

Informacioni sistemi

Računske vežbe:

Java EE

Doc. dr Valentina Nejkoć

valentina@elfak.ni.ac.rs

kancelarija 524

**Katedra za Računarstvo,
Elektronski fakultet, Univerzitet u Nišu**

- **Java Platforme su:**

- ❖ Java Platform, Standard Edition (Java SE)
- ❖ Java Platform, Enterprise Edition (Java EE)
- ❖ Java Platform, Micro Edition (Java ME)
- ❖ JavaFX

- **Sve Java platforme se sastoje iz:**
 - ❖ **Java Virtual Machine (JVM)**
 - JVM je program za odgovarajući hardver i softver koji može da izvršava aplikacije razvijene pomoću Java tehnologije.
 - ❖ **Application Programming Interface (API)**
 - API je kolekcija softverskih komponenti koje se mogu koristiti za kreiranje drugih softverskih komponenti ili aplikacija.

- **Svaka Java platforma obezbeđuje JVM i API, što omogućuje aplikacijama pisanim za tu platformu da se izvršavaju na bilo kom kompatibilnom sistemu sa osobinama Java programskog jezika. Aplikacije su tada:**
 - ❖ **Nezavisne je od platforme**
 - ❖ **Stabilne**

- **Java SE**

- ❖ Java SE API obezbeđuje osnovnu funkcionalnost Java programskog jezika. Moguće je definisanje počev od osnovnih tipova i objekata do klasa visokog nivoa koje se koriste za mreže, bezbednost, pristup bazama podataka, za razvoj grafičkog korisničkog interfejsa (GUI) i XML parsiranje.
- ❖ Sastoji se od VM, alata za razvoj, tehnologija za deployment i drugih biblioteka i alata koji se najčešće koriste kod aplikacija zasnovanih na Java tehnologiji.

- **Java EE**

- ❖ Java EE platforma je iznad Java SE platforme i obezbeđuje API i runtime okruženje za razvoj velikih, višeslojnih, skalabilnih, pouzdanih i bezbednih aplikacija.

- **Java ME**

- ❖ Java ME platform obezbeđuje API i small-footprint VM za izvršavanje Java programskog jezika na malim uređajima, npr. mobilnim telefonima. Sadrži specijalne biblioteke korisne za razvoj aplikacija na malim uređajima. Java ME aplikacije su često klijenti za servise Java EE platforme.

- **JavaFX**

- ❖ JavaFX je platforma za kreiranje bogatih internet aplikacija upotrebom lightweight API korisničkog interfejsa.
- ❖ JavaFX aplikacije su aplikacije modernog izgleda koje imaju API visokog nivoa koji obezbeđuje konektovanje na mrežne izvore podataka. Ove aplikacije su namenjene klijentima visokih performansi i mogu biti klijenti servisima Java EE platforme.

Pregled Enterprise aplikacija

- **Java EE platforma je dizajnirana da pomogne developer-ima da kreiraju velike, bezbedne, skalabilne, pouzdane i višeslojne aplikacije.**
- **Ove aplikacije su dizajnirane da reše probleme u okviru velikih preduzeća i organizacija.**
- **Enterprise aplikacije nisu samo korisne za velike korporacije, agencije i vlade, već se mogu primeniti i za male organizacije.**

Pregled Enterprise aplikacija

- **Java EE platforma je dizajnirana da redukuje kompleksnost razvoja enterprise aplikacije obezbeđujući:**
 - ❖ Model razvoja
 - ❖ API
 - ❖ Izvršno okruženje koje omogućuje developer–ima da akcenat stave na funkcionalnosti.

Slojevi aplikacije

- **U višeslojnim aplikacijama funkcionalnost aplikacije je podeljena na izolovane funkcionalne delove, koji se zovu slojevi.**
- **Uglavnom, višeslojna aplikacija ima:**
 - ❖ Klijentski sloj
 - ❖ Srednji sloj
 - ❖ Sloj podataka (sloj enterprise informacionog sistema)

Slojevi aplikacije

- **Klijentski sloj** sastoji se od klijentskih programa koji šalju zahteve srednjem sloju.
- U srednjem sloju nalaze se poslovne funkcije koje opslužuju klijentske zahteve i procesiraju podatke aplikacije skladišteći ih u **datastore** (repozitorijumu za podatke) u sloju podataka.
- Razvoj Java EE aplikacije stavlja akcenat na razvoj srednjeg sloja sa ciljem kreiranja menadžmenta enterprise aplikacije lakšim, robusnijim i bezbednijim.

Slojevi aplikacije: Klijentski sloj

- Klijentski sloj se sastoji od klijenata koji pristupaju Java EE server i koji su uglavnom locirani na različitoj mašini od servera. Klijent šalje zahteve serveru, koji procesira njegove zahteve i prosleđuju mu nazad odgovor.
- Različiti tipovi aplikacija mogu biti klijenti. Mogu biti Java EE klijenti, ali i neke aplikacije koje nisu razvijene Java tehnologijom.
- Klijenti mogu da budu web čitači, samostalne aplikacije ili drugi serveri, koji se izvršavaju na različitim mašinama od Java EE servera.

Slojevi aplikacije: Web sloj

- **Web sloj se sastoji iz komponenti koje su zadužene za interakciju između klijenata i poslovnog sloja. Zadaci ovog sloja su:**
 - ❖ Dinamičko generisanje sadržaja različitih formata za klijente
 - ❖ Prikupljanje ulaza od klijenata (preko korisničkog interfejsa klijenata) i vraćanje odgovarajućih rezultata komponenti iz poslovnog sloja
 - ❖ Kontrolise tok ekrana ili strana na klijentu
 - ❖ Održava stanje podataka za sesije korisnika
 - ❖ Izvodi osnovnu logiku i drži neke podatke privremeno.

Slojevi aplikacije: Web sloj

- **Java EE tehnologije web sloja**

Technology	Purpose
JavaServer Faces technology	A user-interface component framework for web applications that allows you to include UI components (such as fields and buttons) on a XHTML page, called a Facelets page; convert and validate UI component data; save UI component data to server-side data stores; and maintain component state
Expression Language	A set of standard tags used in Facelets pages to refer to Java EE components
Servlets	Java programming language classes that dynamically process requests and construct responses, usually for HTML pages
Contexts and Dependency Injection for Java EE	A set of contextual services that make it easy for developers to use enterprise beans along with JavaServer Faces technology in web applications

Slojevi aplikacije: Poslovni sloj

- Poslovni sloj se sastoji od komponenti koje obezbeđuju **poslovnu logiku za aplikaciju**. Poslovna logika je kod koji obezbeđuje funkcionalnost odgovarajućem poslovnom domenu.
- U dobro dizajniranoj enterprise aplikaciji, **osnovne funkcionalnosti sistema nalaze se u komponentama poslovnog sloja**.
- Sledeće Java EE tehnologije se koriste u poslovnom sloju: **Enterprise JavaBeans** (enterprise bean) komponente, **JAX-RS RESTful web servisi**, **Java Persistence API** entiteti

Slojevi aplikacije: Poslovni sloj

Technology	Description
Enterprise JavaBeans (enterprise bean) components	Enterprise beans are managed components that encapsulate the core functionality of an application.
JAX-RS RESTful web services	An API for creating web services that respond to HTTP methods (for example GET or POST methods). JAX-RS web services are developed according to the principles of REST, or representational state transfer.
JAX-WS web service endpoints	An API for creating and consuming SOAP web services.
Java Persistence API entities	An API for accessing data in underlying data stores and mapping that data to Java programming language objects.
Java EE managed beans	Managed components that may provide the business logic of an application, but do not require the transactional or security features of enterprise beans.

