JAVA DOC

Java doc to narzędzie domyślnie dostarczane wraz z javą, które na podstawie kodu i dodatkowych opisów w postaci specjalnie sformatowanych komentarzy tworzy ustrukturyzowaną i spójną dokumentację kodu.

Aby zasygnalizować, że chcemy użyć komentarza typu javadoc musimy użyć następującej formy:

/ Komentarz */**

1. Komentarze możemy stosować do paczki, klasy, metody oraz zmiennej w klasie

```
/** Przykładowa paczka */
package com.example.app;

/** Przykład prostej klasy */
class Example {
    /** Komentarz dla zmiennej */
    public int i;
/**To jest prosty komentarz dla funkcji**/
    public boolean fun(Double x){
        //...
    }
}
```

2. Komentarz opisujący daną metodę, jej parametry oraz to co jest zwracane

```
/**
* Ta metoda zwraca kwadrat
* @param x liczba, która ma być podniesiona do kwadratu
* @return zwraca liczbę podniesiona do kwadratu
*/
public kwadrat(Double x){
    return x*x;
}
```

Dzięki takiemu opisowi metody łatwiej będzie w przyszłości nam zrozumieć za co dana metoda odpowiada.

Użyliśmy tagów param oraz return. W przypadku tagu @param musimy po spacji dodać argument, którego dotyczy.

Tag @return ma za zadanie informować użytkownika co dana funkcja zwraca.

3. Odsyłanie do innych obiektów oraz stron www

Są trzy sposoby na odsyłanie do innych obiektów

1. **/** @see Dokumentacja Java Oracle class Double */** Dzięki temu tagowi możemy dać użytkownikowi gdzie można szukać pomocy.
2. **/** @see Klasa bazowa @link BaseClass*/** Dzięki tagowi @link możemy tworzyć odnośniki

do innych klas lub metod w dokumentacji

3. Trzeci sposób wynika z tego, że javadoc generuje dokumentację w HTML więc możemy się odnieś do strony www.
   / @see Double */**

4. Inne znaczniki dokumentacyjne

@author autor – informacja o autorze
@version versia – informacja o wersji
@deprecated – Oznacza składowe, które zostały zastąpione przez nowsze wersje i używanie ich nie jest zalecane.
@throws wyjątek opis – opisuje wyjątek, który może zostać wyrzucony przez daną metodę
@exception = @throws

5. Generowanie dokumentacji

javadoc [opcje] nazwa pakietu lub javadoc [opcje] *.java

6. Wybrane opcje

- -overiew [ścieżka do strony HTML] – dodanie własnej strony głównej
- -public – dokumentowane są tylko klasy public i ich elementy składowe
- -protected j.w + protected(domyślny tryb)
- -private – dokumentowane są wszystkie klasy i ich elementy składowe
- -sourcepath [ścieżka] – katalog, w którym znajdują się dokumentowane pliki źródłowe
- -classpath [ścieżka] – określa katalogi w których znajdują się pliki .class dla klas, do których są odwołania w dokumentacji
- -exclude [lista pakietów] – lista pakietów, które mają być wykluczone z dokumentacji
- -encoding [nazwa] – określa nazwę kodowania pliku źródłowego
- -d [katalog] – określa katalog docelowy dla dokumentacji
- -version – uwzglednia znacznik @version
- -author – uwzględnia znacznik @author
- -notree – nie generuje hierarchii klas
- -header [kod HTML] – wstawia kod jako nagłówek dla każdej strony
- -footer [kod HTML] – wstawia kod jako stopkę dla każdej strony
- -doencoding [nazwa] – określa nazwę kodowania wynikowego pliku HTML

7. Ustawienie tworzenia dokumentacji jako task w Gradle

Tworzymy nowe zadanie

```
task javadoc(type: Javadoc) {
   source = android.sourceSets.main.java.srcDirs
   classpath += project.files(android.getBootClasspath().join(File.pathSeparator))
   destinationDir = file("../javadoc/")
   failOnError false
}
```

W edycji konfiguracji dodajemy by to zadanie się automatycznie uruchamiało.