

# Dokumentimi i API

Faton Berisha



Fakulteti i Shkencave Kompjuterike  
Universiteti i Prizrenit

# Përmbajtja

- 1 Hyrje
- 2 Dokumentimi i kodit burimor

# Hyrje

- Krijimi i referencave
- Përdorimi i veglës JavaDoc për krijimin e API të një kodi në Java si dokumentacion referencimi

# Programimi ndaj-dhe-pushto

- Rikujtoni parimet bazë të paradigmës OOP
  - Ndajeni programim në pjesë të menagjueshme që mund të implementohen, testohen të izoluar.
    - Struktura e komponentave
  - Definoni interfejsat për pjesët për të komunikuar me njëra tjetrën.
    - Programo sipas interfejsit, jo sipas implementimit
  - Sigurohuni se kontratat respektohen.
    - Klientët shfrytëzojnë interfejsat në mënyrë korrekte
    - Implementuesit implementojnë interfejsat në mënyrë korrekte (Testoni!)

# Programimi ndaj-dhe-pushto (Vazhdim)

- ...
  - Çelësi: dokumentimi i mirë i interfejsit
    - Problemi i gjuhëve programuese: interfejsi i klasës ndërlidhet me pjesën tjetër të definimit të klasës. Nevojitet prezantim i veçantë.
    - API si dokumentacion referencimi

# Dokumentimi i kodit burimor

- Probleme permanente
  - Shkruarja e API dokumentacionit për një sistem është njëra nga punët më të pakëndshme me të cilën do të përballet ndonjëherë një zhvillues.
    - Application Programming Interface
    - Punë e tipit që mund t'u shpie në dëshprim
  - Pa dokumentacion —→ pa kod
  - Dokumentacioni „joformal“ nuk është standard
    - Tutorialët, udhërrëfyesit tematikë
  - Duke evoluuar softueri, dokumentacioni „joformal“ dhe kodi bëhen të pasinkronizuar!
    - Më në fund, dokumentacioni bëhet i pashfrytëzueshëm që e bën kodin të vështirë për ta kuptuar dhe për ta përditësuar.

# JavaDoc

- *JavaDoc*: Sistem në gjuhën programuese Java për dokumentim kodi dhe prezantim të interfejsëve të kodit
- Vegla javadoc bën shkrimin dhe mirëmbajtjen e përditësuar të dokumentacionit të kodit më të lehtë!
  - Vjen me JDK
- Nxjerr dokumentacionin nga komponentat e kodit
  - Kërkon komente të formatizuara në mënyrë përkatëse

# JavaDoc (Vazhdim)

- Definon një bashkësi komentesh të formatizuara në mënyrë të posaçme
  - Mund të shtohen për të dokumentuar çdo
    - pako,
    - klasë (dhe interfejs),
    - metodë, dhe
    - fushë
- Përdoret për të gjeneruar HTML dokumentacion të klasave ose pakove pas parsimit të komenteve
  - Dokumentacion ne formatin HTML të API
  - Bazat e HTML: <http://www.w3schools.com/html>
- Dokumentacioni në JavaDoc mbi bibliotekat standarde të Java:  
<http://docs.oracle.com/javase/7/docs/api/>



# JavaDoc komentet

- *Menjëherë* i paraprijnë komponentës të cilën e përshkruajnë
- Konsistojnë nga
  - Përshkrimi i komponentës
    - Të kopjuar siç është në faqen e dokumentacionit
  - Lista e tagëve
    - Të formatizuar nga javadoc në stil konsistent
    - Të përdorur në klasa, interfejsa, metoda dhe fusha

# Formati i JavaDoc komenteve

```
/**  
 * Fjali përmbledhëse.  
 * Informatë më e përgjithshme mbi  
 * programin, klasën, metodën ose fushën  
 * e cila vijon pas komentit, duke përdorur  
 * sa nevojiten rreshta.  
 *  
 * zero ose më tepër tag për të specifikuar  
 * informata të llojit më specifik, si parametrat  
 * dhe vlerat kthyesë të një metode.  
 */
```

# JavaDoc komentet (Vazhdim)

- Duhet të jepen për çdo klasë ose interfejs *publik*
- Duhet të jepen për çdo metodë ose konstruktor *publik*

# JavaDoc tagët

- `@author emri`
  - Emërton njërin nga autorët e komponentës
  - Përdorni tag të shumëfishtë për më tepër autorë
  - Përdoret në klasë, interfejs, metodë.
  - P.sh., `@author Tringa Kosova, lab X`
- `@version versioni`
  - Indikon versionin e softuerit që përmban këtë komponentë.
  - Përdoret në klasë, interfejs.