Slojevi aplikacije: Sloj enterprise informacionog sistema (EIS)tj. sloj podataka



- **EIS sloj se sastoji iz:**
 - ❖ Servera baze podataka
 - ❖ Enterprise Sistema za planiranje resursa
 - ❖ Drugih izvora podataka
- **Resursi su obično locirani na odvojenoj mašini od Java EE servera i pristupaju im komponente poslovnog sloja.**

Slojevi aplikacije: Sloj enterprise informacionog sistema (EIS)tj. sloj podataka



- **Sledeće Java EE tehnologije se koriste za pristup EIS sloju u Java EE aplikacijama:**
 - ❖ The Java Database Connectivity API (JDBC)
 - ❖ The Java Persistence API
 - ❖ The Java EE Connector Architecture
 - ❖ The Java Transaction API (JTA)

Slojevi aplikacije: Sloj enterprise informacionog sistema (EIS)tj. sloj podataka

Technology	Description
The Java Database Connectivity API (JDBC)	A low-level API for accessing and retrieving data from underlying data stores. A common use of JDBC is to make SQL queries on a particular database.
The Java Persistence API	An API for accessing data in underlying data stores and mapping that data to Java programming language objects. The Java Persistence API is a much higher-level API than JDBC, and hides the complexity of JDBC from the user.
The Java EE Connector Architecture	An API for connecting to other enterprise resources, like enterprise resource planning or customer management system software.
The Java Transaction API (JTA)	An API for defining and managing transactions, including distributed transactions or transactions that cross multiple underlying data sources.

Java EE server



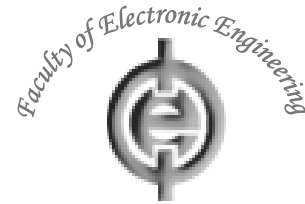
- **Java EE server je serverska aplikacija koja implementira API Java EE platforme i obezbeđuje standardne Java EE servise.**
- **Java EE server se zove i aplikacioni server, jer obezbeđuju aplikacione podatke klijentima.**
- **Java EE server hostuje nekoliko tipova aplikacionih komponenti koje odgovaraju slojevima u višeslojnoj aplikaciji. Java EE server obezbeđuje servise tim komponentama u formi kontejnera.**

Java EE kontejneri



- **Java EE kontejneri predstavljaju interfejs između komponente i funkcionalnosti koju obezbeđuje platforma da bi podržala tu komponentu.**
- **Funkcionalnost kontejnera je definisana platformom i različita je za svaki tip komponente.**
- **Server dozvoljava da različiti tipovi komponenti rade zajedno da bi obezbedili funkcionalnost u nekoj enterprise aplikaciji.**
- **Postoje: Web , EJB i aplikacioni klijentski kontejneri.**

Java EE kontejneri: Web kontejner



- **Web kontejner je interfejs između web komponenti i web servera.**
- **Web komponenta može da bude servlet ili JavaServer Faces Facelets (JSF) strana.**
- **Kontejner upravlja životnim ciklusom komponenti, prosleđuje zahteve aplikacionim komponentama i obezbeđuje interfejse kontekstnim podacima (npr. Informacija o trenutnom zahtevu).**

Java EE kontejneri: EJB kontejner



- **EJB kontejner je interfejs između enterprise bean-ova, koji obezbeđuju poslovnu logiku u Java EE aplikacijama, i Java EE servera.**
- **EJB kontejner se izvršava na Java EE serveru i upravlja izvršenjem enterprise bean-ova aplikacije.**

Java EE kontejneri: Aplikacioni klijentski kontejner



- **Aplikacioni klijentski kontejner je interfejs između Java EE aplikacionih klijenata (posebno Java SE aplikacija koje koriste komponente Java EE servera) i Java EE servera.**
- **Aplikacioni klijentski kontejner se izvršava na klijentskoj mašini i predstavlja gateway između klijentske aplikacije i komponenti Java EE servera koje klijent koristi.**

Šta ćemo mi da koristimo?

● Poslovni sloj

- ❖ Java EE server (JBoss, WildFly, GlassFish)
- ❖ EJB kontejner
- ❖ Tehnologija: Enterprise JavaBeans

● Sloj podataka

- ❖ Server baze podataka → tehnologija: The Java Database Connectivity API (JDBC)

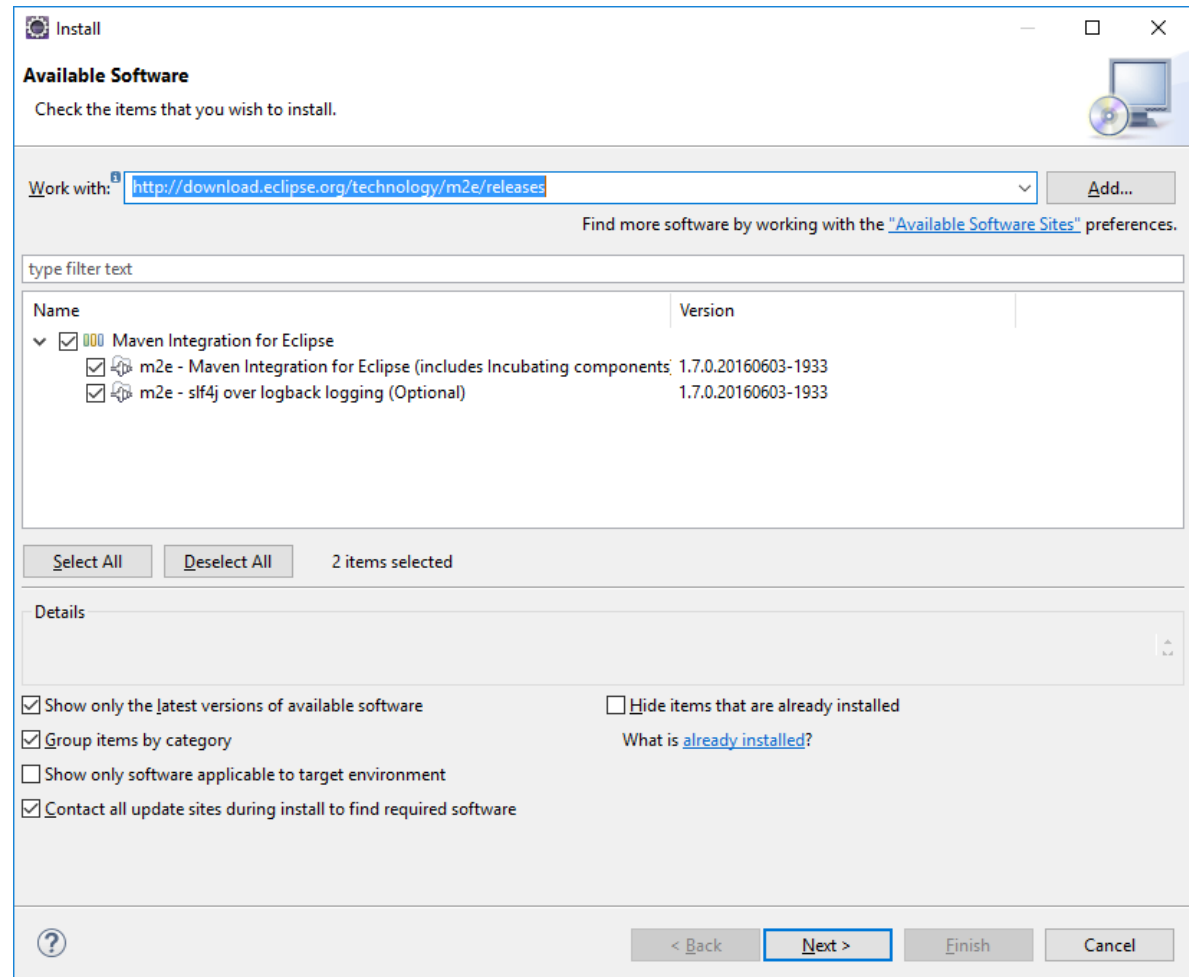
● Maven

● Zašto Maven?

