumentaciju.

Programski komentari počinju oznakom `/**` a završavaju oznakom `*/`.

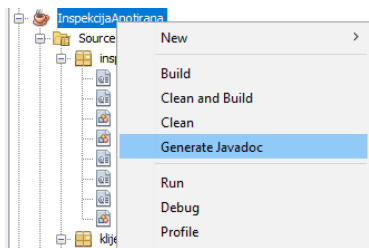
```
1  /**
2      *
3      * @param ins
4      * Niz inspektora
5      * <p><b>Opis:</b> Metoda <b> zaposliInspektora </b>
   dodaje inspektore
6      * iz polja <i> ins</i> među inspektore inspektorata.
7      * Metoda dodijeli inspektorima jedinstveni identifikator,
   postaju (na
8      * slucajan nacin) i inicijalizira optrecenje na 0.</p>
9      */
10 public void zaposliInspektora(Inspektor ...ins){...}
```

Primjer dokumentacijskog komentara.

Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

Dokumentacijski komentari mogu sadržavati `html` kod (masna i ukošena slova, poveznice, prijelaz u novi red, odjeljke itd.).

Dokumentacija se unutar okruženja *Apache NetBeans* generira tako da na trenutnom projektu napravimo desni klik -> **Generate Javadoc**.



Tako generirana dokumentacija **neće** sadržavati tekst anotacija `@author` i `@version`. Njih možemo dodati tako da u trenutnom projektu napravimo: desni klik -> **Properties**->**Build**->**Documenting**->**Document additional Tags** -> postaviti kvačicu na `@author` i `@version`.

Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

Project Properties - InspekcijaAnotirana

Categories:

- Sources
- Libraries
- Build
 - Compiling
 - Packaging
 - Deployment
 - Documenting
- Run
- Application
 - Web Start
- License Headers
- Formatting
- Hints

Include Private and Package Private Members

Generate:

- Class Hierarchy Tree
- Class and Package Usage Pages
- Navigation Bar
- Index
 - Separate Index per Letter
- HTML 5

Document Additional Tags:

- @author
- @version

Browser Window Title:

Additional Javadoc Options:
(e.g. -overview <file> or -header "Some header")

Preview Generated Javadoc

OK Cancel Help

Definirane anotacije koje se mogu prikazati unutar dokumentacije:

- `@author` - koristi se na razini paketa, klase ili pregleda. Dodaje se samo ukoliko se to **eksplicitno** definira unutar postavki. Označava autora paketa ili klase.
- `@param` - označava parametar konstruktora ili metode (tamo se i koristi).
- `@deprecated` - daje do znanja da se klasa ili metoda više **ne koristi** (zastarjela je). Može se koristiti u kombinaciji sa anotacijom `@see`.
- `@return` - opisuje što vraća metoda.
- `@see` - kreira listu **povezanog sadržaja**.
- `@since` - označava verziju od koje je neka značajka podržana.
- `@throws` - opisuje iznimku koju metoda prijavljuje. Mora obavezno biti naveden tip iznimke (ekvivalentan s i novija verzija anotacije `@exception`).
- `@hidden` - element nije opisan dokumentacijom.
- `@serialData` - opisuje tipove i poredak serijaliziranih podataka.

Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

Dokumentacija može sadržavati i anotacije **definirane od strane programera**. Podržane su sve vrste anotacija obrađene na predavanjima.

```
1 public @Documented @interface RazvojniTim {  
2     String[] value();  
3 }
```

Definicija anotacije razvojni tim.

Da bi se anotacija definirana od strane programera mogla prikazati unutar dokumentacije, anotacija mora biti označena **dodatnom anotacijom** `@Documented`.

```
1 @RazvojniTim({"Matej Mihelcic"})  
2 public void zaposliInspektora(Inspektor ...ins){...}
```

Korištenje anotacije u izvornom kodu.

Prethodno navedeni dokumentacijski komentari i anotacije definirane od strane programera su se mogle koristiti za opisivanje klasa, metoda, parametara itd. Da bi u dokumentaciji opisali **pakete** moramo napraviti poseban dokument `package-info.java` u koji zapisujemo dokumentacijske komentare i dodajemo odgovarajuće anotacije definirane od strane programera.

Da bi unutar okruženja *Apache NetBeans* kreirali navedeni dokument, unutar paketa radimo: desni klik -> New -> Other -> Java Package Info

Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

New File

Steps

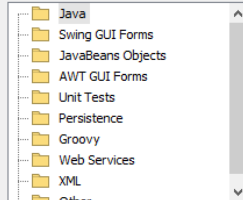
1. Choose File Type
2. ...

Choose File Type

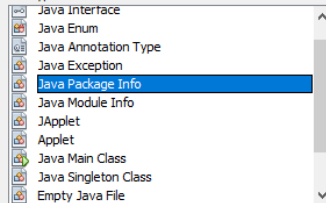
Project: InspekcijaAnotirana

Filter:

Categories:



File Types:



Description:

Creates a new Java package-info. To be able to use package annotations, your project must use JDK 5.0 or compatible.

< Back

Next >

Finish

Cancel

Help

Kreiranje dokumentacije u *Java*-i korištenjem alata *Javadoc*

```
1 /**
2  * <p> Paket <b> inspekcijaanotirana </b> nudi definiciju
3  *   klasa <b> Inspekcija </b>
4  *   i <b> Inspektor </b> koje su klucne za rad inspeksijske
5  *   sluzbe. Paket takoder sadrzi
6  *   definiciju nekih jednostavnijih anotacija.
7  * </p>
8  *
9  * @since 16.0
10 */
11 package inspekcijaanotirana;
```

Primjer sadržaja dokumenta package-info za paket inspekcijaanotirana.

Unutar komentara za dokumentaciju smijete koristiti i hrvatska slova, međutim ta slova će se unutar novo uvedenih anotacija zamijeniti odgovarajućim UTF kodovima.

Zadatak 1

Dokumentirajte kod koji implementira neke funkcionalnosti potrebne za rad inspeksijske službe.