

Języki Programowania

dr inż. Tomasz Kubik

tomasz.kubik.staff.iiar.pwr.edu.pl

Program kursu

1. Wprowadzenie do języka Java poprzez porównanie z językami C/C++. Kod bajtowy i wirtualna maszyna. Kompilacja i uruchamianie aplikacji w zintegrowanym środowisku programowania.
2. Klasy, interfejsy, obiekty i cykl ich życia, typy podstawowe i referencje, strukturalna obsługa wyjątków, wzorce projektowe.
3. Technologie składowe i podstawowe pakiety klas platformy Java SE (w tym: ciągi znaków, tablice i kolekcje, strumienie).
4. Model obsługi zdarzeń. Budowa graficznego interfejsu użytkownika.
5. Realizacja wielowątkowości, sekcja krytyczna i wzajemne wykluczanie.
6. Pakiet klas do realizacji połączeń sieciowych.
7. Elementy programowania rozproszonego (architektura klient-serwer, mechanizmy zabezpieczeń), zdalne wywoływanie procedur.
8. Repetytorium.

Zasady zaliczania

- Na kurs składają się
 - laboratoria (rozliczane na podstawie wykonanych zadań programistycznych)
 - wykład (rozliczany na podstawie wyników kolokwium przeprowadzonego w formie testu)
- Aby można było przystąpić do testu z wykładu należy pozytywnie zaliczyć laboratoria (tj. uzyskać pozytywną ocenę za tę część kursu)
- Ocena za kurs to średnia ważona ocen:
$$P = 0,6 * F1 + 0,4 * F2$$
gdzie F1 to ocena za część laboratoryjną,
a F2 to ocena za część wkładową

Narzędzia programowe

- JDK (ang. Java Development Kit)
 - wersja
 - obowiązkowo JDK 11 (11.02)
 - fakultatywnie JDK 17
 - dystrybucja
 - zalecane OpenJDK
 - <http://openjdk.java.net/>
 - <http://jdk.java.net/17/>
 - <http://jdk.java.net/11/>
 - <http://jdk.java.net/archive/>
 - opcjonalnie JDK from Oracle
 - <https://www.oracle.com/java/technologies/downloads/>- JRE (ang. Java Runtime Environment)



JDK 11 Releases

The JDK 11 Early Access Program has concluded. Please visit Java SE Downloads for production ready builds.

Older releases, **which do not include the most up to date security vulnerability fixes and are no longer recommended for use in production**, remain available in the OpenJDK Archive.

Java 17 available now

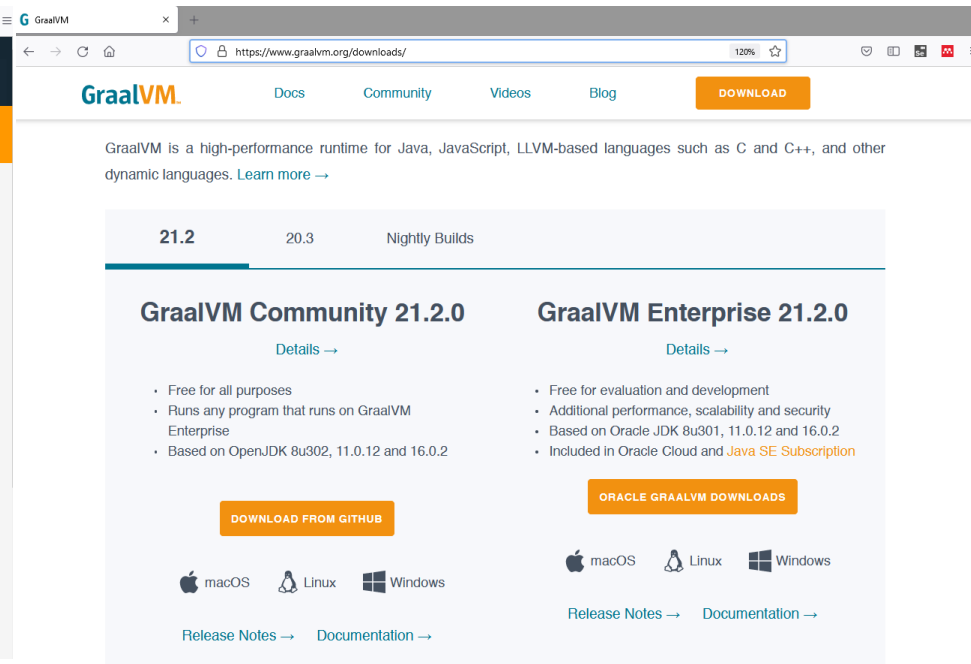
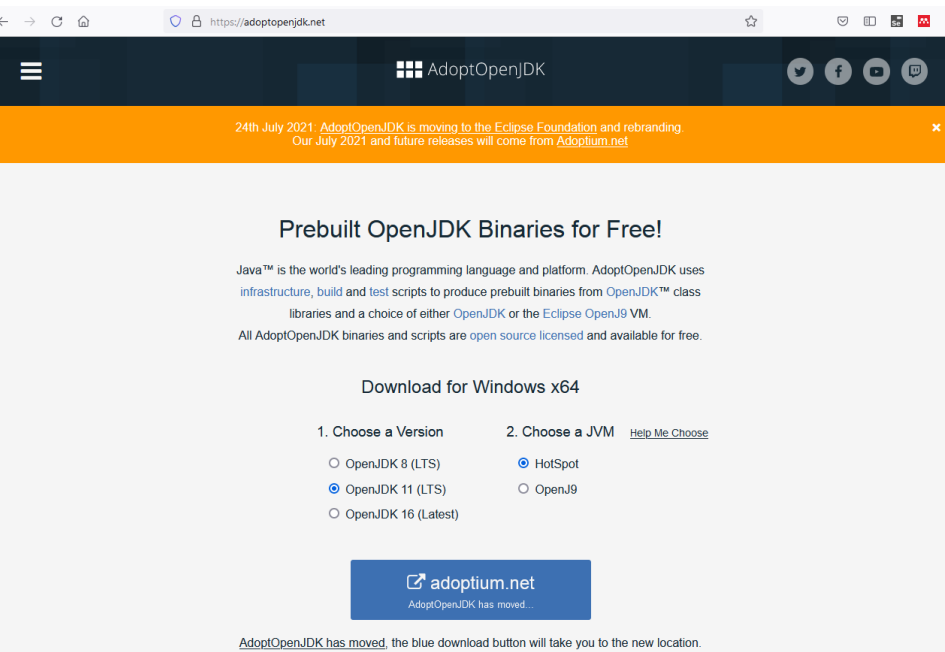
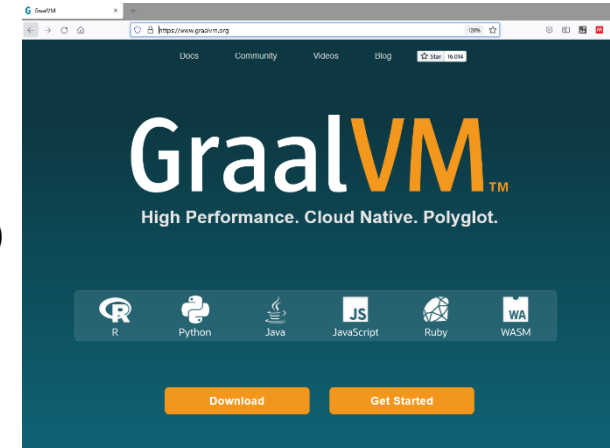
[Learn about Java SE Subscription](#)

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions License](#).

JDK 17 will receive updates under these terms, until at least September 2024.

Narzędzia programowe

- JDK (ang. Java Development Kit)
 - dystrybucja
 - GraalVM (środowisko wspierające kilka języków)
 - <https://www.graalvm.org/>
 - <https://www.graalvm.org/downloads/>
 - AdoptOpenJDK
 - <https://adoptopenjdk.net/>



Narzędzia programowe

<https://wiki.openjdk.java.net/display/JDKUpdates/JDK11u>

- Wsparcie JDK

Oracle Java SE Support Roadmap^{††}

Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Support
7 (LTS)	July 2011	July 2019	July 2022 ^{*****}	Indefinite
8 (LTS) ^{**}	March 2014	March 2022	December 2030 ^{*****}	Indefinite
9 (non-LTS)	September 2017	March 2018	Not Available	Indefinite
10 (non-LTS)	March 2018	September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	September 2026	Indefinite
12 (non-LTS)	March 2019	September 2019	Not Available	Indefinite
13 (non-LTS)	September 2019	March 2020	Not Available	Indefinite
14 (non-LTS)	March 2020	September 2020	Not Available	Indefinite
15 (non-LTS)	September 2020	March 2021	Not Available	Indefinite
16 (non-LTS)	March 2021	September 2021	Not Available	Indefinite
17 (LTS)	September 2021	September 2026 ^{****}	September 2029 ^{****}	Indefinite
18 (non-LTS) ^{***}	March 2022	September 2022	Not Available	Indefinite
19 (non-LTS) ^{***}	September 2022	March 2023	Not Available	Indefinite
20 (non-LTS) ^{***}	March 2023	September 2023	Not Available	Indefinite
21 (LTS) ^{***}	September 2023	September 2028	September 2031	Indefinite

* Oracle Java SE product dates are provided as **examples** to illustrate the support policies. Customers should refer to the [Oracle Lifetime Support Policy](#) for the most up-to-date information. Timelines may differ for Oracle Products with a Java SE dependency ([My Oracle Support Note 1557737.1 - Support Entitlement for Java SE When Used As Part of Another Oracle Product](#) – Requires Support Login).

** These support timelines apply to Java client and server deployments of Java with the exception of Web Deployment Technology and JavaFX. For more information on those features, see below.

*** LTS designation and dates, as noted in the above example, are subject to change.

**** Or later.

***** The Extended Support uplift fee will be waived for the period June 2019 - July 2022 for Java SE 7. The Extended Support uplift fee will be waived for the period March 2022 - December 2030 for Java SE 8. During this period, you will receive Extended Support as described in the Oracle Technical Support Level sections of the Technical Support Policies.

† Excluding Deployment Technology and JavaFX, which is described in a separate section.

<https://www.oracle.com/java/technologies/java-se-support-roadmap.html>

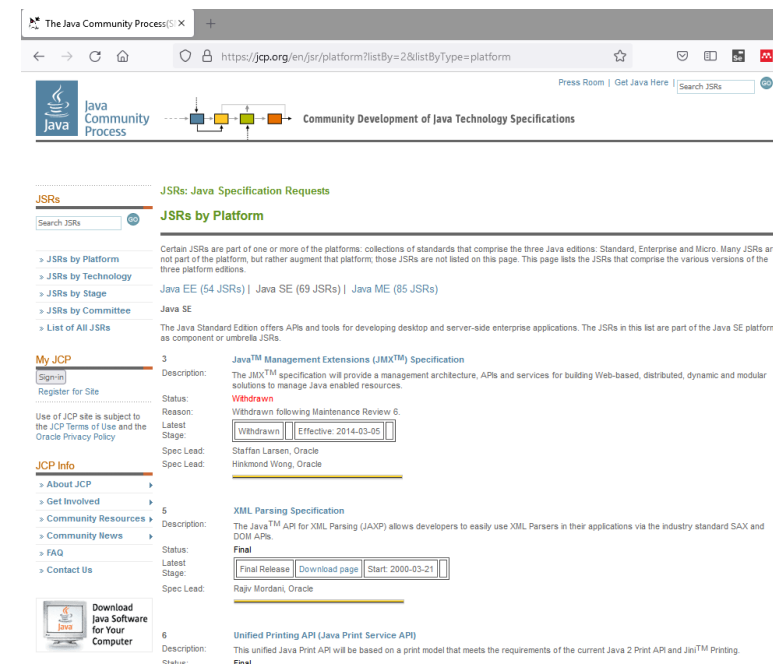
<https://www.java.com/releases/>

<https://www.codejava.net/java-se/java-se-versions-history>

<https://endoflife.date/java>

Źródła

- Standaryzacja
 - Java Community Process
 - <https://jcp.org/en/home/index>
 - Specyfikacja języka
 - <https://docs.oracle.com/javase/specs/>