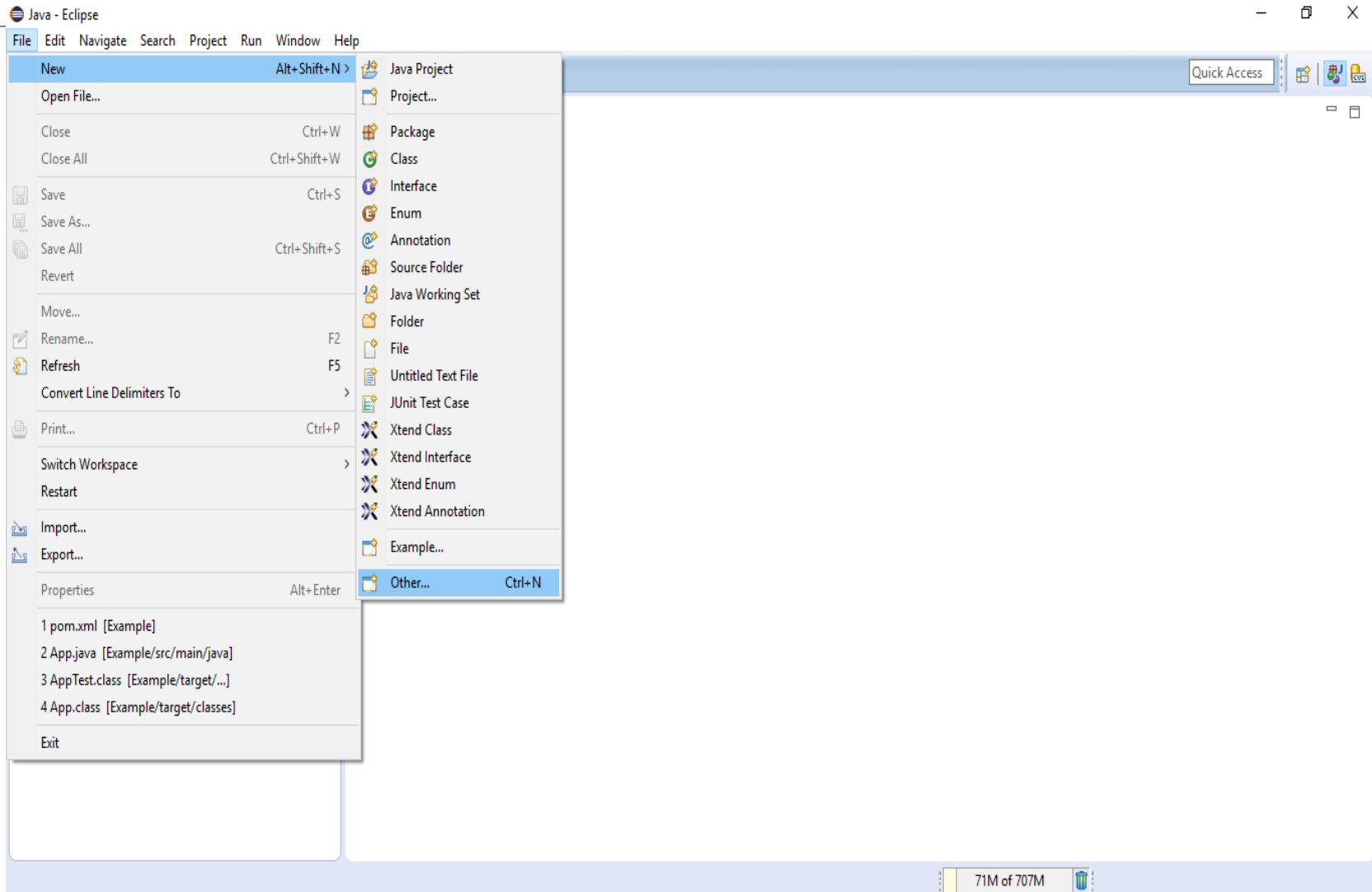
- ❖ **Maven** je veoma snazan alat za menadzment projekta
- ❖ Koristi se build projekta, postavljanje zavisnosti i dokumentaciju.
- ❖ Pojednostavljuje build-ovanje aplikacije
- ❖ Upoređenje aplikacije koja sadrži deo za testiranje razvijena bez Maven-a i iste te aplikacije razvijene sa Maven-om:
<http://www.javacreed.com/why-should-we-use-maven/>

Maven: Instaliranje

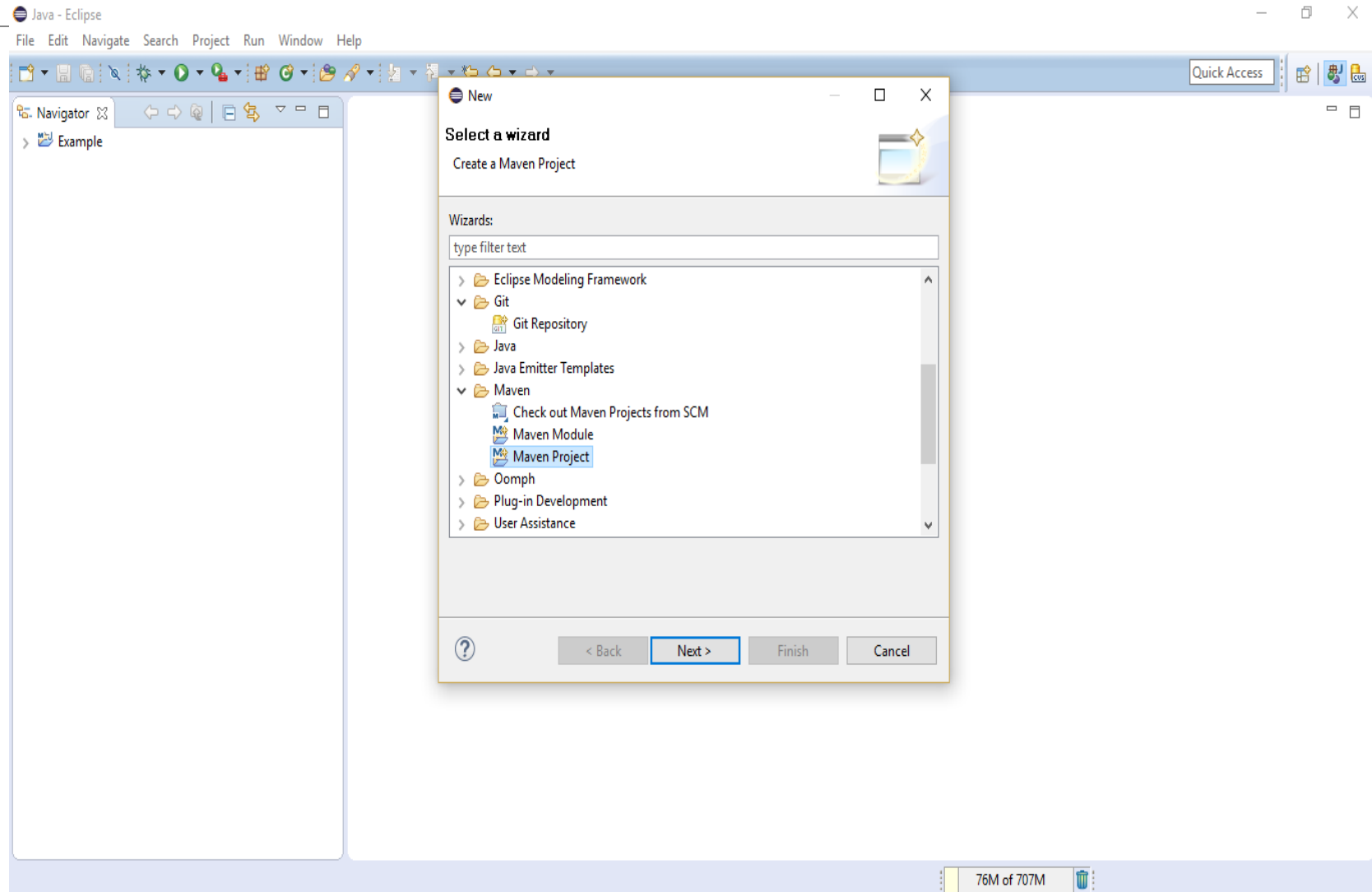
- <http://download.eclipse.org/technology/m2e/releases>



