

# PART 19

## 19 Javadoc

Java, üç tür comment(yorum) desteklemektedir. İlk ikisi // ve /\* \*/ dır. Üçüncü tür ise dökümantasyon yorumu olarak adlandırılır. /\*\* karakter dizi ile başlayıp, \*/ ile biter.

Dökümantasyon yorumları, programın kendisi hakkında program bilgi gömmenize yarar. Sonra javadoc hizmetini kullanarak, program hakkında bilgi çıkartıp bu bilgiyi bir HTML dosyasına koyabilirsiniz.

Dökümantasyon yorumları, programınızın dökümanını elverişli kılar.

### 19.1 Dökümantasyon Yorumları

/\*\* ile başladıktan sonra, ilk satır veya satırlar sınıfınızın, değişkeninizin veya metodunuzun ana açıklaması olur.

Ondan sonra, bir veya birden fazla @ etiketi ekleyebilirsiniz. Her @ etiketi, yeni satırın başlangıcında veya satırların başında (\*) asterisk sembolünden sonra gelebilir.

Bir sınıfın dökümantasyon yorumu örneği:

```
/**
 * This class draws a bar chart.
 * @author Zara Ali
 * @version 1.2
 */
```

### 19.2 javadoc Etiketleri

Bazı javadoc etiketleri:

Tag	Description	Example
@author	Sınıfın yazarını tanımlar	@author tanımlama
@exception	Metot tarafından atılmış istisnayı tanımlar.	@exception istisna-ismi açıklama
@param	Metot paramentleri gösterir.	@param parametre-ismi açıklama
@return	Metodun return değerini gösterir.	@return açıklama
@since	Belirli bir değişiklik yapıldıysa release'i vurgular.	@since release
@throws	@exception ile benzer.	@throws etiketi @exception etiketi ile aynı anlama gelmektedir.
@version	Sınıfın versiyonunu belirtir.	@version info

## 19.3 javadoc Çıktıları

javadoc programı, girdi olarak Java programınızın kaynak dosyasını alır ve çıktı olarak programın dökümantasyonunu içeren bir çok HTML dosyası alır.

Her sınıf için bilgi, kendisinin HTML dosyasında olacaktır. Java hizmeti olan **javadoc** ayrıca, bir index ve hiyerarşi ağacını da çıktı alır. Diğer HTML dosyaları üretilebilir.

### Örnek

Aşağıdaki, dökümantasyon yorumlarını kullanan örnek bir programdır. Her yorum, tanımladığı maddeden önce gelmektedir.