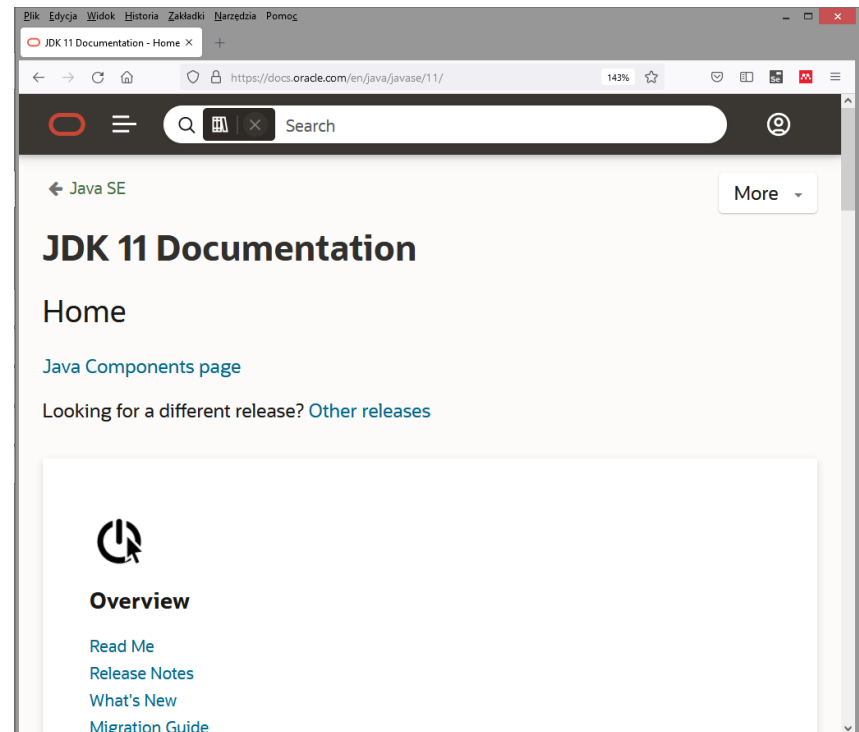
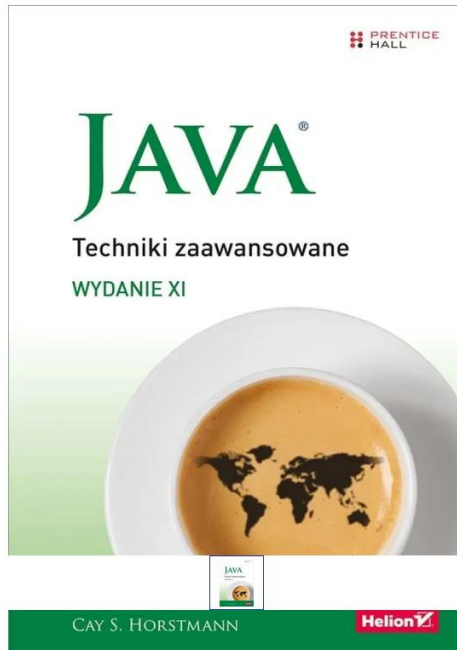
- Kod źródłowy JDK
 - <https://github.com/openjdk/jdk>
 - <https://github.com/openjdk/jdk/tree/jdk-11%2B28/src/hotspot/share/gc>
 - <https://openjdk.java.net/groups/build/doc/building.html>



Aby skompilować nową wersję JDK trzeba mieć JDK w wersji równej lub mniejszej o 1 (!)

Źródła

- Dokumentacja
 - <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>
 - <https://docs.oracle.com/en/java/javase/11/>
- Książki
 - Cay S. Horstmann: Java. Techniki zaawansowane. Wydanie 11



Narzędzia programowe

- Eclipse IDE

- <https://www.eclipse.org/downloads/>
- <https://www.eclipse.org/downloads/packages/>

The image displays two screenshots of the Eclipse website. The left screenshot shows the main Eclipse Foundation homepage, featuring a banner for 'Download Eclipse Technology that is right for you' and a section for 'Get Eclipse IDE 2021-09'. The right screenshot shows the 'Eclipse Packages' page for Eclipse IDE 2021-09 R, detailing download links for various operating systems and a list of available packages.

Left Screenshot: Eclipse Foundation Homepage

- Header: Eclipse Foundation, Projects, Working Groups, Members, More-
- Banner: Download Eclipse Technology that is right for you
- Section: Get Eclipse IDE 2021-09. Install your favorite desktop IDE packages. Download x86_64.
- Section: Tool Platforms. Eclipse Che, Orion.
- Section: IoT & Edge Developer Survey 2021.

Right Screenshot: Eclipse Packages Page

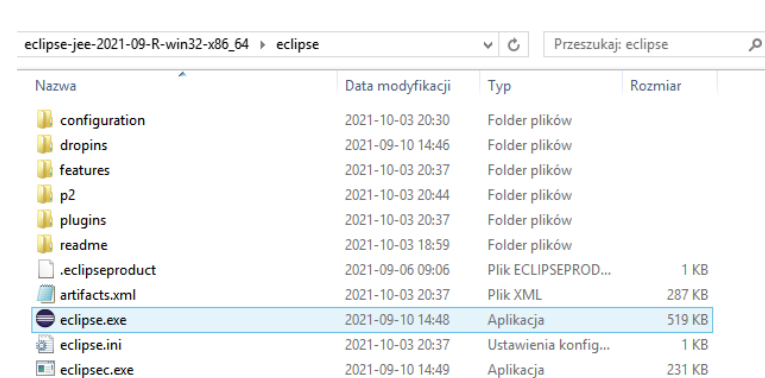
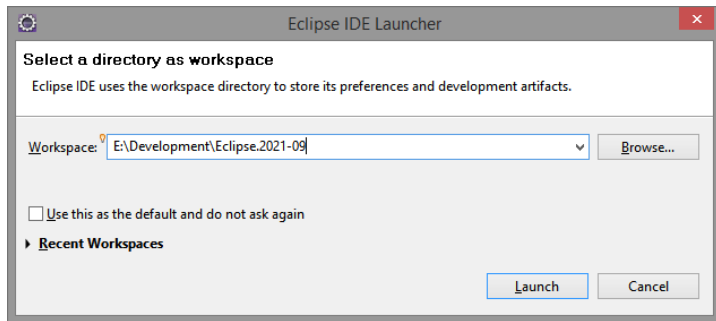
- Header: Eclipse Foundation, Projects, Working Groups, Members, More-
- Breadcrumb: Home / Downloads / Packages / Release / Eclipse IDE 2021-09 / R
- Section: Try the Eclipse Installer 2021-09 R. The easiest way to install and update your Eclipse Development Environment. Download links for macOS x86_64, Windows x86_64, Linux x86_64 | AArch64.
- Section: Eclipse IDE 2021-09 R Packages.
- Package: Eclipse IDE for Java Developers. 322 MB, 387,100 DOWNLOADS. Windows x86_64, macOS x86_64, Linux x86_64 | AArch64.
- Package: Eclipse IDE for Enterprise Java and Web Developers. 517 MB, 255,805 DOWNLOADS. Windows x86_64, macOS x86_64, Linux x86_64 | AArch64.

Eclipse IDE - launch

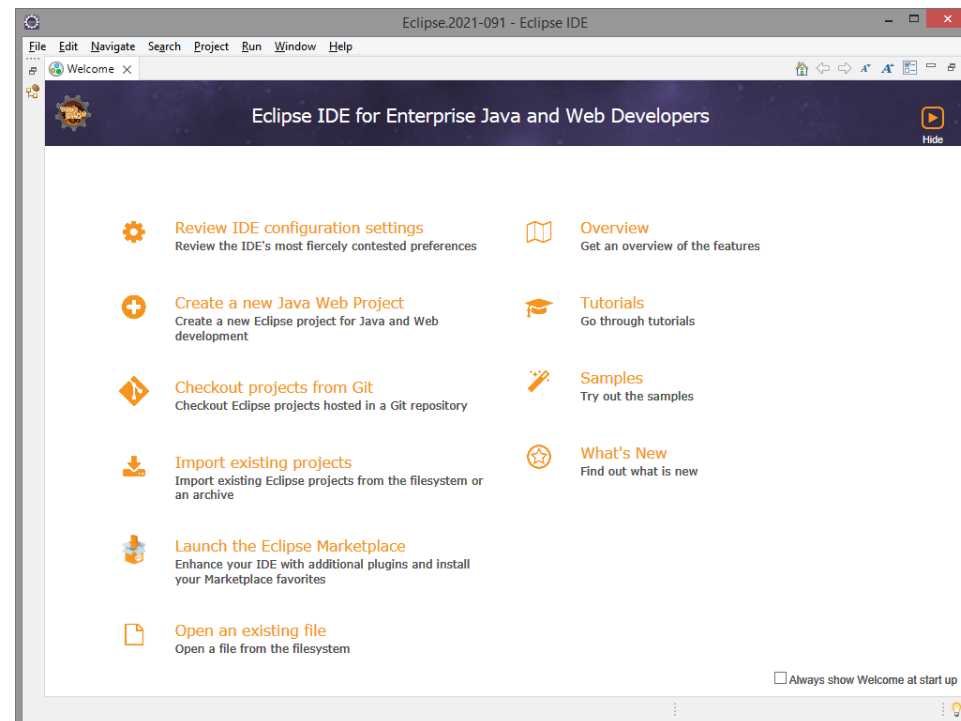
Uruchomienie

- przez klik na `eclipse.exe` w menadżerze plików lub
- z linii komend w katalogu `eclipse`:
> `eclipse.exe`

Podczas uruchomienia pojawia się monit o wybranie przestrzeni roboczej

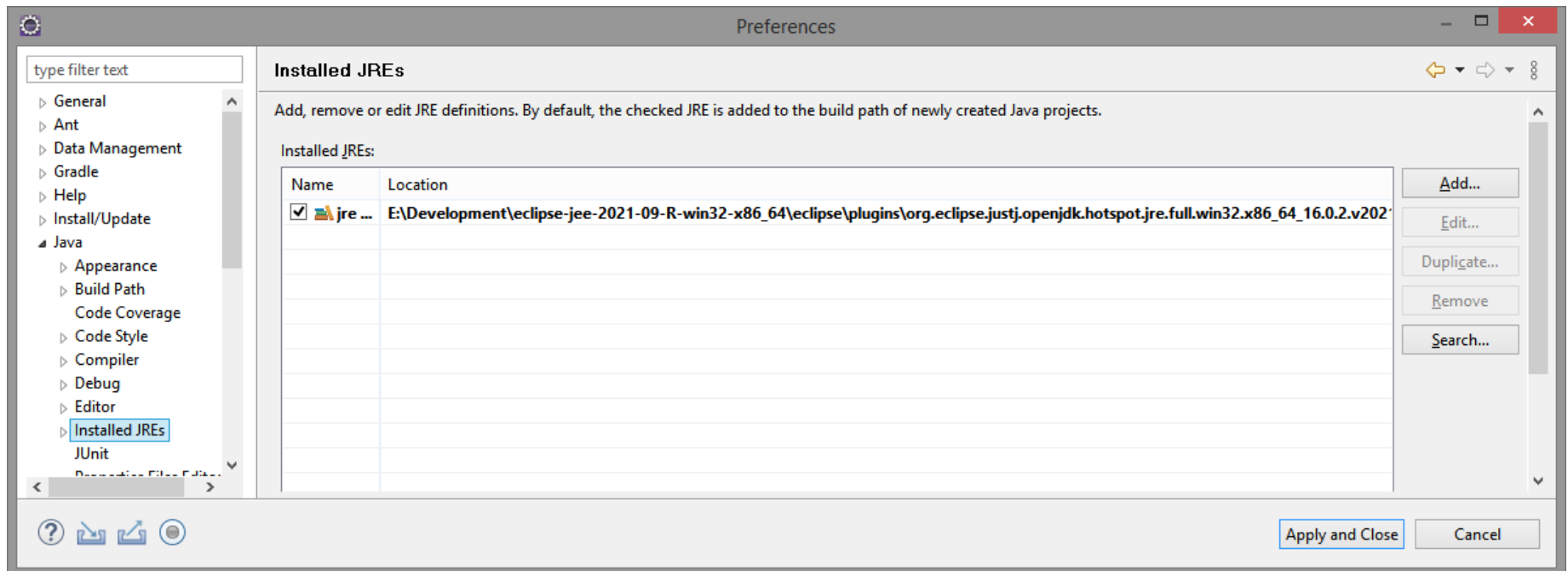
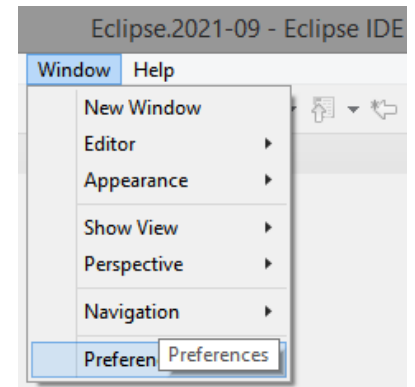