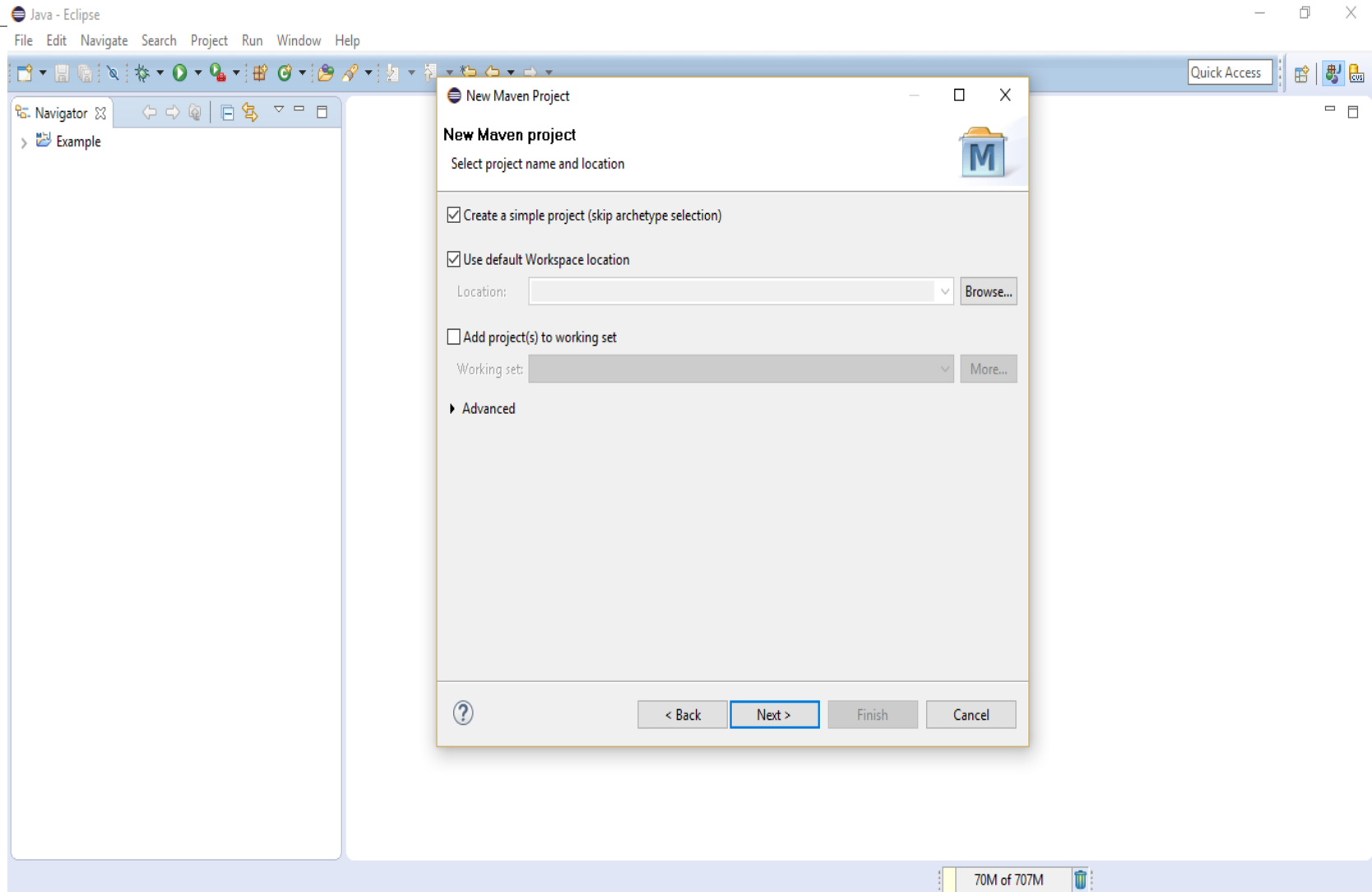
Maven: Kreiranje projekta



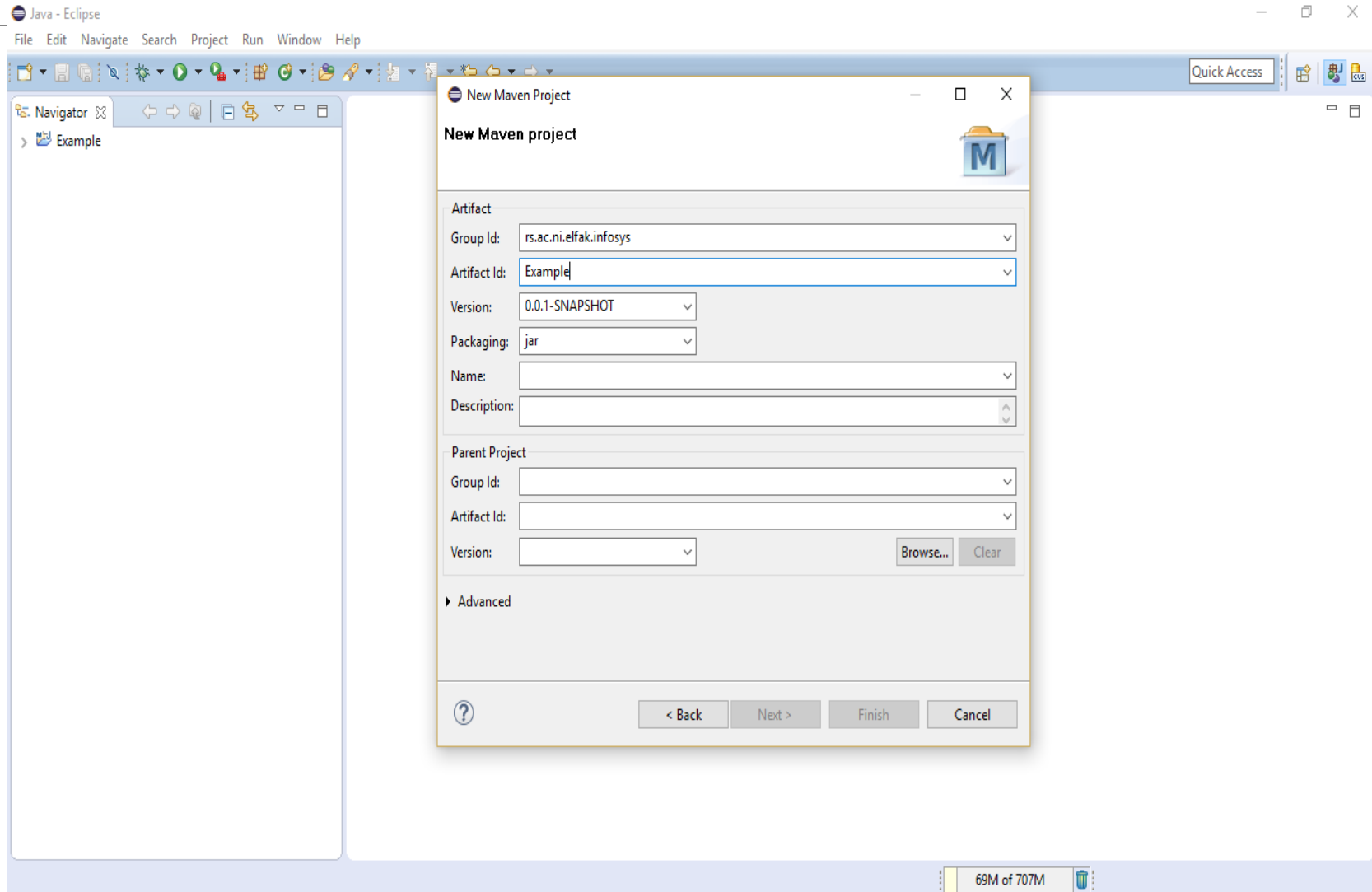
Maven: Kreiranje projekta



Maven: Kreiranje projekta



Maven: Kreiranje projekta

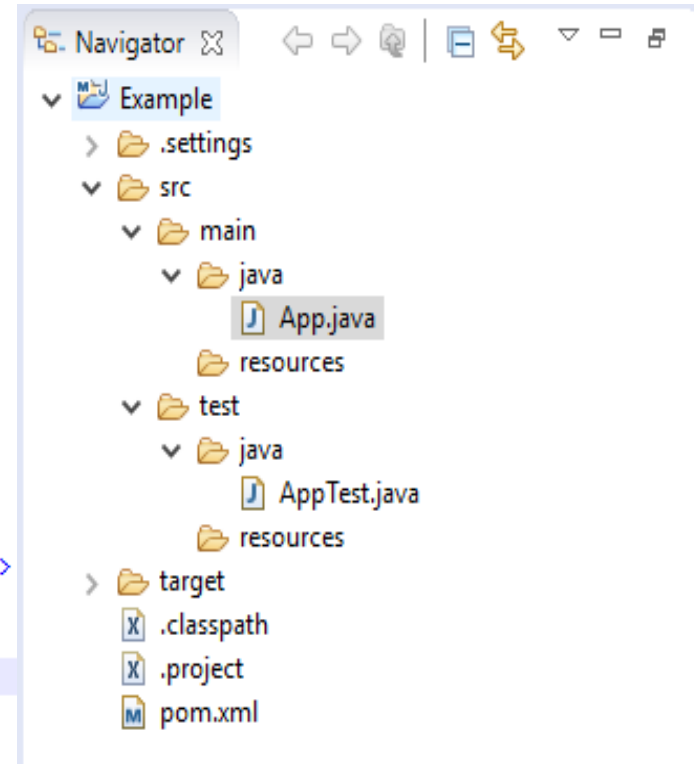


Maven: Primer strukture projekta

Defaults:

- 1.) src/main/java
- 2.) src/test/java
- 3.) pom.xml

```
<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 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>rs.ac.ni.elfak.infosys</groupId>
  <artifactId>Example</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</project>
```



Maven: Primer -> App.java

Java - Example/src/main/java/App.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Navigator

- Example
 - .settings
 - src
 - main
 - java
 - resources
 - test
 - java
 - resources
 - target
 - .classpath
 - .project
 - pom.xml

App.java

```
1 /**
2  * Hello world!
3  *
4  */
5 public class App
6 {
7     App()
8     {
9         System.out.println( "Hello World!" );
10    }
11
12    public int calculateSomething() {
13        return 0;
14    }
15 }
16
```

Writable Smart Insert 16 : 1 77M of 707M

Maven: Kreiranje JAR fajla

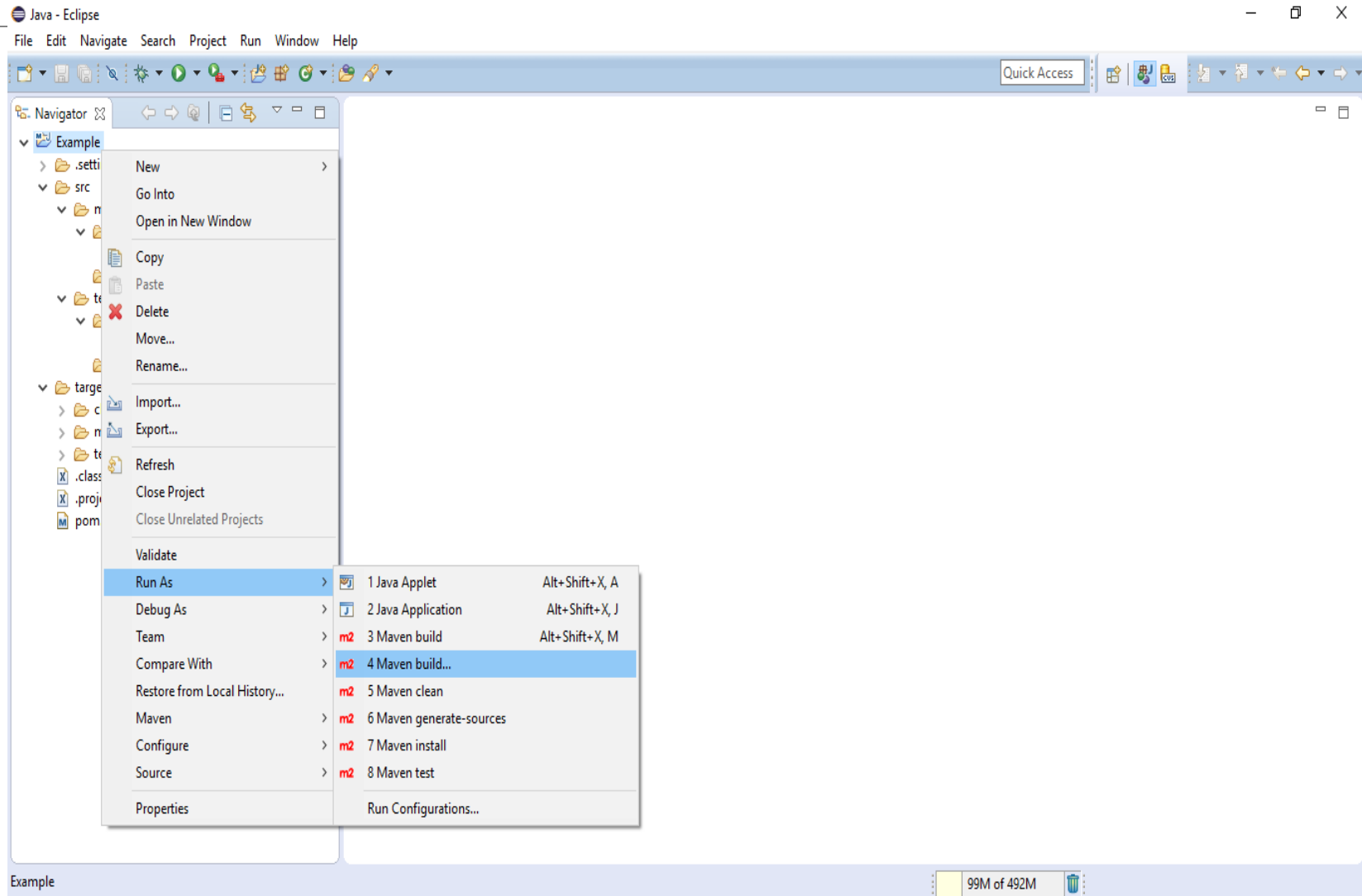
1) App.java -> App.class

2) AppTest.java -> AppTest.class

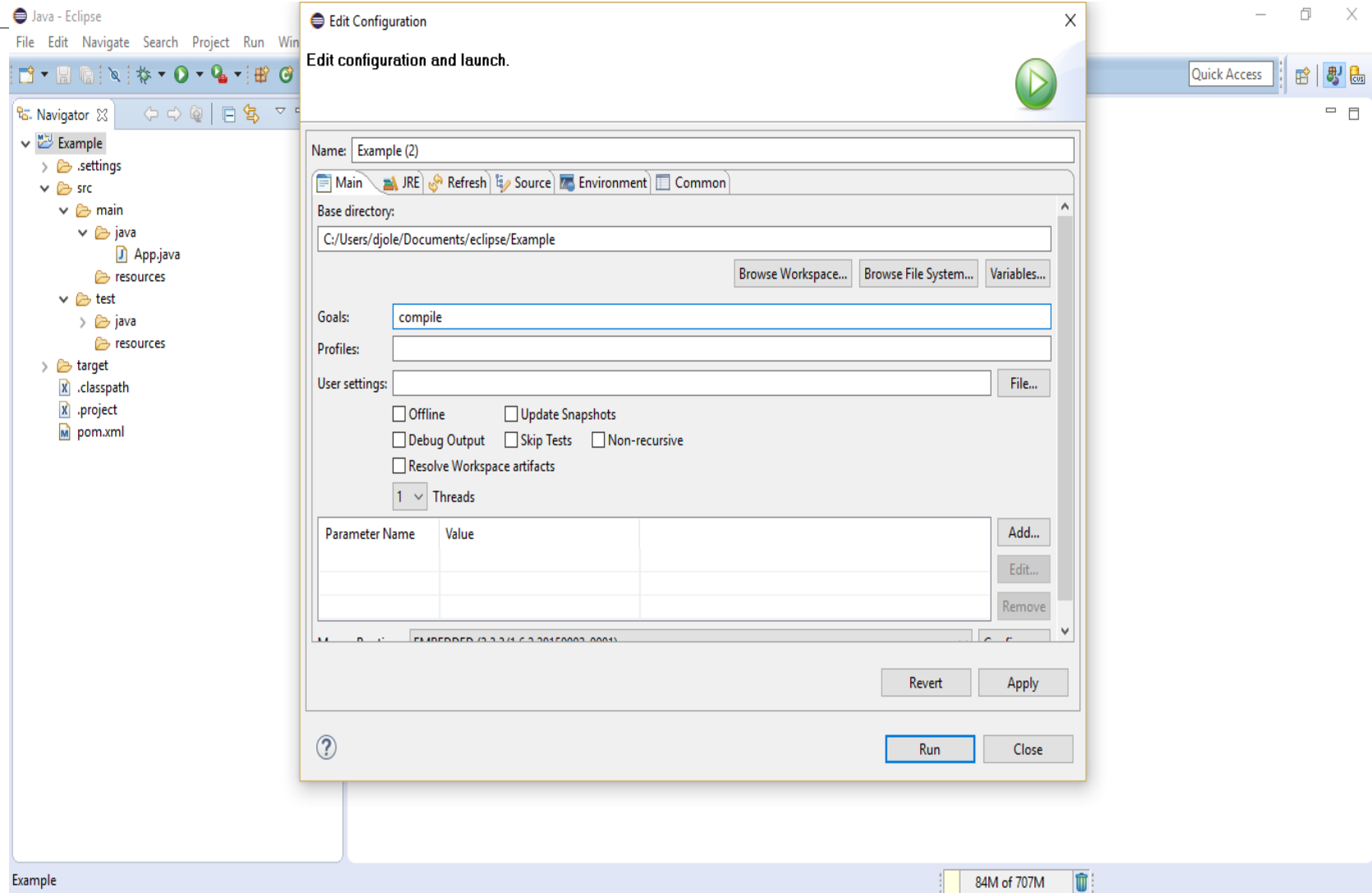
3) Izvršavanje testa

4) Kombinovanje svih class fajlova u jedan JAR

Maven: Kompajliranje



Maven: Kompajliranje



Java - Eclipse

File Edit Navigate Search Project Run Win

Navigator

- Example
 - .settings
 - src
 - main
 - java
 - App.java
 - resources
 - test
 - java
 - resources
 - target
 - .classpath
 - .project
 - pom.xml

Edit Configuration

Edit configuration and launch.