## JavaDoc tagët (Vazhdim)

- @see *caku*
  - Drejto lexuesin nga diçka tjetër relevante për të lexuar, si ndonjë klasë tjetër ose metodë specifike
  - Inserton një link i cili pointon nga caku
  - Përdoret në të gjitha
  - P.sh., @author Tringa Kosova, lab X
- @deprecated *tekst*
  - Markon një komponentë si të vjetëruar dhe drejton lexuesin nga çfarë të përdorin në vend të saj përmes një @see tagu
  - Përdoret në të gjitha

# Tagu @see

- Drejton nga një tjetër pako, klasë, metodë, fushë ose madje URL
  - javadoc krijon një link
- Sintaksa për tekstin pas @see varet prej nga çfarë drejton:
  - @see *pakoja* → Link nga *pakoja*
  - @see *klasa* → Link nga *klasa* në pakon aktuale
  - @see *pakoja.klasa* → Link nga *klasa* në pakon *pakoja*
  - @see *#metoda* → Link nga *metoda* në klasën aktuale
  - @see *klasa#metoda* → Link nga *metoda* në klasën *klasa*
  - @see *pakoja.klasa#metoda* → Link nga *metoda* në klasën *klasa* në pakon *pakoja*
  - @see <a href=url>text</a> → Link nga URL *text*

# Shembuj

```
//komponente nga e njëjta klasë  
@see #field  
@see#Constructor(Type,...)  
@see #method(Type, Type,...)
```

```
//komponente nga e njëjta pako  
@see Class  
@see Class#field  
@seeClass#Constructor(Type,...)  
@see Class#method(Type,...)
```

## Shembuj (Vazhdim)

```
//komponente nga pako tjetër  
@see package  
@see package.Class  
@see package.Class#field  
@see package.Class#Constructor(Type,...)  
@see package.Class#method(Type, Type,...)  
  
//URL  
@see <a href="spec.html#section">Java Spec</a>
```



# JavaDoc tagët (Vazhdim)

- `@param` *parametri përshkrim i shkurtër*
  - Përshkruan parametrin *parametri* të kësaj metode
  - Evitoni tagun në qoftë se metoda nuk ka parametra
  - Përdorni tag të shumëfishtë për më tepër parametra
  - Përdoret në metoda
  - P.sh., `@param size` numri i elementëve ne vargun
- `@return` *tekst*
  - Përshkruan vlerën e kthyer nga kjo metodë
  - Evitoni tagun në qoftë se metoda nuk vlerë kthyesë
  - Përdoret në metoda
  - Paraqitet pas tagëve `@param`
  - P.sh., `@return` elementin më të madh të vargut

# Shembull

```
/**  
 * Validon një lëvizje shahu.  
 *  
 * @author Filan Fisteku  
 * @param theFromLoc lokacioni i figurës që lëvizet  
 * @param theRank rangi i figurës që lëvizet  
 * @param theToLoc lokacioni i fushës së caktuar  
 * @return true në qoftë se lëvizja valide, false përndryshe  
 */  
boolean isValidMove(int theFromLoc, int theRank, int theToLoc){  
    //...  
}
```

## JavaDoc tagët (Vazhdim)

- `@throws` *përrjashtimi përshkrim i situatës nën të cilën hidhet*
  - Përshkruan përrjashtimin *përrjashtimi* të papërpunuar
  - Evitoni tagun në qoftë se metoda nuk hedh përrjashtim
  - Përdorni tag të shumëfishtë për më tepër përrjashtime, në renditje alfabetike
  - Paraqitet pas tagëve `@param` dhe `@return`
  - Përdoret në metoda
  - P.sh., `@throws FileNotFoundException` në qoftë se emri i fajlit jovalid

## JavaDoc tagët (Vazhdim)

- <http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html>
- <http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html#javadoctags>

# Kërkesat minimale

- Klasat
  - @author
  - @version
- Metodat
  - @param
  - @return
  - @throws

# Ekzekutimi i Javadoc nga vija komanduese

- javadoc [opcionet] [pakot] [fajlat]
- Mund të listohen një ose më tepër pako
  - Nuk mund të përdoret \*
- Mund të listohen një ose më tepër java fajla
  - Mund të përdoret \*

# JavaDoc duke përdorur Eclipse

- Code formatter
  - Selektioni kodin
  - Sorce > Format
- JavaDoc comment generator
  - Selektioni projektin ose fajlin në Package Explorer
  - Project > Generate JavaDoc...
  - Ndiqni magjistarin

# Udhëzime për lexim të mëtejme

- <http://www.fberisha.org>
- <http://www.w3schools.com/html/>
- <http://www.w3schools.com/html/>
- <http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html>
- <http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html#javadocTags>



# Përfundim

- API si dokumentacion referencimi
- JavaDoc komentet
- JavaDoc tagët
- Krijimi i API me anë të JavaDoc