javadoc çalıştırıldıktan sonra, SquareNum sınıfı ile ilgili dökümantasyon SqureNum.html dosyasında bulunacaktır.

```
import java.io.*;
import java.util.Scanner;
```

```

/**
 * This class demonstrates documentation comments.
 * @author Ayan Amhed
 * @version 1.2
 */

public class SquareNum {

    /**
     * This method returns the square of num.
     * This is a multiline description. You can use
     * as many lines as you like.
     * @param num The value to be squared.
     * @return num squared.
     */
    public double square(double num) {
        return num * num;
    }

    /**
     * This method inputs a number from the user.
     * @return The value input as a double.
     * @see IOException
     */
    public double getNumber() {

        Scanner inData = new Scanner(System.in);
        String str = inData.nextLine();

        return Double.parseDouble(str);
    }

    /**
     * This method demonstrates square().
     * @param args Unused.
     * @return Nothing.
     */
    public static void main(String args[]) throws IOException
    {
        SquareNum ob = new SquareNum();
        double val;
        System.out.println("Enter value to be squared: ");
        val = ob.getNumber();
        val = ob.square(val);
        System.out.println("Squared value is " + val);
    }
}

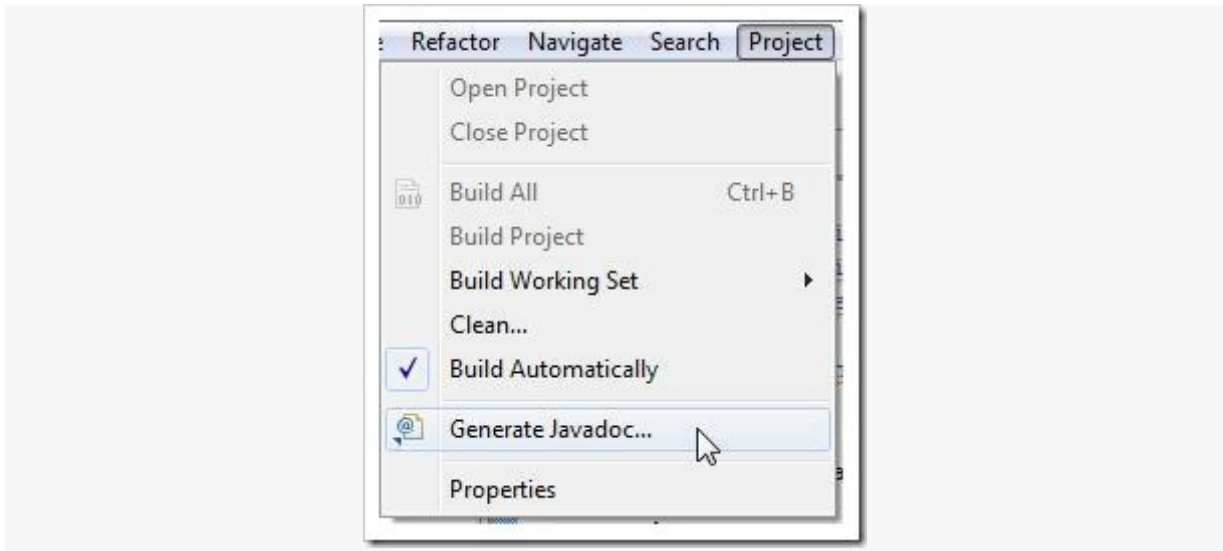
```

Yukarıdaki SquareNum.java dosyasını javadoc hizmeti ile birlikte çalıştıralım:

```
$ javadoc SquareNum.java
Loading source file SquareNum.java...
Constructing Javadoc information...
Standard Doclet version 1.5.0_13
Building tree for all the packages and classes...
Generating SquareNum.html...
Generating package-frame.html...
Generating package-summary.html...
Generating package-tree.html...
Generating constant-values.html...
Building index for all the packages and classes...
Generating overview-tree.html...
Generating index-all.html...
Generating deprecated-list.html...
Building index for all classes...
Generating allclasses-frame.html...
Generating allclasses-noframe.html...
Generating index.html...
Generating help-doc.html...
Generating stylesheet.css...
```