Name: Example (2)

Main JRE Refresh Source Environment Common

Base directory:
C:/Users/djole/Documents/eclipse/Example

Browse Workspace... Browse File System... Variables...

Goals: compile

Profiles:

User settings: File...

☐ Offline ☐ Update Snapshots
☐ Debug Output ☐ Skip Tests ☐ Non-recursive
☐ Resolve Workspace artifacts

1 Threads

Parameter Name	Value

Add... Edit... Remove

Revert Apply

Run Close

Example

84M of 707M

Maven: Kompajliranje

Java - Eclipse

File Edit Navigate Search Project Run Window Help

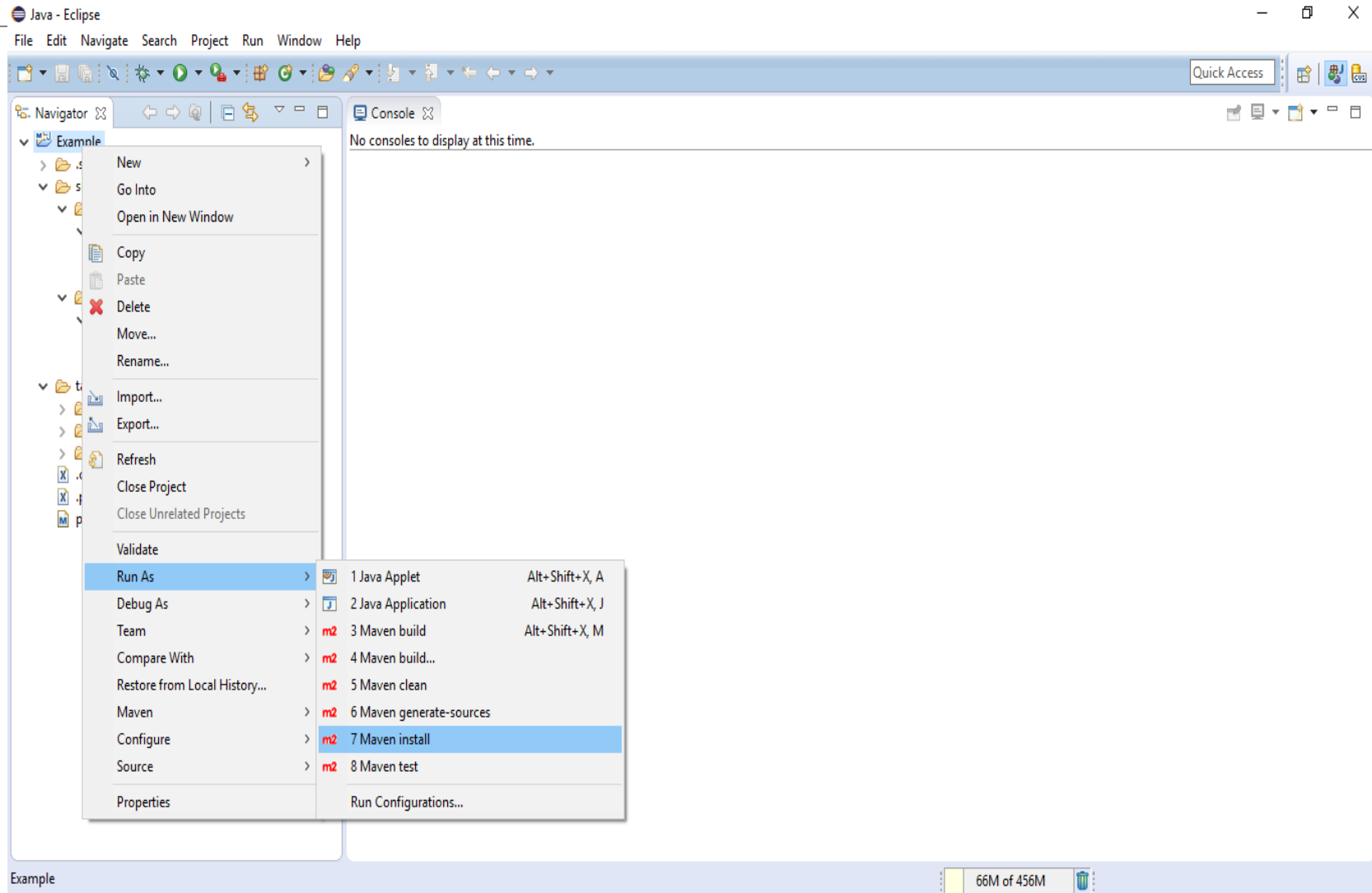
Quick Access

Console

```
<terminated> Example (1) [Maven Build] C:\Program Files\Java\jdk1.8.0_91\bin\javaw.exe (May 5, 2016, 1:42:15 PM)
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Example 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Example ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Example ---
[INFO] Nothing to compile - all classes are up to date
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.219 s
[INFO] Finished at: 2016-05-05T13:42:34+02:00
[INFO] Final Memory: 7M/65M
[INFO] -----
```

102M of 492M

Maven: install



install

```
1 [INFO] Scanning for projects...
2 [INFO] -----
3 [INFO] Building Example 0.0.1-SNAPSHOT
4 [INFO] -----
5 [INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Example ---
6 [WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
7 [INFO] Copying 0 resource
8 [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Example ---
9 [INFO] Changes detected - recompiling the module!
10 [INFO] Compiling 1 source file to C:\Users\djole\Documents\eclipse\Example\target\classes
11 [INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Example ---
12 [WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
13 [INFO] Copying 0 resource
14 [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Example ---
15 [INFO] Changes detected - recompiling the module!
16 [WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
17 [INFO] Compiling 1 source file to C:\Users\djole\Documents\eclipse\Example\target\test-classes
18 [INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Example ---
19 [INFO] Surefire report directory: C:\Users\djole\Documents\eclipse\Example\target\surefire-reports
20 -----
21 T E S T S
22 -----
23 Results :
24 Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
25 [INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Example ---
26 [INFO] Building jar: C:\Users\djole\Documents\eclipse\Example\target\Example-0.0.1-SNAPSHOT.jar
27 [INFO] --- maven-install-plugin:2.4:install (default-install) @ Example ---
28 [INFO] Installing C:\Users\djole\.m2\repository\rs\ac\ni\elfak\infosys\Example\0.0.1-SNAPSHOT\Example-0.0.1-SNAPSHOT.jar
29 [INFO] Installing xml to C:\Users\djole\.m2\repository\rs\ac\ni\elfak\infosys\Example\0.0.1-SNAPSHOT\Example-0.0.1-SNAPSHOT.pom
30 [INFO] -----
31 [INFO] BUILD SUCCESS
32 [INFO] -----
33 [INFO] Total time: 10.245 s
34 [INFO] Finished at: 2016-05-05T22:48:00+02:00
35 [INFO] Final Memory: 15M/115M
```

Maven: Primer lokalnog repozitorijuma

← → ▾ ↑ This PC > Local Disk (C:) > Users > djole > .m2 > repository > rs > ac > ni > elfak > infosys > Example > 0.0.1-SNAPSHOT

Name	Date modified	Type	Size
_remote.repositories	05-May-16 10:48 P...	REPOSITORIES File	1 KB
Example-0.0.1-SNAPSHOT	05-May-16 10:48 P...	Executable Jar File	2 KB
Example-0.0.1-SNAPSHOT.pom	30-Apr-16 6:57 PM	POM File	1 KB
maven-metadata-local	05-May-16 10:48 P...	XML Document	1 KB

- JAR fajlovi
- Jednostavan pristup

Maven: Dodavanje JAR fajla

Java - Example2/pom.xml - Eclipse

File Edit Navigate Search Project Run Design Window Help

Quick Access

Navigator

- Example
- Example2
 - .settings
 - src
 - main
 - java
 - App1.java
 - resources
 - test
 - java
 - resources
 - target
 - .classpath
 - .project
 - pom.xml

Example2/pom.xml

Filter:

Dependencies

Dependencies

Dependency Management

Add...