Po uruchomieniu pojawia się okno powitalne



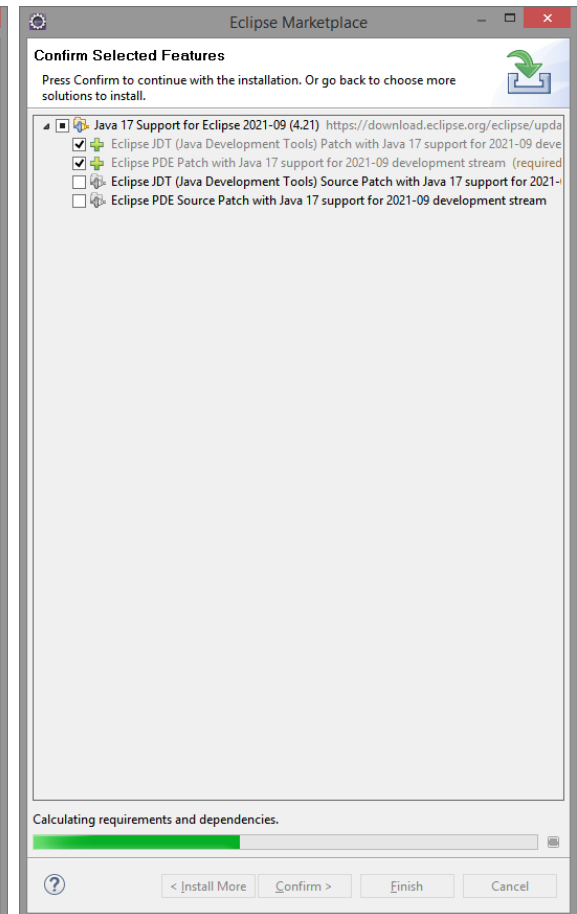
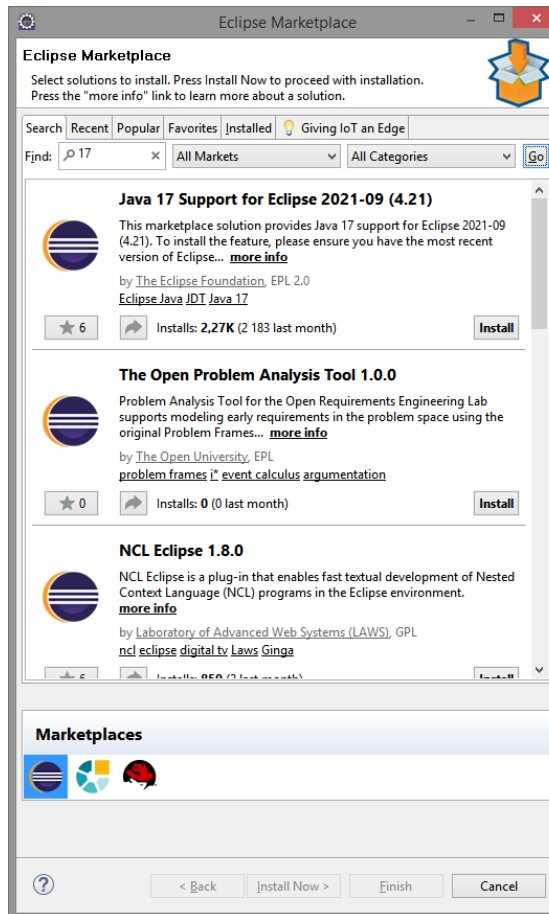
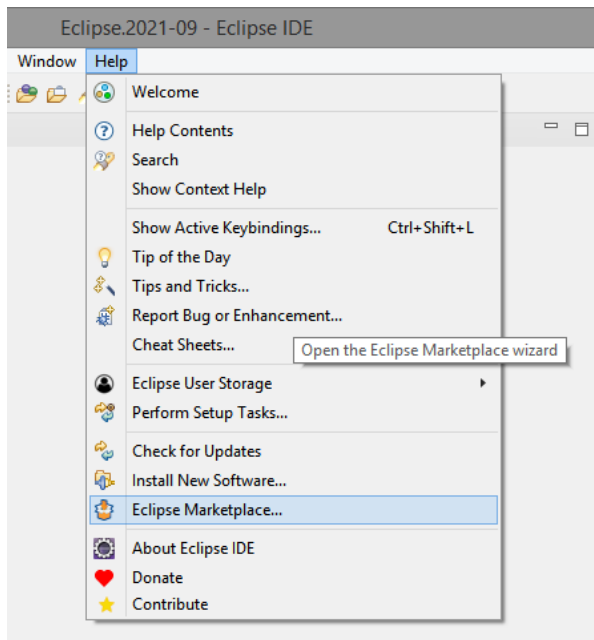
Eclipse IDE - Java

- Środowisko Eclipse 2021-09 (4.21) dostarczane jest razem z JRE w wersji 16
- Można się o tym przekonać sprawdzając preferencje



Eclipse IDE – Java 17 Support

- Eclipse 2021-09 (4.21) wspiera składnię Java do wersji 16.
- Aby uzyskać wsparcie wersji 17 należy zainstalować odpowiednie rozszerzenie



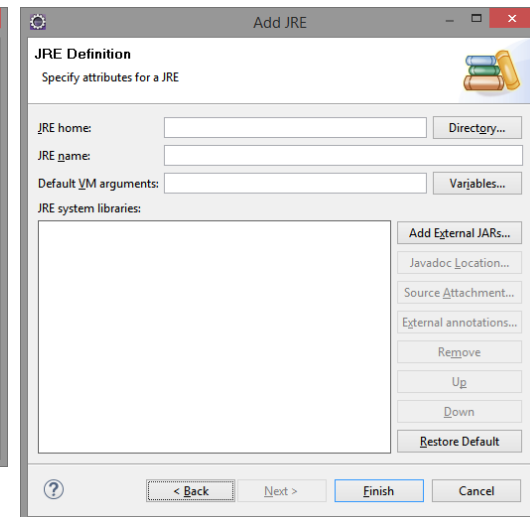
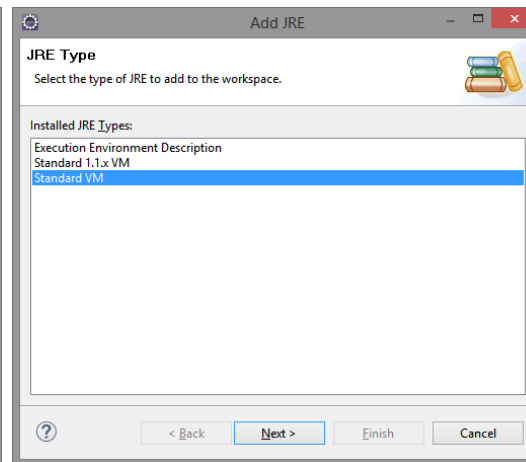
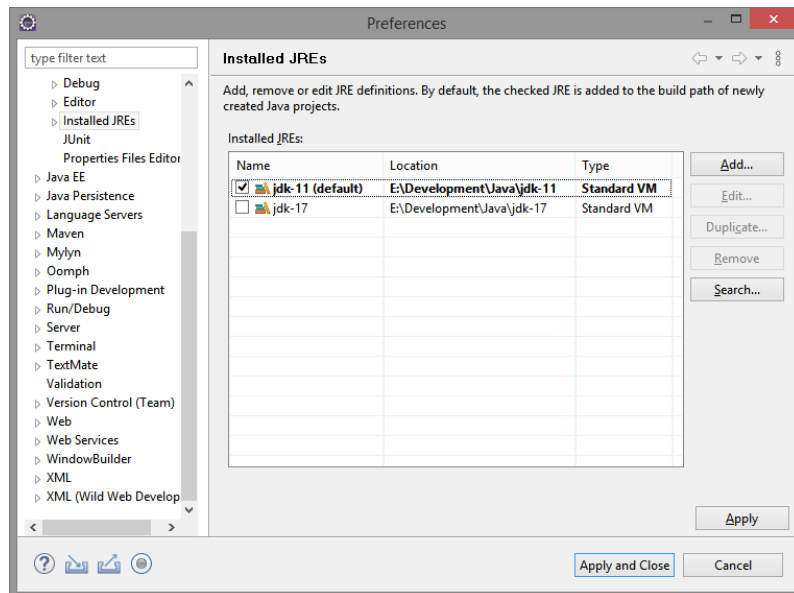
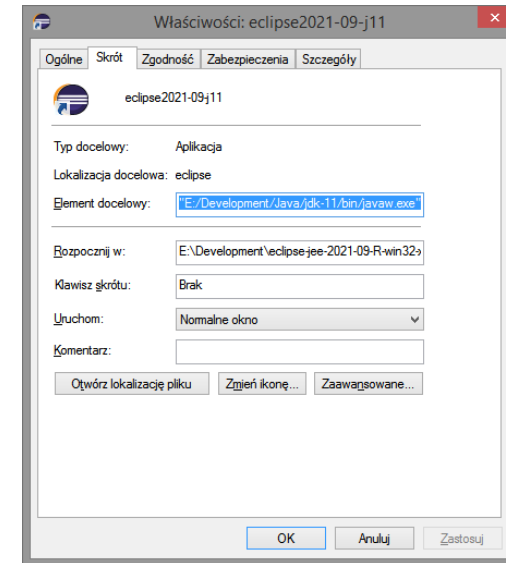
Eclipse IDE – Java

Środowisku można wskazać JRE, z jakim ma pracować:

- poprzez użycie opcji `-vm` (ustawiającej domyślne JRE) podczas uruchomienia `eclipse` z linii komend lub przez skrót:

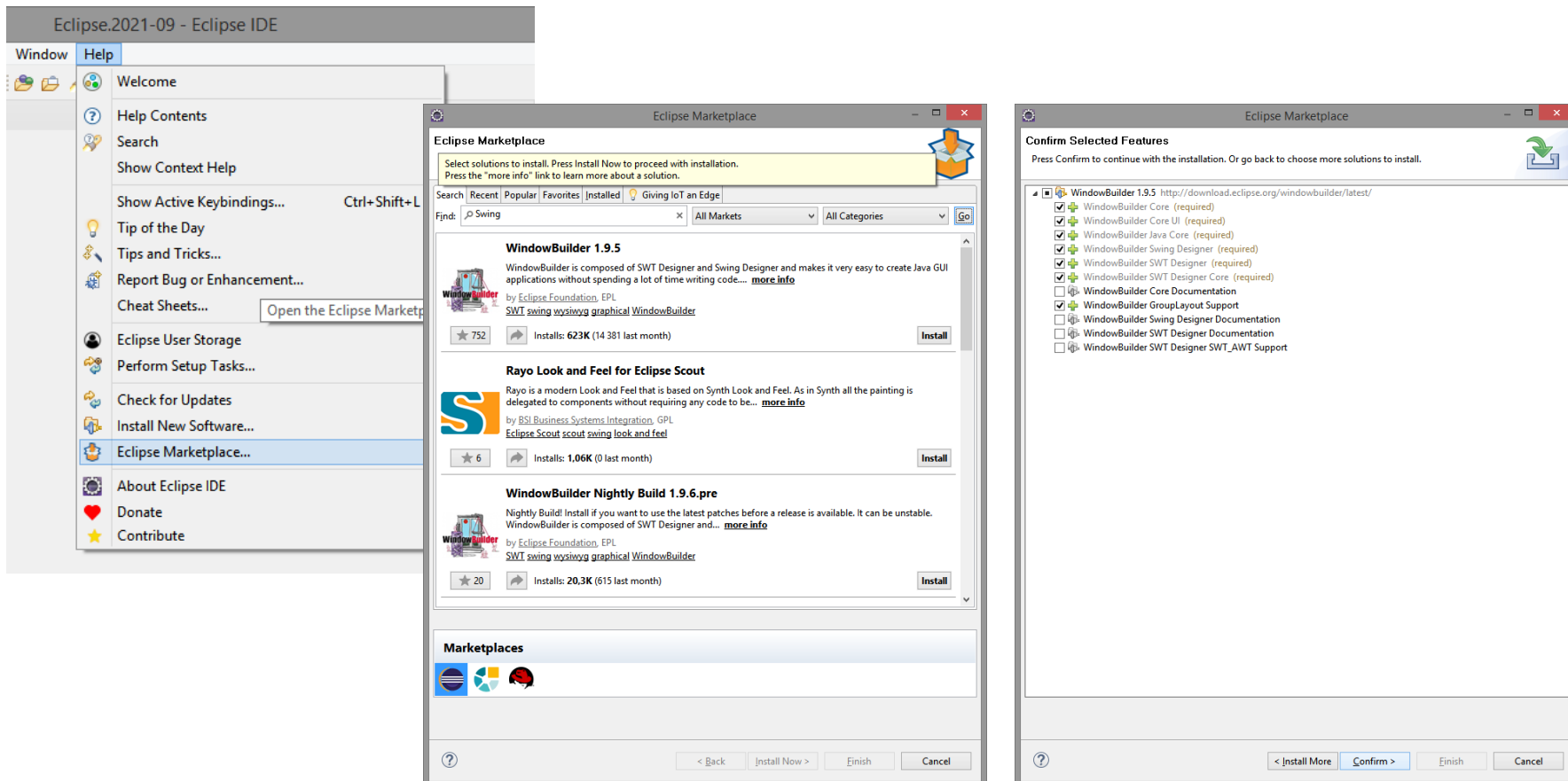
```
> eclipse.exe -vm "E:/Development/Java/jdk-11/bin/javaw.exe"
```