## 19.4 Eclipse'te JavaDoc Üretme

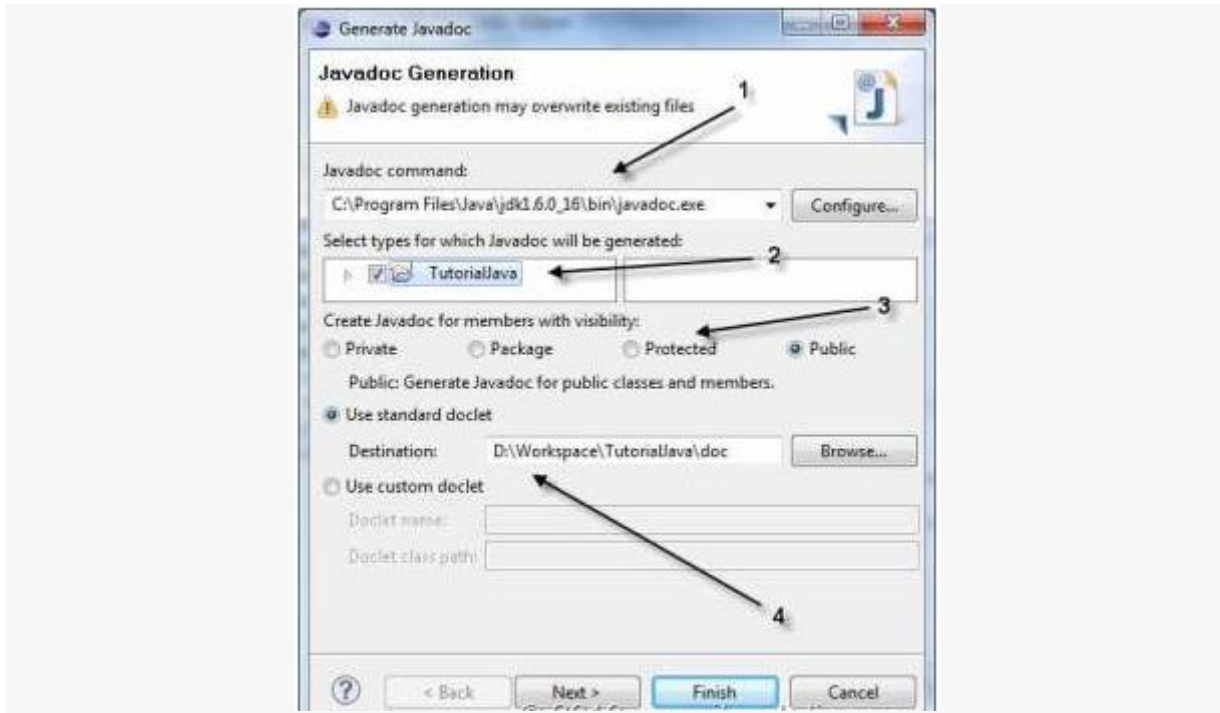
1. Eclipse Projesi açalım
2. Project -> Generate Javadoc kısmını seçelim



3. Sihirbazın ilk adımında, ayarları şu şekilde tanımlayalım:

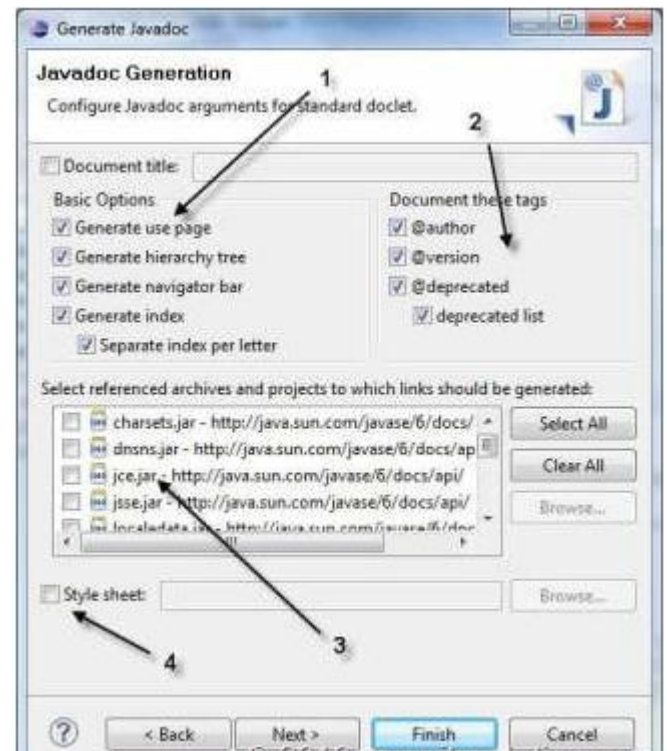
1. *javadoc.exe* aracı için JDK'dan yol
2. JavaDoc üreten proje kaynakları;
3. görünübilirliğine dayanarak JavaDoc üreten sınıflar ve metotlar;

4. Javadoc konumu (by default it will be placed in the *doc* folder in the project location)



4. İkinci adımda:

1. dökümantasyon yapısı;
2. İşlenecek Javadoc etiketleri;
3. diğer kaynaklar(archives, projects);
4. Dökümantasyon için, CSS style sheet;



5. Son adımda finish butonuna tıklayın.