Remove

Properties...

Manage...

Add...

Remove

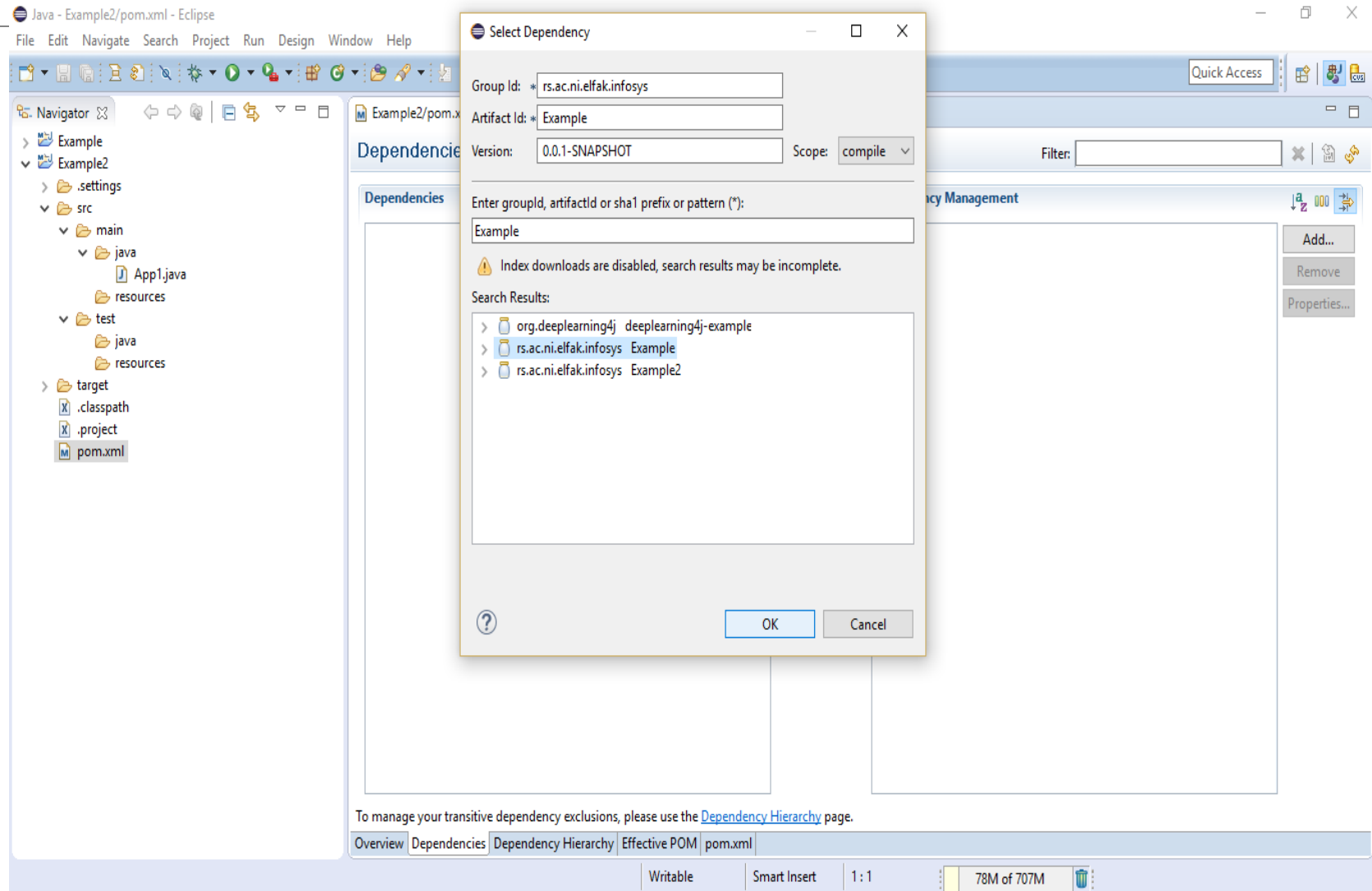
Properties...

To manage your transitive dependency exclusions, please use the [Dependency Hierarchy](#) page.

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Writable Smart Insert 1 : 1 97M of 705M

Maven: Dodavanje JAR fajla



Java - Example2/pom.xml - Eclipse

File Edit Navigate Search Project Run Design Window Help

Navigator

- Example
- Example2
 - .settings
 - src
 - main
 - java
 - App1.java
 - resources
 - test
 - java
 - resources
 - target
 - .classpath
 - .project
 - pom.xml

Dependencies

Enter groupId, artifactId or sha1 prefix or pattern (*):

Example

Index downloads are disabled, search results may be incomplete.

Search Results:

- > org.deeplearning4j deeplearning4j-example
- > rs.ac.ni.elfak.infosys Example
- > rs.ac.ni.elfak.infosys Example2

OK Cancel

To manage your transitive dependency exclusions, please use the [Dependency Hierarchy](#) page.

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Writable Smart Insert 1:1 78M of 707M

Maven: Dodavanje JAR fajla

Java - Example2/pom.xml - Eclipse

File Edit Navigate Search Project Run Design Window Help

Quick Access

Navigator Project Explorer

Example2

- src/main/java
- src/main/resources
- src/test/java
- src/test/resources
- JRE System Library [J2SE-1.5]
- Maven Dependencies
 - Example
- src
- target
- pom.xml

Dependencies

Filter:

Dependencies

- Example : 0.0.1-SNAPSHOT

Add...
Remove
Properties...
Manage...

Dependency Management

Add...
Remove
Properties...

To manage your transitive dependency exclusions, please use the [Dependency Hierarchy](#) page.