- poprzez ustawienie preferencji (trzeba wskazać typ JRE oraz ścieżkę)



Eclipse IDE – WindowBuilder

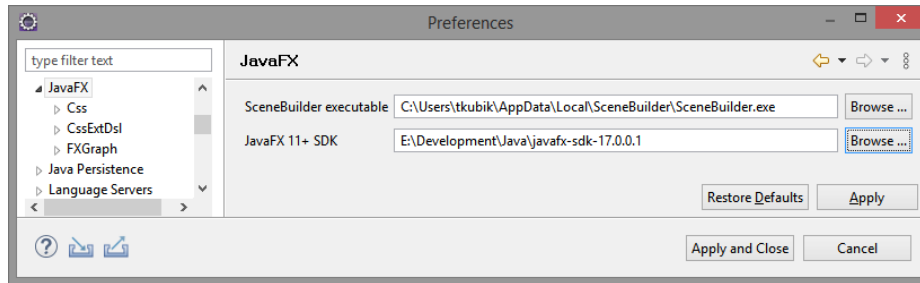
- Eclipse domyślnie nie oferuje narzędzia do projektowania GUI dla programów Java. Takie narzędzia należy sobie doinstalować.
- WindowBuider pozwala na projektowanie GUI za pomocą klas Swing, AWT, SWT



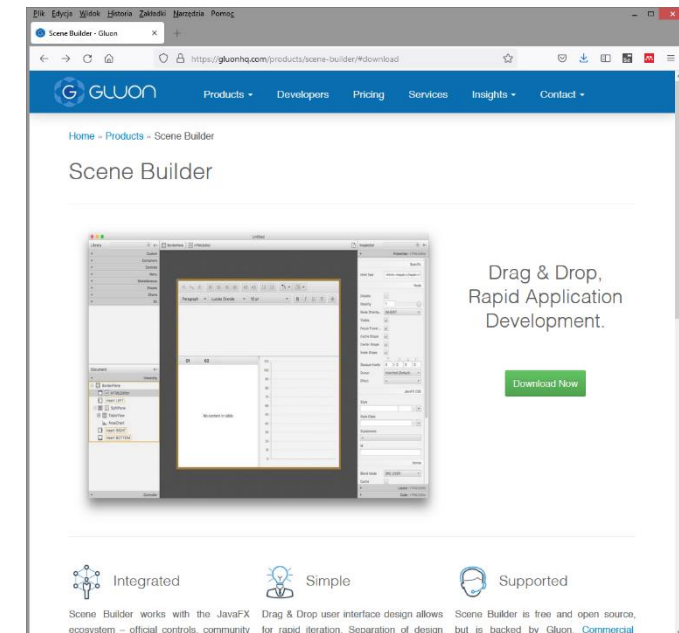
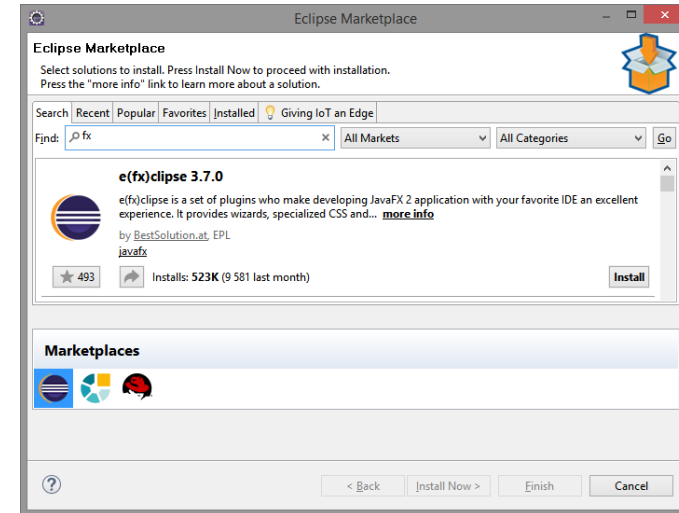
Eclipse IDE – e(fx)clipse

W środowisku można również uzyskać wsparcie projektowania aplikacji JavaFX, ale wymaga to

- obecności JavaFX na komputerze
 - do pobrania ze strony:
<https://gluonhq.com/products/javafx/>
- zainstalowania rozszerzenia e(fx)clipse z Marketplace
- zainstalowania SceneBuilder do projektowania GUI (wersja 17.0.0 działa z JDK11 i wyżej)
 - domyślnie instalowane jest w katalogu:
C:\Users\<userName>\AppData\Local\Scene Builder\
– ścieżkę tę należy ustawić w preferencjach eclipse



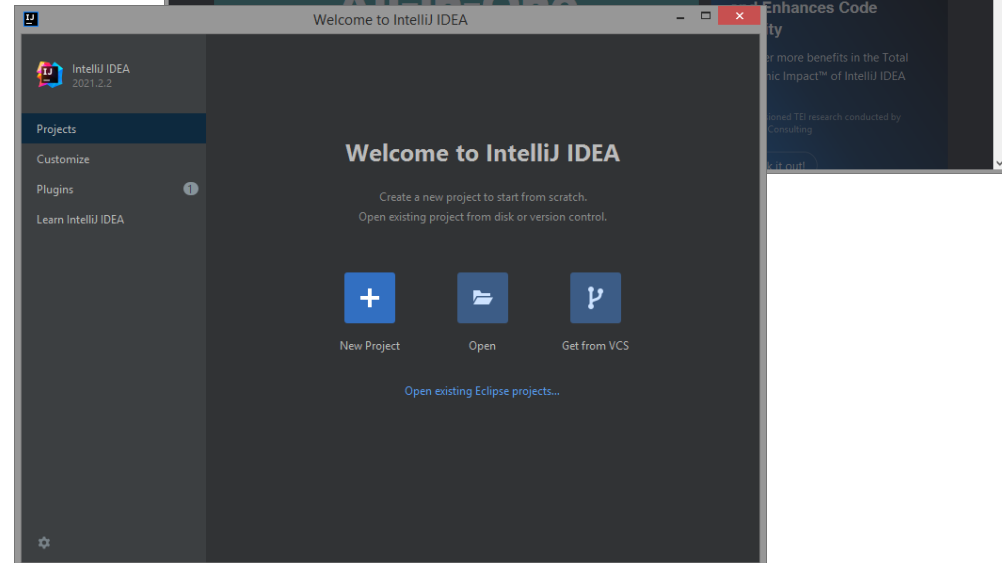
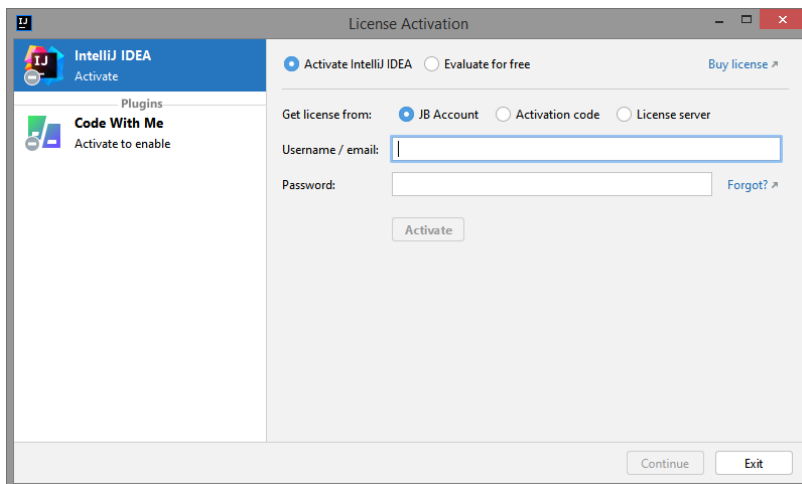
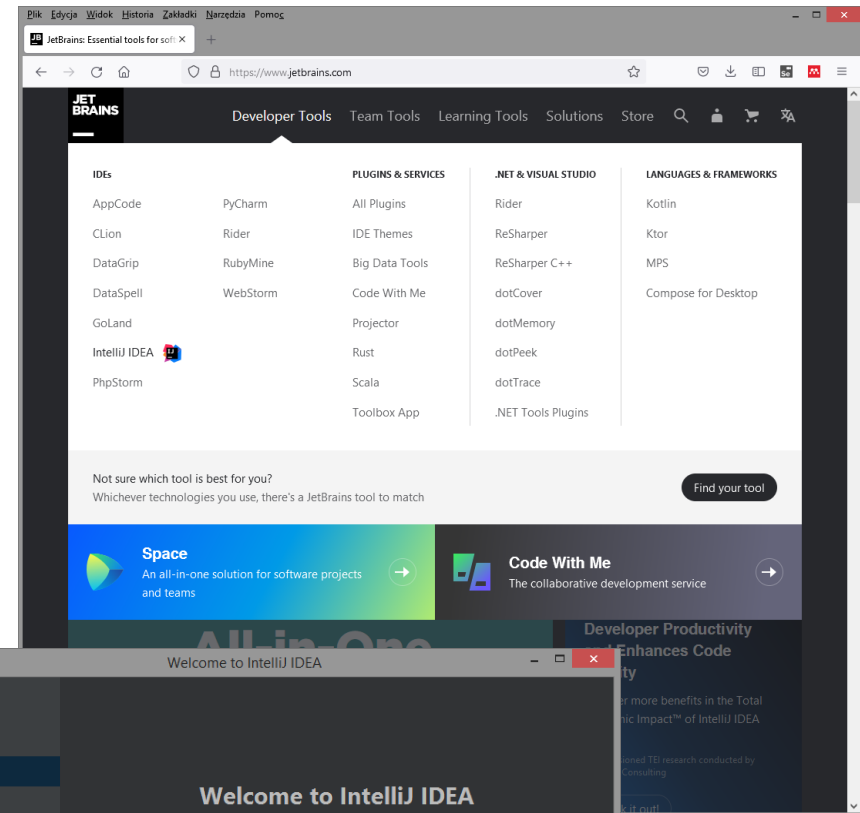
<https://gluonhq.com/products/scene-builder/#download>



IntelliJ IDEA

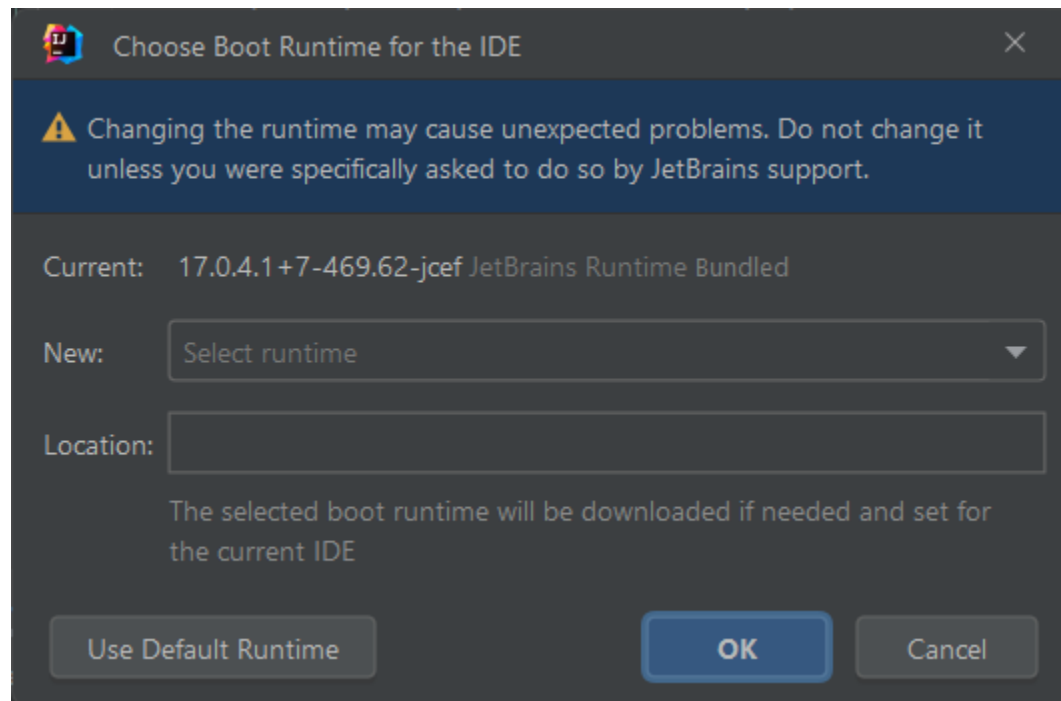
<https://www.jetbrains.com/idea/download/>

- Oprogramowanie w wersji
 - Ultimate (komercyjna, ale z dostępną licencją edukacyjną)
 - Community (darmowa)
- Pobrać można
 - instalator (exe) albo
 - archiwum (zip)
- Przy pierwszym uruchomieniu wersji Ultimate pojawi się
 - monit o aktywację licencji, a potem (po wpisaniu parametrów uwierzytelnienia użytkownika zarejestrowanego na stronie JetBrains)
 - okno powitalne



