

Kurs programowania

Wykład 7

Wojciech Macyna

Narzędzie do zapakowania wszystkich klas w jeden plik. Działa na podstawie kompresji zip. Zawiera plik `manifest.mf` w katalogu `meta-inf` który może opisywać pewne własności aplikacji.

Przykład pliku `manifest.mf`

```
Manifest-Version: 1.0
```

```
Main-Class: ShowFrame
```

Użycie: `jar [opcje] files`

`-c` – utwórz nowe archiwum

`-t` – wyświetl zawartość archiwum

`-u` – odśwież zawartość archiwum

`-x` – rozpakuj archiwum

`-v` – wyświetl pliki

`-f jar-file` – nazwa archiwum

`-m manifest` – dodaj podany manifest

Przykładowe wywołania

- Tworzenie pliku jar: `jar -cvfe ShowFrame.jar ShowFrame *.class`
- Wyświetlenie zawartości archiwum jar: `jar tf ShowFrame.jar`
- Rozpakowanie archiwum jar: `jar xf ShowFrame.jar`
- Wykonywanie archiwum jar: `java -jar ShowFrame.jar`

Więcej na ten temat:

<https://docs.oracle.com/javase/7/docs/technotes/tools/windows/jar.html>

Program javadoc

Java jest wyposażona w generator dokumentacji który na podstawie odpowiednio napisanych komentarzy w kodzie programu (które jednak nie są konieczne), generuje dokumentację w HTML-u.

Sposób generowania komentarzy do dokumentacji jest zgodny z większością standardów przestrzeganych również przez inne generatory dokumentacji.

Najważniejsze opcje javadoc

- `-overview <file>` – czytaj ogólny opis z pliku html
- `-public` – wyświetl tylko publiczne klasy
- `-protected` – wyświetl wszystko co nie jest prywatne
- `-private` – wyświetl wszystko
- `-encoding <name>` – ustawia kodowanie pliku wejściowego
- `-d <directory>` – katalog docelowy
- `-version` – dodaje paragrafy z dyrektywą `@version`
- `-author` – dodaje paragrafy z dyrektywą `@author`
- `-windowtitle <text>` – dodaje tytuł domyślny stron
- `-header <html-code>` – dodaje nagłówek stron
- `-footer <html-code>` – dodaje stopkę stron
- `-top <html-code>` – dodaje tekst na górze stron
- `-bottom <html-code>` – dodaje tekst na dole stron
- `-notree` – nie generuj hierarchii klas
- `-noindex` – nie generuj indeksu

Komentarz dokumentujący `/** ... */`

Umieszczany zazwyczaj przed dokumentowanym obiektem (klasą, funkcją, ...).

Pierwszy znak `*` w kolejnych wierszach jest pomijany.

Wewnątrz takiego komentarza mogą pojawić się bezpośrednio również znaczniki HTML-a które będą przepisane do wynikowego pliku dokumentacji. Jednak jak wskazuje praktyka lepiej tego unikać, szczególnie jeśli chcielibyśmy użyć innego generatora dokumentacji.

Wybrane dyrektywy dokumentacyjne

`@see`

Odwołanie się do innych miejsc w dokumentacji (system automatycznie tworzy odwołanie).

`@see nazwa-klasy`

`@see pelna-nazwa-klasy`

`@see pelna-nazwa-klasy#nazwa-metody`

`{@link pakiet.klasa#skladowa etykieta}`

Pozwala na umieszczenie odnośnika w innym miejscu niż sekcja **See Also**.

`@version`

Po dyrektywie umieszczamy informacje istotne dla wersji.

`@author`

Informacje o autorze. Wyświetlane jeśli generowanie jest z parametrem `-author`.

Wybrane dyrektywy dokumentacyjne

@param

Opisuje parametry używane przez metody i ma postać:

```
@param nazwa-parametru opis
```

Opis kończy się w momencie wystąpienia innej dyrektywy lub końca komentarza.

@return

Opisuje wynik zwracany przez metodę i ma postać:

```
@return opis
```

Opis kończy się w momencie wystąpienia innej dyrektywy lub końca komentarza.

@throws

Opisuje wyjątki generowane przez metodę i ma postać:

```
@throws pełna-nazwa-klasy-wyjatku opis
```

Opis kończy się w momencie wystąpienia innej dyrektywy lub końca komentarza.

`@deprecated`

Sugeruje aby nie używać danej metody gdyż w przyszłości może zostać usunięta.

Przykładowe wywołania

Tworzy dokumentację, przy czym każda strona ma napis "Page top" (italics), zastosowano kodowanie UTF-8 oraz wyświetlane są atrybuty prywatne:

```
javadoc -top "<p><i> Page top </i></p>" -encoding cp1250  
-charset cp1250 -private *.java
```

Doxygen

System generowania dokumentacji przystosowany do większości stosowanych języków programowania, zwracający dokumentację w wielu formatach (html, latex -> pdf, rtf, man, ...).

Przykłady wygenerowanej dokumentacji za pomocą javadoc dla plików `.java` jak również za pomocą Doxygena dla plików `.java`, `.cpp` oraz `.py` znajdują się w kodach do tego wykładu.