

The background of the slide is a solid dark blue. Overlaid on this are several sets of thin, white, curved lines that create a sense of motion and depth, resembling stylized waves or a wireframe mesh. These lines are more concentrated in the upper half of the slide and fade out towards the bottom.

# **Pisanje Javadoc dokumentacije**

# Vrste komentara u Javi

- Komentar u jednom retku

//Ovo je komentar u jednom retku

- Komentar kroz više redaka

/\*

Ovo je komentar

kroz

više

redaka

\*/

- Javadoc komentar

/\*\*

Ovo je javadoc komentar

\*/

# Javadoc dokumentacija

- Sadržava objašnjenja i detalje vezane uz programski kod i služi kao izvor podataka za generiranje API dokumentacije pomoću javadoc.exe alata
- Piše se iznad klasa i metoda:

```
/**  
 * Predstavlja entitet vozila koji je definiran brojem putnika i  
 * potrošnjom.  
 *  
 * @author Aleksander  
 */
```

```
public class Vozilo {...
```

```
/**  
 * Inicijalizira podatak o broju putnika i potrošnji.  
 *  
 * @param brojPutnika podatak o broju putnika  
 * @param potrosnja podatak o potrošnji  
 */
```

```
public Vozilo(int brojPutnika, float potrosnja) {  
    this.brojPutnika = brojPutnika;  
    this.potrosnja = potrosnja;  
}
```

# Primjer Javadoc dokumentacije (1/3)

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param url    an absolute URL giving the base location of the image
 * @param name   the location of the image, relative to the url argument
 * @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;
    }
}
```

## Primjer Javadoc dokumentacije (2/3)

- Rezultat generiranja Javadoc dokumentacije:

### **getImage**

```
public Image getImage(URL url, String name)
```

Returns an `Image` object that can then be painted on the screen. The `url` argument must specify an absolute URL.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

#### Parameters:

`url` - an absolute URL giving the base location of the image

`name` - the location of the image, relative to the `url` argument

#### Returns:

the image at the specified URL

#### See Also:

`Image`

## Primjer Javadoc dokumentacije (3/3)

- Rezultat pregleda HTML dokumentacije u pregledniku:

### **getImage**

```
public Image getImage(URL url)
```

Returns an `Image` object that can then be painted on the screen. The `url` that is passed as an argument must specify an absolute URL.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

#### **Parameters:**

`url` - an absolute URL giving the location of the image.

#### **Returns:**

the image at the specified URL.

#### **See Also:**

`Image`



# Javadoc i nasljeđivanje

- Javadoc dokumentaciju nije potrebno nanovo pisati u sljedećim slučajevima
  - Kod nadjačavanja metoda u podklasama (automatski se ubacuje oznaka “overrides”)
  - Kod nadjačavanja metoda u podsučeljima (automatski se ubacuje oznaka “overrides”)
  - Kod implementiranja metoda u klasama koje implementiraju metode iz sučelja (automatski se ubacuje oznaka “Specified by”)

## Prva rečenica u Javadoc dokumentaciji

- Svaka prva rečenica javadoc dokumentacije mora biti sažeta rečenica koja ukratko opisuje najvažnije karakteristike API elementa (ne smije ovisiti o drugim rečenicama, mora biti zasebna)
- API elementi su članske varijable, klase, sučelja ili opisi paketa
- Javadoc alat kopira tu prvu rečenicu i kod generiranja javadoc dokumentacije postavlja je iznad API elemenata



# Javadoc tagovi

- `@author` (koristi se kod klasa i sučelja, obvezan)
- `@version` (koristi se kod klasa i sučelja, obvezan)
- `@params` (metode i konstruktori)
- `@return` (metode)
- `@exception` (kod bacanja iznimaka - sinonim `@throws`)
- `@see` (koristi se kod referenciranja na druge izvore)
- `@since` (označava inačicu otkad je nešto uvedeno)
- `@serial` (specifikacija serijalizacije - ili `@serialField` ili `@serialData`)
- `@deprecated` (oznaka zastarjelosti)

## Stil pisanja Javadoc dokumentacije

- Kod spominjanja elemenata programskog koda potrebno je koristiti tag `<code></code>`
- Kod navođenja konstruktora ili metoda nije potrebno pisati zagrade (`metoda` umjesto `metoda()`)
- Potrebno je pisati u trećem licu (npr. dohvaća String oblik objekta)
- Opisi metoda moraju početi glagolom

## Detaljnije informacije

- <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>