IntelliJ IDEA

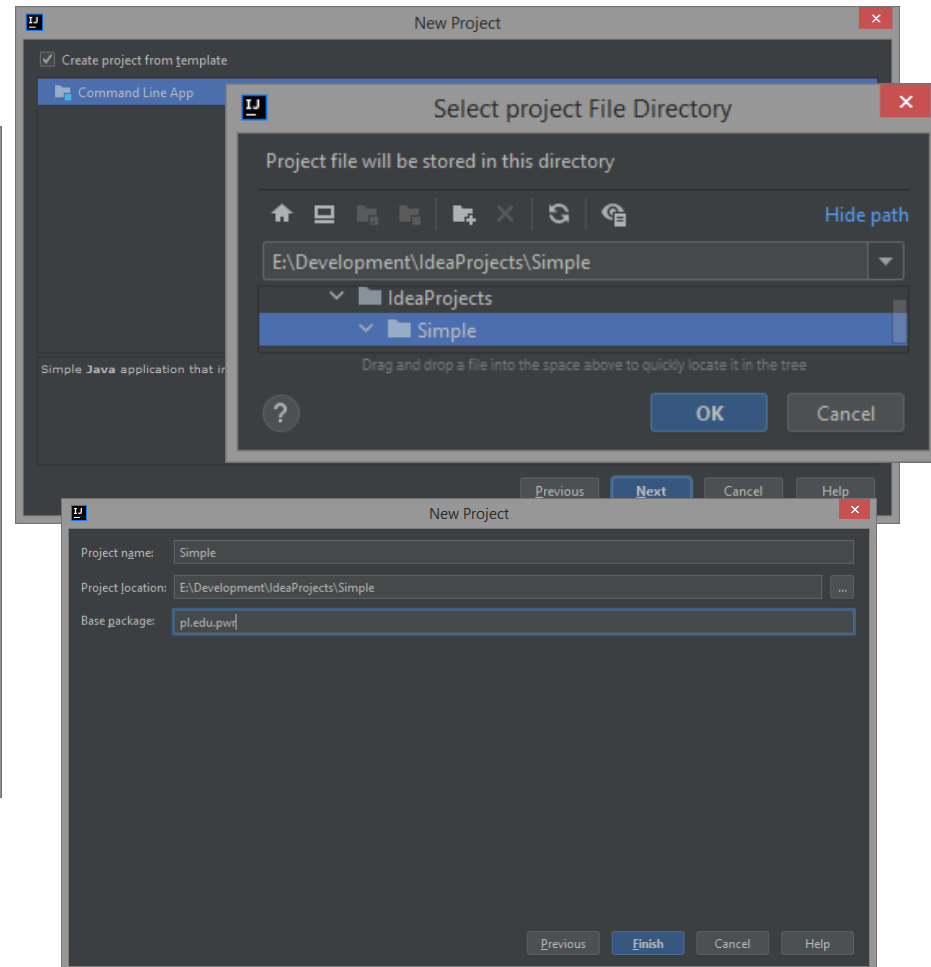
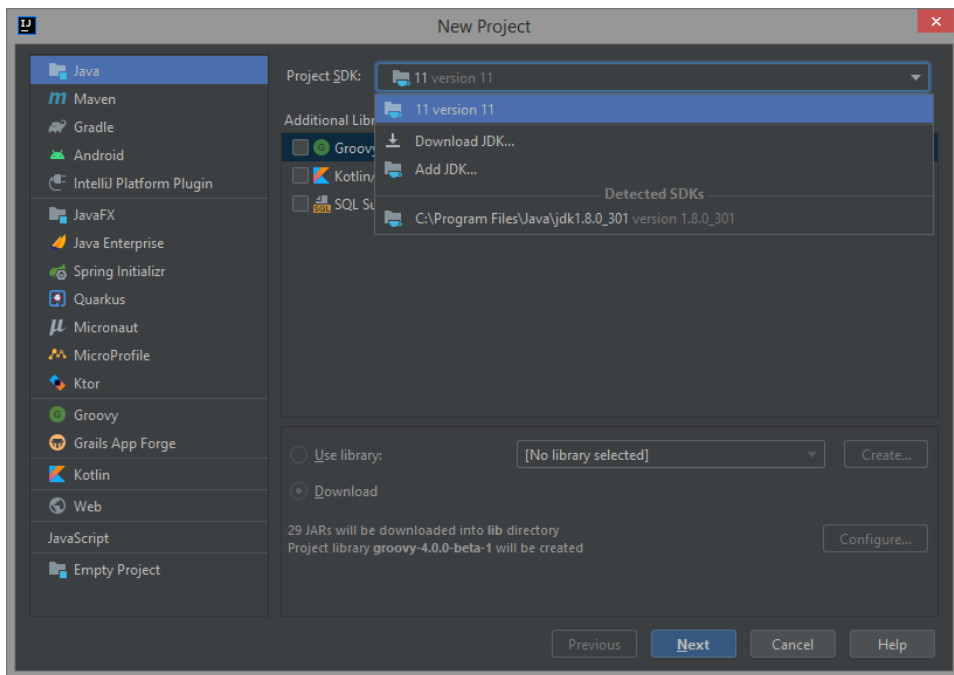
- Domyślnie środowisko IntelliJ IDEA 2022.2.3 (Ultimate Edition) działa w JRE 17.0.4.1+7-469.62-jcef
 - można się o tym przekonać próbując zmienić Boot Runtime



<https://www.jetbrains.com/help/idea/switching-boot-jdk.html>

IntelliJ IDEA - praca ze środowiskiem

- Podczas tworzenia projektów można zadeklarować, z którym JDK mają być one kompilowane i uruchamiane

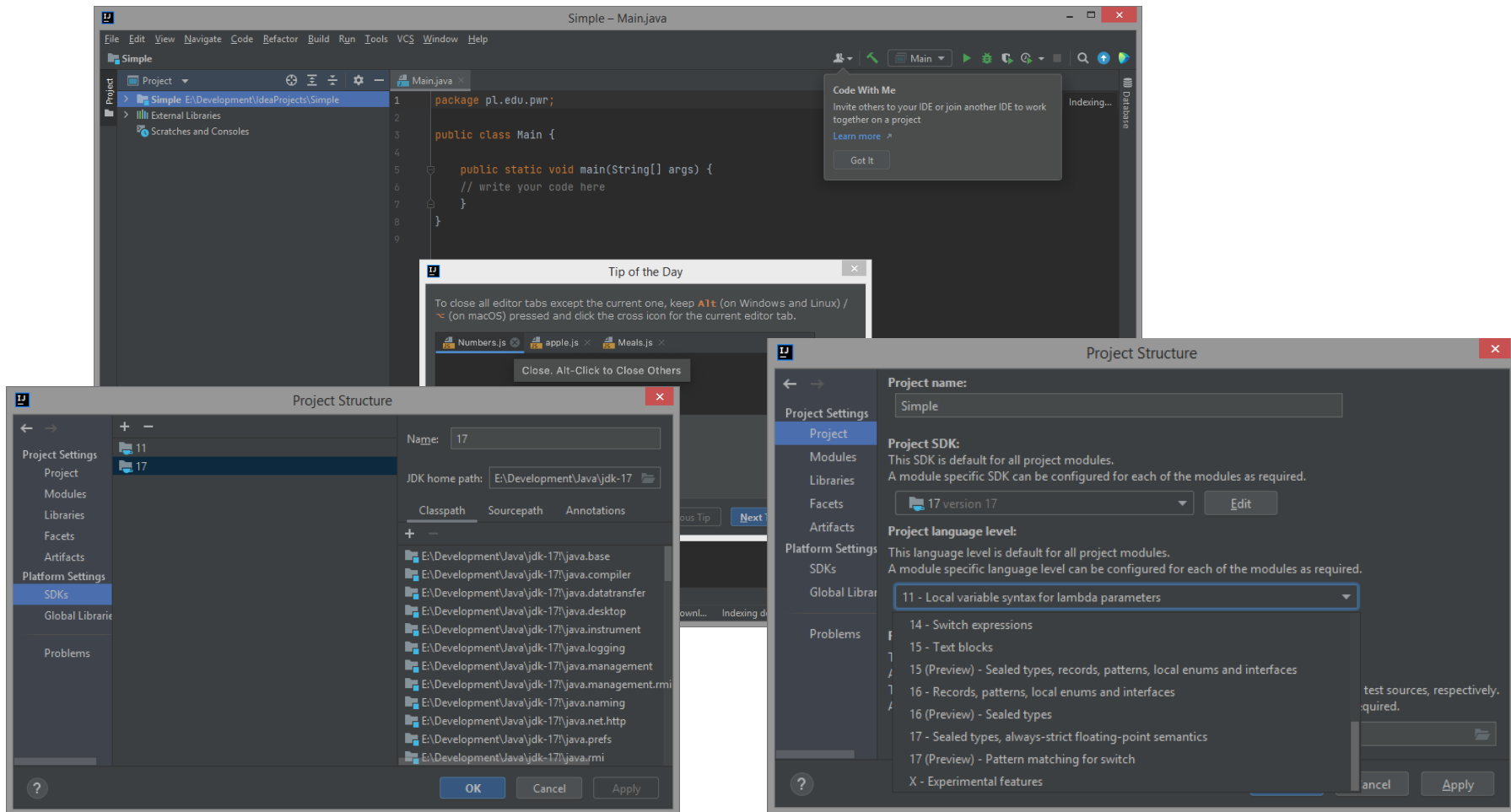


Do poczytania:

<https://blog.jetbrains.com/idea/2021/09/java-17-and-intellij-idea/>

IntelliJ IDEA – praca ze środowiskiem

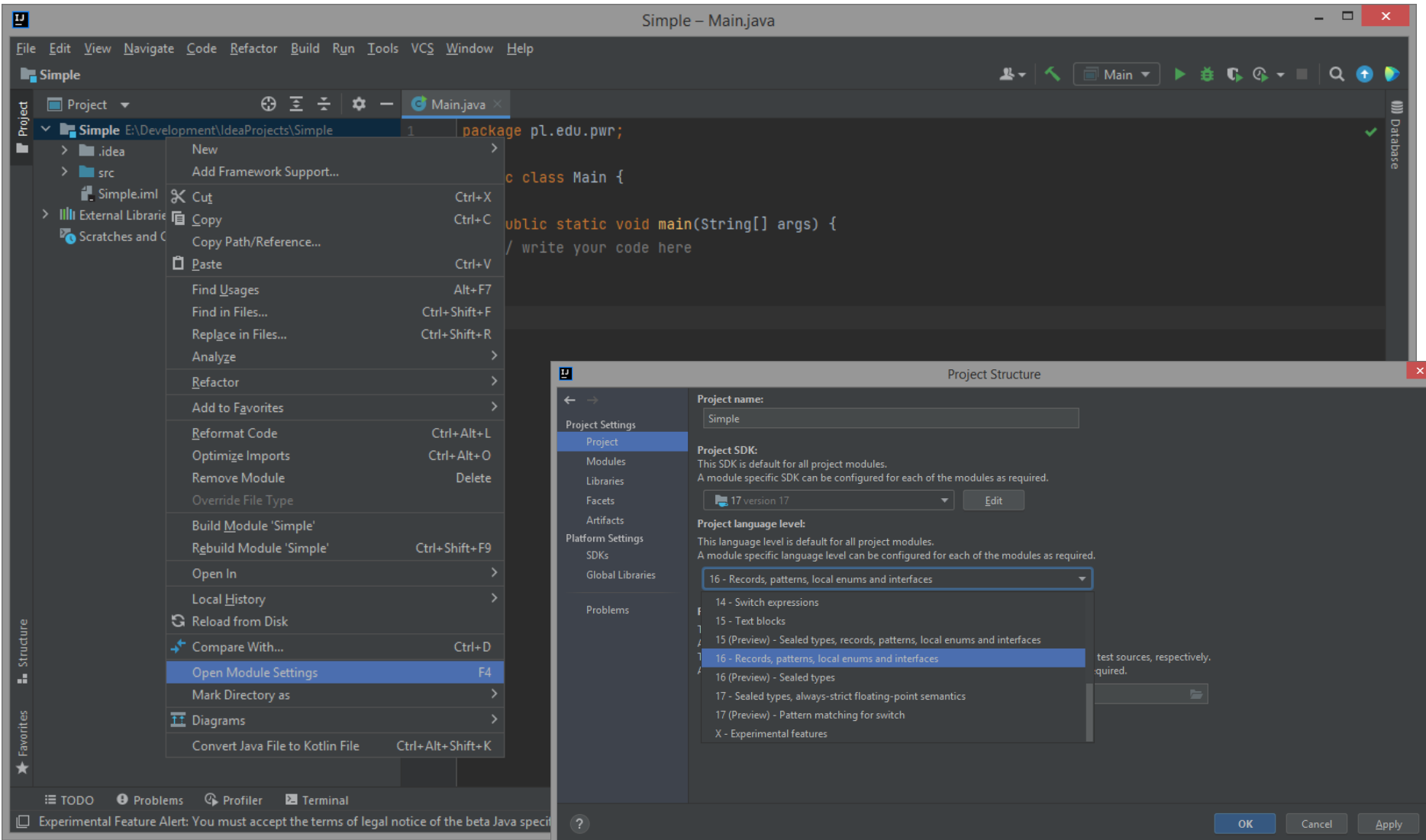
- Po uruchomieniu projektu można zmienić JDK w dowolnym momencie w ustawieniach Project Structure (Ctrl-Alt-Shift-S)