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

0 items selected

101M of 709M

Maven: Dodavanje JAR fajla

Java - Example2/pom.xml - Eclipse

File Edit Source Navigate Search Project Run Window Help

Quick Access

Navigator Project Explorer

Example2

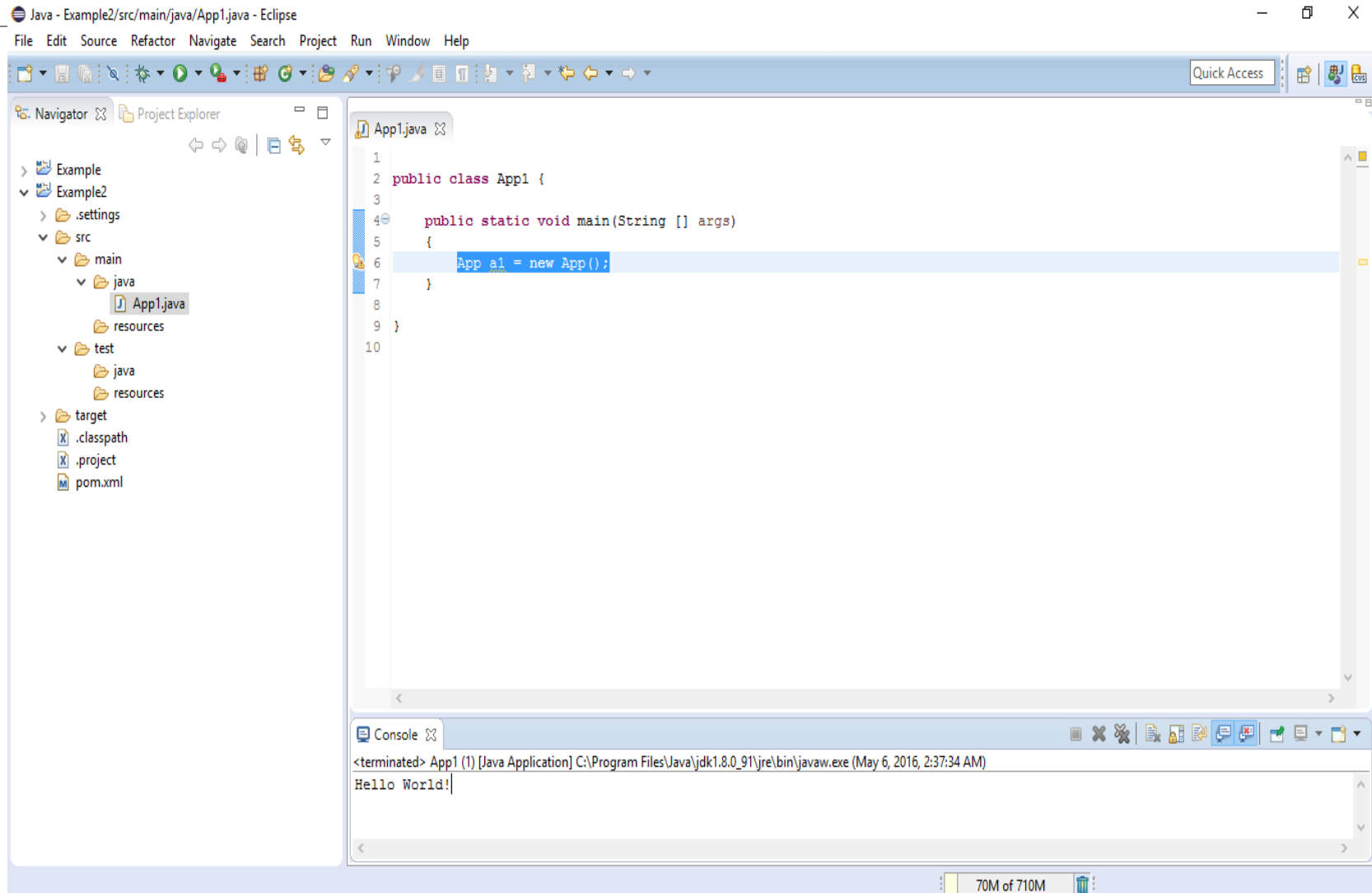
- src/main/java
- src/main/resources
- src/test/java
- src/test/resources
- JRE System Library [J2SE-1.5]
- Maven Dependencies
 - Example
- src
- target
- pom.xml

```
1 <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 http://maven.apache.org/xsd/maven-4.0.0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>rs.ac.ni.elfak.infosys</groupId>
4   <artifactId>Example2</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <dependencies>
7     <dependency>
8       <groupId>rs.ac.ni.elfak.infosys</groupId>
9       <artifactId>Example</artifactId>
10      <version>0.0.1-SNAPSHOT</version>
11    </dependency>
12  </dependencies>
13 </project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

project/dependencies/dependency/#text Writable Smart Insert 10:42 75M of 708M

Maven: Dodavanje JAR fajla



Maven: Arhetipovi

- **Arhetipovi se koriste da bi se kreirala odgovarajuća struktura projekta**

```
mvn archetype:create
  -DgroupId=[your project's group id]
  -DartifactId=[your project's artifact id]

.
|-- src
|   |-- main
|   |   |-- java
|   |   |   |-- [your project's package]
|   |   |   |-- App.java
|   |-- test
|   |   |-- java
|   |   |   |-- [your project's package]
|   |   |   |-- AppTest.java
|-- pom.xml
```

Maven: Arhetipovi

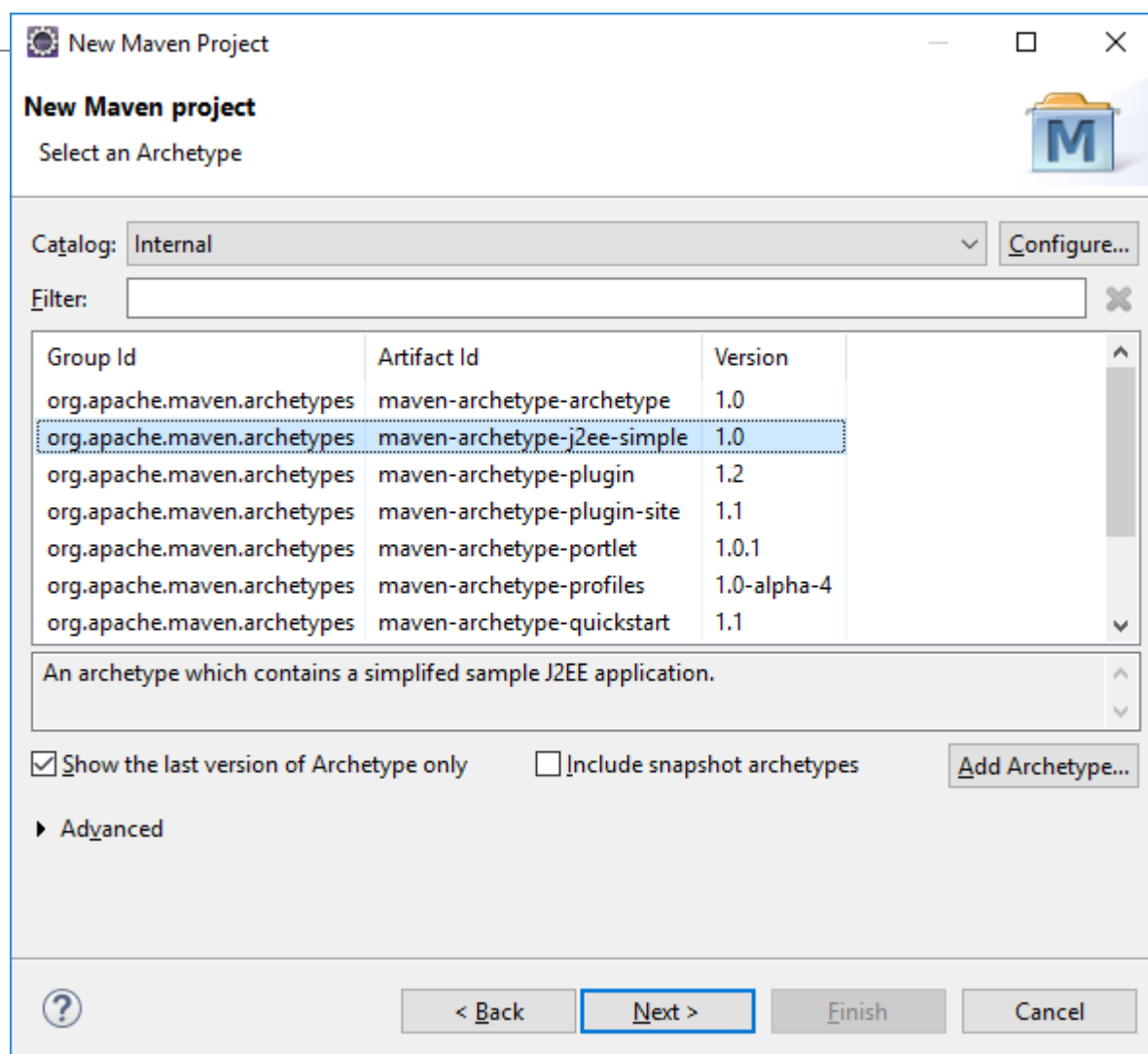
```
mvn archetype:create
```

```
-DgroupId=your.simple.java.gid
```

```
-DartifactId=your-simple-java-aid
```

```
.
|-- src
|   |-- main
|   |   |-- java
|   |   |   |-- your
|   |   |   |   |-- simple
|   |   |   |   |   |-- java
|   |   |   |   |   |   |-- aid
|   |   |   |   |   |   |   |-- App.java
|   |-- test
|   |   |-- java
|   |   |   |-- your
|   |   |   |   |-- simple
|   |   |   |   |   |-- java
|   |   |   |   |   |   |-- aid
|   |   |   |   |   |   |   |-- AppTest.java
|-- pom.xml
```

Maven: Arhetipovi



Maven: Transformacija

- **Za već kreiran projekat u Eclipse-u moguće je izvršiti njegovu transformaciju u Maven.**

❖ Desni klik na projekat/Configure→Convert to Maven Project

