

Specyfikacje JAVA

Java to termin, który w dziedzinie informatyki kojarzony jest z:

- platformą do budowy aplikacji, oferującą programy i narzędzia wspierające budowę aplikacji w języku Java i ich uruchamianie (ang. *Java Platform*)
- obiektywnym językiem programowania (ang. *Java language*)
- wirtualną maszyną, interpretującą kod bajtowy będący wynikiem kompilacji kodu źródłowego napisanego w języku Java (ang. *Java Virtual Machine*)

Java powstała w 1995 roku, ale jej początki sięgają roku 1991. Historię języka można prześledzić na stronach:

- <https://www.geeksforgeeks.org/the-complete-history-of-java-programming-language/>
- [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Rozwój Javy odbywa się w ramach programu JCP (ang. *Java Community ProcessSM (JCPSM) Program*, <https://www.jcp.org/en/home/index>), w którego każdy uczestnik może brać udział w recenzowaniu oraz dostarczaniu informacji zwrotnej do publikowanych kolejnych specyfikacji JSR (ang. *Java Specification Requests (JSRs)*).

Na stronach JCP można znaleźć, na przykład, specyfikację:

- JSR 384: Java™ SE 11 (18.9) (<https://www.jcp.org/en/jsr/detail?id=384>)

Specyfikacje często składają się z powiązanych ze sobą dokumentów. Na przykład specyfikację platformy (*The Java SE 11 (18.9) Platform Specification*) zbudowano na specyfikacji języka (ang. *Java Language Specification*), wirtualnej maszyny (ang. *Java Virtual Machine Specification*) oraz API (ang. *Java SE APIs*).

Poszukując źródłowych dokumentów tych specyfikacji można znaleźć:

The Java Language Specification (JLS) <http://cr.openjdk.java.net/~iris/se/11/spec/fr/java-se-11-fr-spec/java-se-11-jls-draft-diffs.pdf>

The Java Virtual Machine Specification (JVMS)
<http://cr.openjdk.java.net/~iris/se/11/spec/fr/java-se-11-fr-spec/java-se-11-jvms-draft-diffs.pdf>

Complete API Specification

<http://cr.openjdk.java.net/~iris/se/11/spec/fr/java-se-11-fr-spec/api/index.html>)

API Specification differences (porównanie)

<http://cr.openjdk.java.net/~iris/se/11/spec/fr/java-se-11-fr-spec/apidiffs/overview-summary.html>

Wyszukując specyfikacje dla poszczególnych wersji Java SE (<https://jcp.org/en/jsr/summary?id=JAVA+TM+SE>) można znaleźć:

JSR#	Opis
337	Java™ SE 8 Release Contents (https://download.oracle.com/otndocs/jcp/java_se-8-fr-eval-spec/index.html)
379	Java™ SE 9 Release Contents
383	Java™ SE 10 (18.3)

384	JavaTM SE 11 (18.9) (https://download.oracle.com/otndocs/jcp/java_se-11-final-spec/index.html)
386	JavaTM SE 12
388	JavaTM SE 13
389	JavaTM SE 14
390	JavaTM SE 15
391	JavaTM SE 16
392	JavaTM SE 17

Specyfikację języka Java oraz wirtualnej maszyny Java (w formacie pdf i html) znajduje się również na stronach Oracle, pod adresem:

- <https://docs.oracle.com/javase/specs/index.html>

Jeśli chodzi o specyfikację Java API, to dość ciekawie wygląda konflikt między Google i Oracle:

https://en.wikipedia.org/wiki/Google_LLC_v._Oracle_America,_Inc.

Budowanie JDK ze źródeł

W ogólnym ujęciu do zbudowania własnej dystrybucji JAVA potrzebne są: kod źródłowy, narzędzia (w tym kompilator) i zależności (dodatkowe biblioteki). Co więcej, do kompilacji JAVA wymagana jest wersja dystrybucyjna JAVA (o tym samym, co budowana lub wcześniejszym numerze).

Referencyjną implementację specyfikacji Java z kodami źródłowymi opublikowano na stronach społeczności OpenJDK (<http://openjdk.java.net/>).

Sposób budowania opisano w oficjalnym dokumencie <http://openjdk.java.net/groups/build/doc/building.html>

Źródła referencyjnej implementacji JDK 11 (*The source code of the Reference Implementation binaries is available under the GPLv2*) udostępniono pod adresem:

https://download.java.net/openjdk/jdk11/ri/openjdk-11+28_src.zip

Budowa pod Windows wymaga pewnych dodatkowych zabiegów i narzędzi (środowiska Cygwin, kompilatora MS Visual Studio – i to odpowiedniej wersji). O zależnościach można przeczytać na stronie: <https://wiki.openjdk.java.net/display/Build/Supported+Build+Platforms>

Dobry tutorial dotyczący budowania openJDK opisano na stronie: <https://stuefe.de/posts/build-openjdk-on-windows/>

W szczególności dobra jest sekcja o budowaniu z pobraniem źródeł narzędziem `wget`.

Sekwencja akcji prowadząca do skompilowania źródeł (w terminalu `cygwin`):

```
$ mkdir openjdk11
$ cd openjdk11/
$ wget https://builds.shipilev.net/workspaces/jdk-updates-jdk11u.tar.xz
$ tar -xf jdk-updates-jdk11u.tar.xz $ tar -xf jdk-updates-jdk11u.tar.xz
$ mv jdk-updates-jdk11u source
$ cd source/
```

```
$ hg pull
$ hg update
$ cd ..
$ mkdir output
$ cd output/
$ bash ../source/configure --with-boot-jdk=/cygdrive/e/proper-
path/jdk-11.0.1 --with-debug-level=release --with-target-bits=64 --
with-toolchain-version=2017
```

tutaj trzeba było wyedytować wygenerowany automatycznie plik spec.gmk, bo z jakiegoś powodu java, javac, javadoc, jar i inne narzędzia nie chciały się odpalić bez rozszerzenia .exe (mimo odpalania ich z użyciem fixpath.exe).

```
JAVA_CMD:=/cygdrive/e/Development/Java/jdk-11.0.1/bin/java.exe
JAVAC_CMD:=/cygdrive/e/Development/Java/jdk-11.0.1/bin/javac.exe
JAVADOC_CMD:=/cygdrive/e/Development/Java/jdk-11.0.1/bin/javadoc.exe
JAR_CMD:=/cygdrive/e/Development/Java/jdk-11.0.1/bin/jar.exe
JLINK_CMD := $(JDK_OUTPUTDIR)/bin/jlink.exe
JMOD_CMD := $(JDK_OUTPUTDIR)/bin/jmod.exe
JARSIGNER_CMD:=/cygdrive/e/Development/Java/jdk-
11.0.1/bin/jarsigner.exe
SJAVAC_SERVER_JAVA_CMD:=/cygdrive/e/Development/Java/jdk-
11.0.1/bin/java.exe
```

```
$ make images
```

Oczywiście wcześniej należało zainstalować cygwin (z wszystkimi wymaganymi pakietami) oraz MS Visual Studio 2017. Jak widać, do kompilacji openJDK 11 (w wersji bieżącej) użyto wcześniej pobranej wersji dystrybucyjnej JDK 11.0.1. Ta zależność może wydać się dziwna – budując JDK trzeba mieć JDK wersji wcześniejszej. Ale tak właśnie jest. Spora część implementacji referencyjnej jest stworzona w języku Java.