<https://mkyong.com/intellij/how-to-change-the-intellij-idea-jdk-version/>

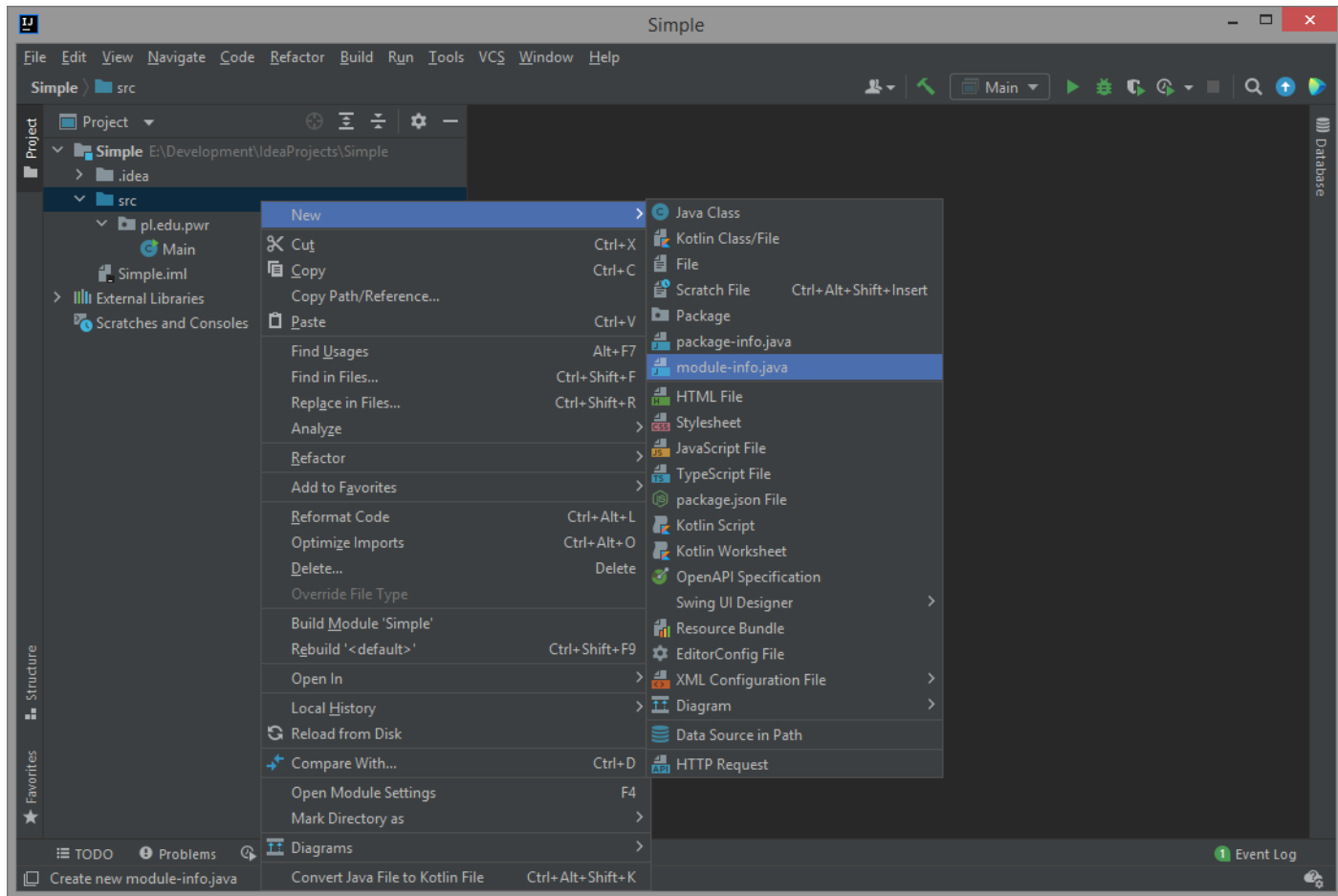
IntelliJ IDEA – praca ze środowiskiem

- Do ustawień można dojść również poprzez Open Module Settings



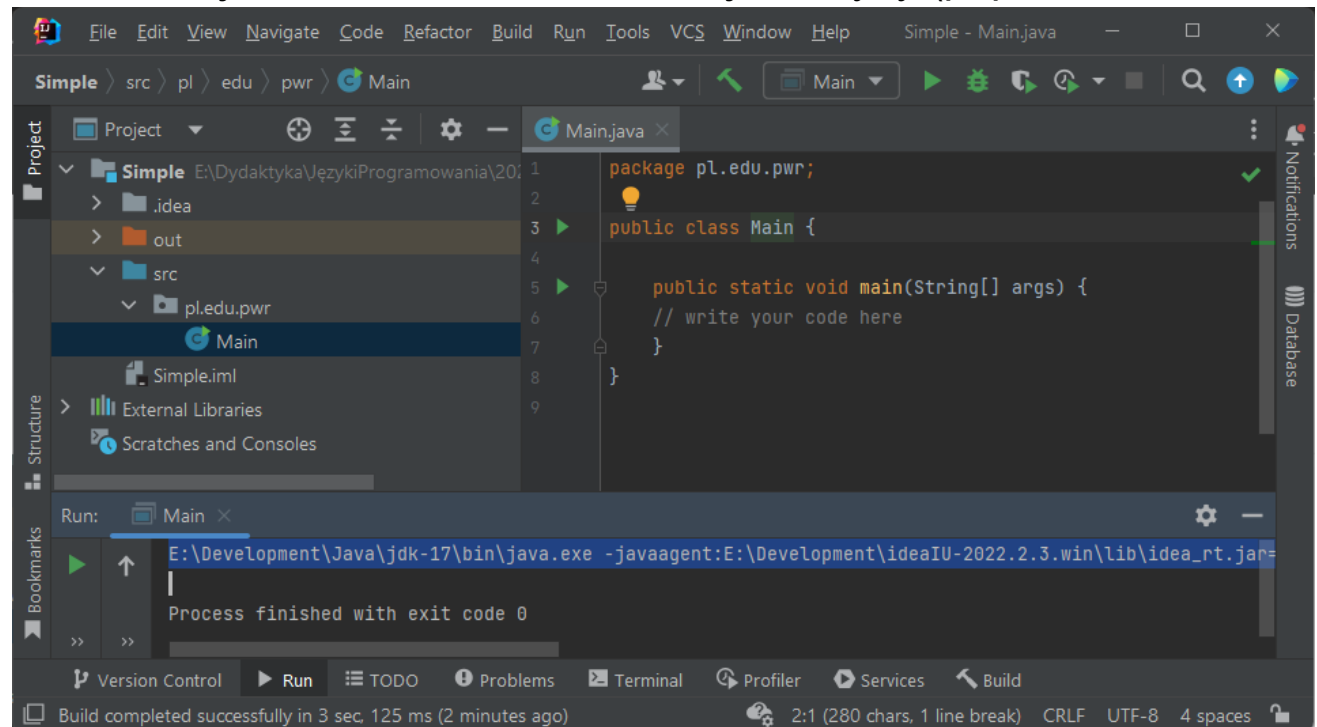
IntelliJ IDEA – praca ze środowiskiem

- Projekty Java domyślnie tworzone są bez `module-info.java`
- Plik ten jednak można w każdej chwili dodać



IntelliJ IDEA – praca ze środowiskiem

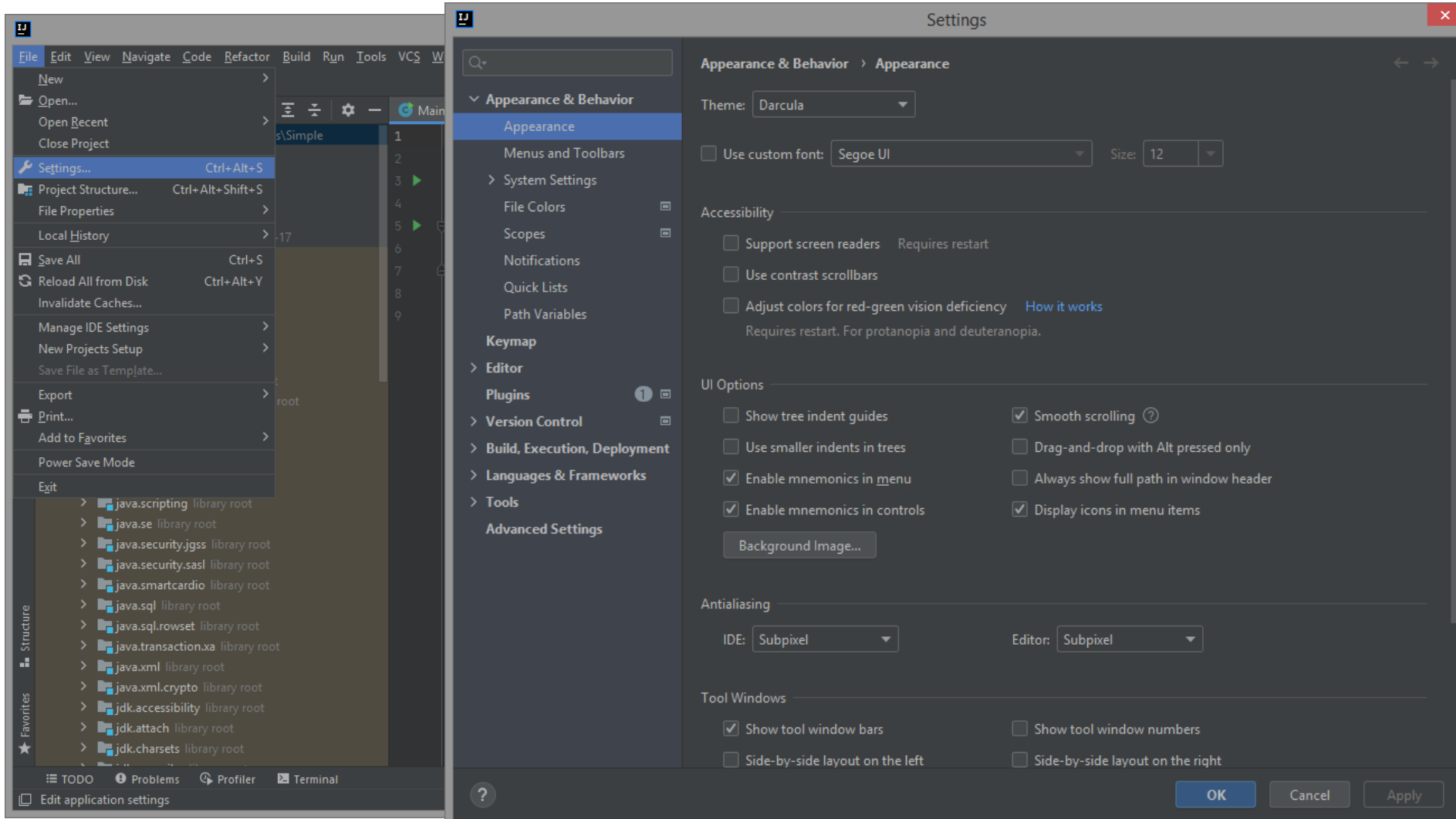
- Po uruchomieniu można zobaczyć, jaką komendę rzeczywiście odpaliło środowisko:
 - wersja z `module-info.java` (środowisko automatycznie ustawia ścieżkę modułową)
 - `E:\Development\Java\jdk-17\bin\java.exe -javaagent: E:\Development\idealU-2022.2.3.win\lib\idea_rt.jar=51005:E:\Development\idealU-2022.2.3.win\bin -Dfile.encoding=UTF-8 -p E:\Dydaktyka\JęzykiProgramowania\2022\projektyIntelliJ\Simple\out\production\Simple -m Simple/pl.edu.pwr.Main`
 - wersja projektu bez `module-info.java` (środowisko automatycznie ustawia ścieżkę klas)
 - `E:\Development\Java\jdk-17\bin\java.exe -javaagent: E:\Development\idealU-2022.2.3.win\lib\idea_rt.jar=51005:E:\Development\idealU-2022.2.3.win\bin -Dfile.encoding=UTF-8 -classpath E:\Dydaktyka\JęzykiProgramowania\2022\projektyIntelliJ\Simple\out\production\Simple pl.edu.pwr.Main`
- Widać, że w opcjach wirtualnej maszyny pojawił się `javaagent` – dzięki niemu środowisko ma możliwość zajrzenia do środka wirtualnej maszyny (poprzez instrumentację)



<https://www.baeldung.com/java-instrumentation>

IntelliJ IDEA – opcje interfejsu

- Środowisko pozwala dopasować wygląd interfejsu zgodnie z preferencjami



IntelliJ IDEA - JavaFX

- Projekt typu JavaFX to projekt mavenowy (konfiguracja w pliku `pom.xml`)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>pl.edu.pwr</groupId>
  <artifactId>SimpleFX</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>SimpleFX</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <junit.version>5.7.1</junit.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-controls</artifactId>
      <version>11</version>
    </dependency>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-fxml</artifactId>
      <version>11</version>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

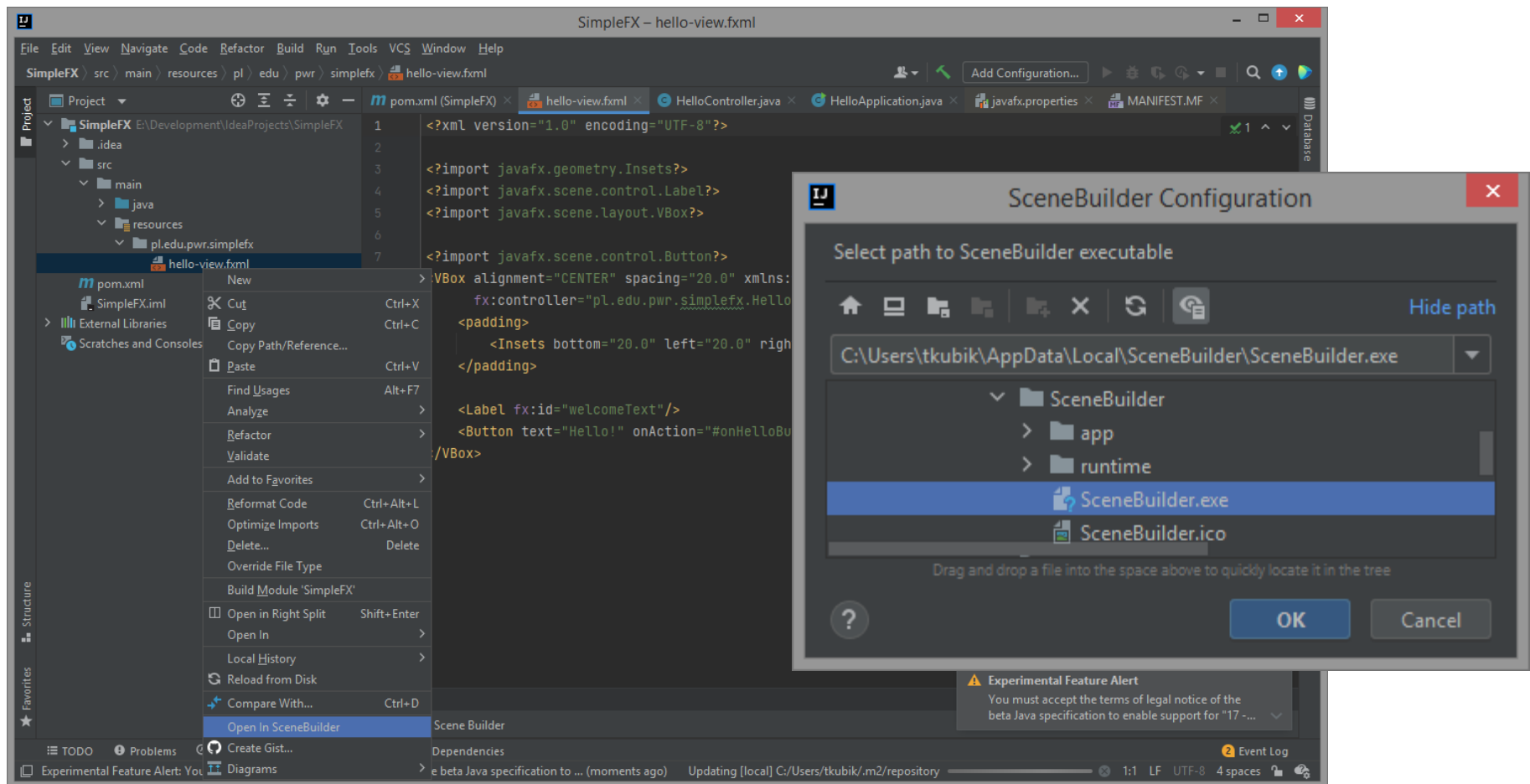
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>11</source>
          <target>11</target>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.openjfx</groupId>
        <artifactId>javafx-maven-plugin</artifactId>
        <version>0.0.6</version>
        <configuration>
          <stripDebug>true</stripDebug>
          <compress>2</compress>
          <noHeaderFiles>true</noHeaderFiles>
          <noManPages>true</noManPages>
          <jlinkImageName>simplefx</jlinkImageName>
          <launcher>launcher</launcher>
          <jlinkZipName>simplefxzip</jlinkZipName>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <mainClass>pl.edu.pwr.simplefx/pl.edu.pwr.simplefx.HelloApplication</mainClass>

  <configuration>
    <executions>
      </executions>
    </configuration>
  </project>
```


IntelliJ IDEA - JavaFX

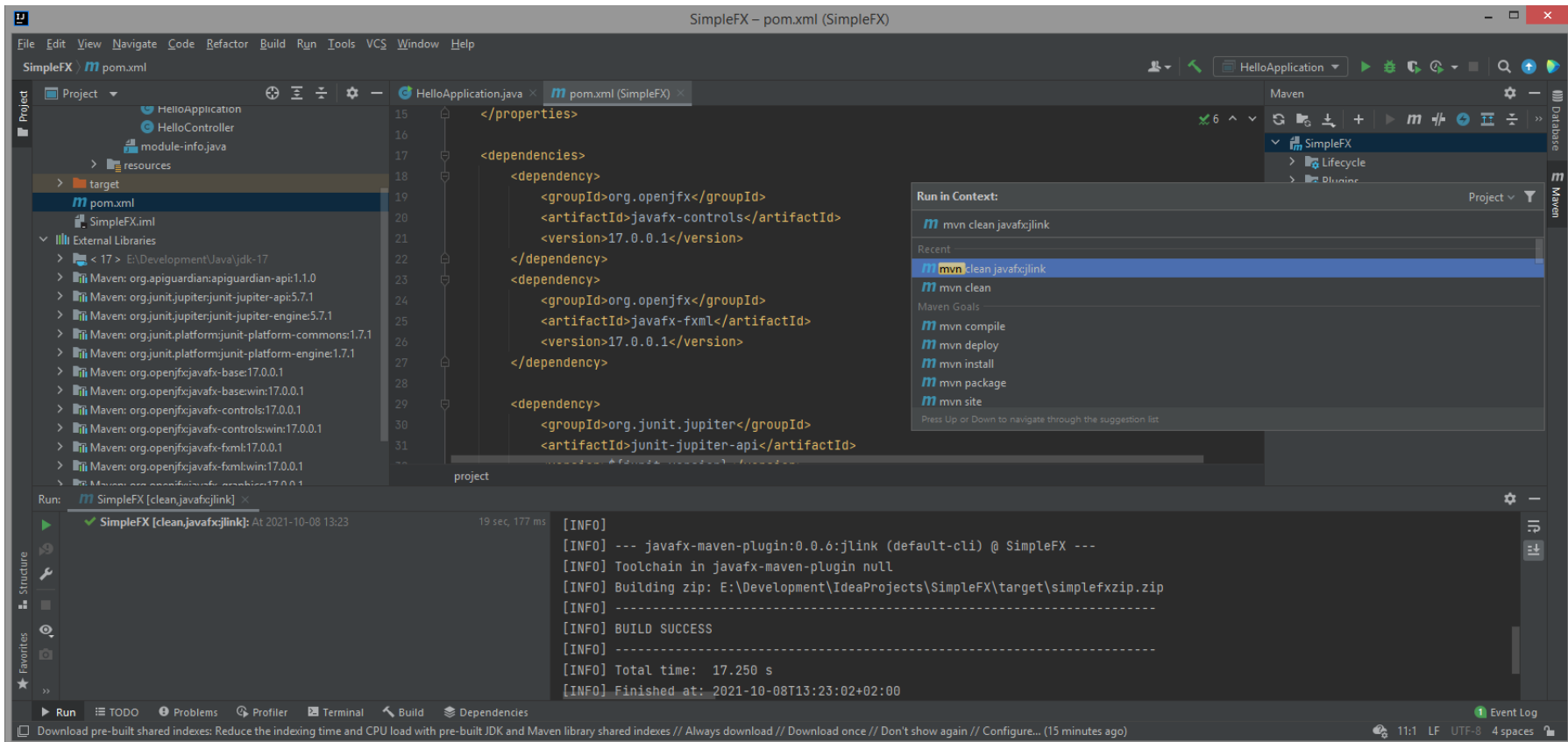
- Aby dało się uruchomić SceneBuilder należy ustawić odpowiednią ścieżkę (SceneBuilder dostępny jest też jako plugin)



<https://www.jetbrains.com/help/idea/javafx.html#create-project>

IntelliJ IDEA – JavaFX

- Pakowanie aplikacji z użyciem plugina mavenowego i jlink działa dla
 - JavaFX 17.0.0.1
 - JavaFX 11

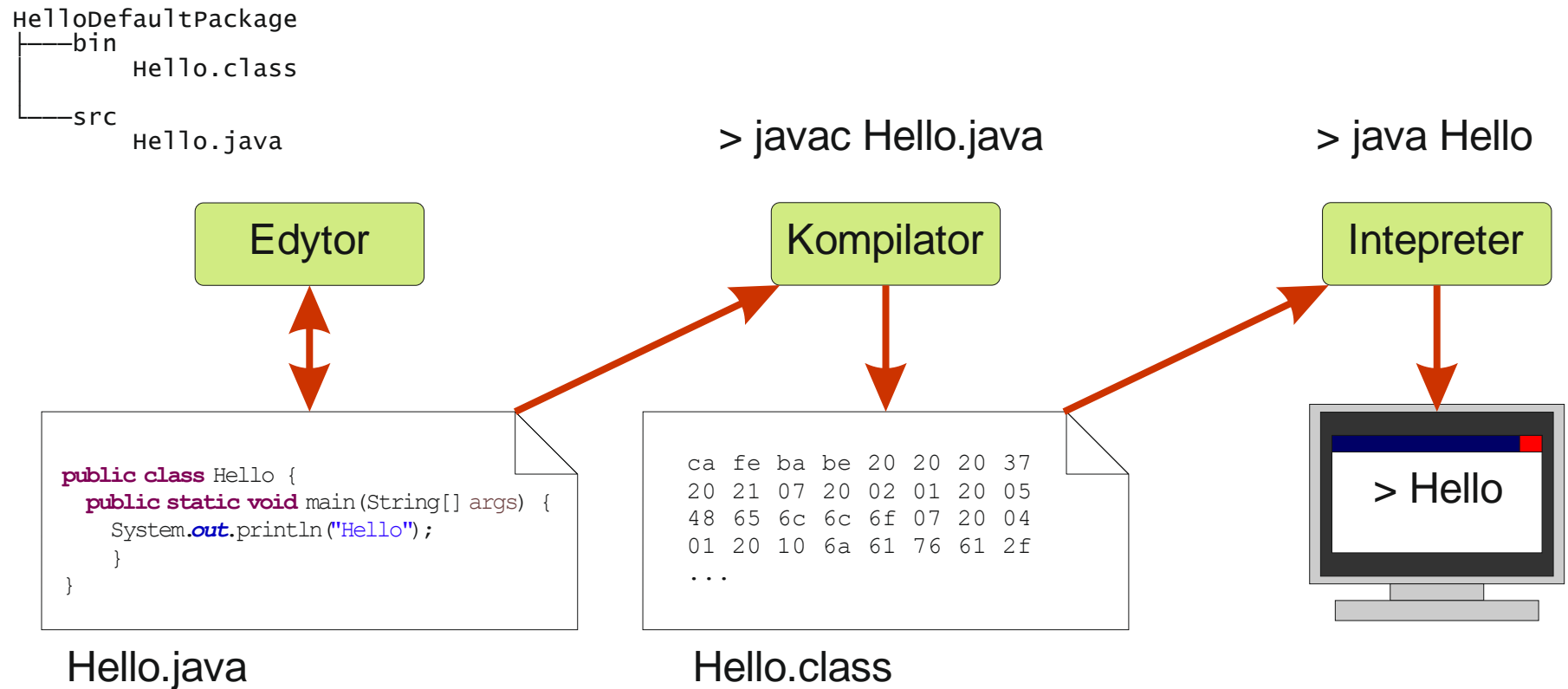


<https://www.jetbrains.com/help/idea/packaging-javafx-applications.html>

<https://openjfx.io/openjfx-docs/#modular>

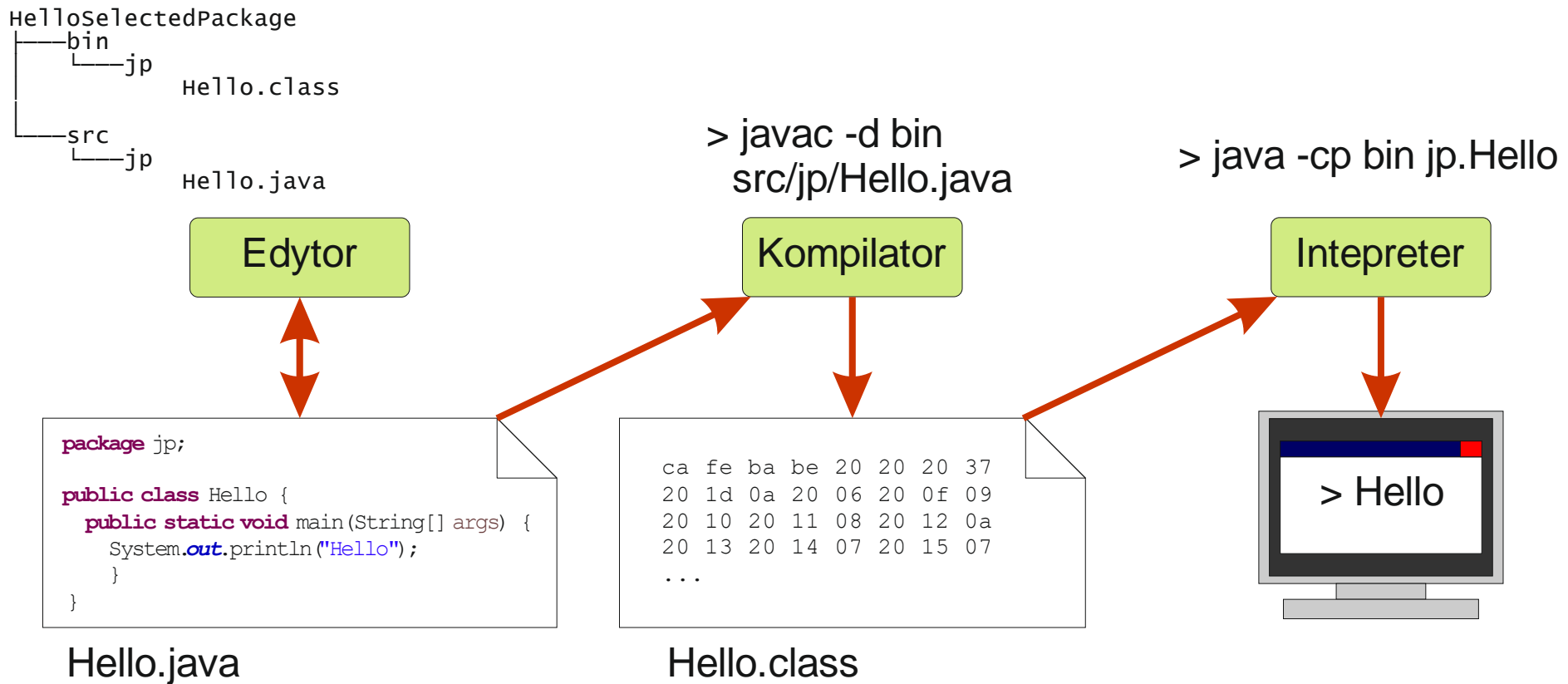
Wytwarzanie oprogramowania w języku JAVA

- podejście standardowe (klasa w pakiecie domyślnym)



Wytwarzanie oprogramowania w języku JAVA

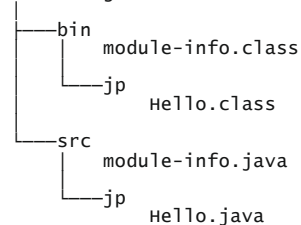
- podejście standardowe (klasa w pakiecie zdefiniowanym)



Wytwarzanie oprogramowania w języku JAVA

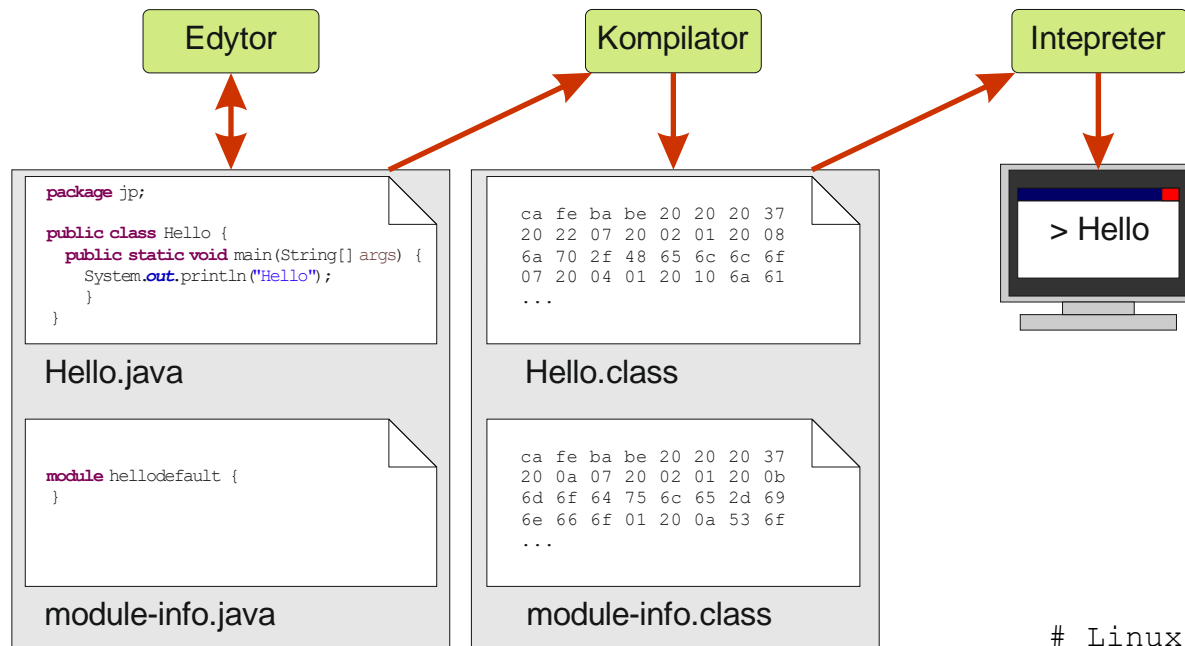
- podejście z jednym, domyślnym modułem

HelloSingleJPMS:



```
> javac -d bin
src\jp\Hello.java
src\module-info.java
```

```
> java --module-path bin
--module hellodefautl/jp.Hello
```

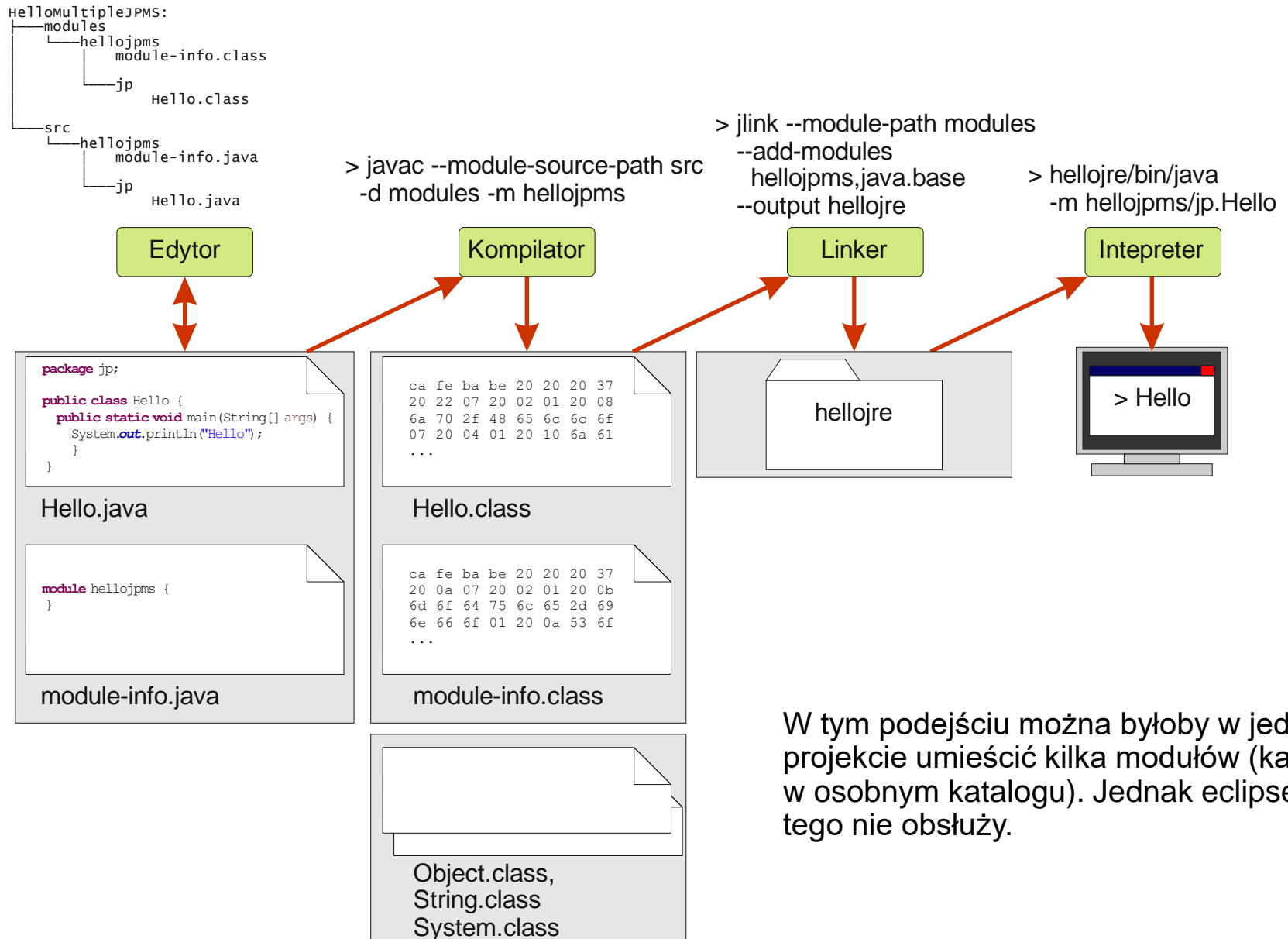


```
# Linux / MacOS
$ find -name "*.java" > sources.txt
$ javac @sources.txt
```

```
:: Windows
> dir /s /B *.java > sources.txt
> javac @sources.txt
```

Wytwarzanie oprogramowania w języku JAVA

- podejście z osobnym katalogiem dla modułu



Pliki metadanych

- module-info.java
 - zawiera informacje o zależnościach modułowych, może zawierać dokumentację modułu

```
module SimpleFX {  
    requires javafx.controls;  
    requires javafx.fxml;  
  
    opens application to javafx.graphics, javafx.fxml;  
}
```

```
{@author}, {@deprecated}, {@provides},  
{@see}, {@since}, {@serialField}, {@uses},  
{@version} {@code}, {@docRoot}, {@index},  
{@link}, {@linkplain}, {@literal}, {@summary}
```

<https://stackoverflow.com/questions/53957486/javadoc-comments-in-module-info>

- package-info.java
 - zwykle zawiera dokumentację pakietu oraz adnotacje wykorzystywane przez różne frameworki,

```
/**  
 * @author tkubik  
 *  
 */  
package application;
```

```
@NonNullApi  
@NonNullFields  
package com.baeldung.nullibility;  
  
import org.springframework.lang.NonNullApi;  
import org.springframework.lang.NonNullFields;
```

<https://www.baeldung.com/java-package-info>

Przydatne linki

- Tworzenie i kompilacja projektów modułowych
 - <https://www.logicbig.com/tutorials/core-java-tutorial/modules/modes.html>
 - <https://www.baeldung.com/java-9-modularity>
 - <https://www.developer.com/java/ent/using-a-java-9-module-as-a-jar-file.html>
 - <https://www.geeksforgeeks.org/jlink-java-linker/>
- Java + JavaFX modularnie
 - <https://openjfx.io/openjfx-docs/modular>
- Uruchamianie projektów jednoplikowych poleceniem `java` bez konieczności jawnej kompilacji
 - <https://www.infoq.com/articles/single-file-execution-java11/>
- Specyfikacja Java
 - <https://docs.oracle.com/javase/specs/>
- Specyfikacja linkera
 - <https://openjdk.java.net/jeps/282>