

Аспектно - ориентированное программирование

- ▼ objects
 - SonyHand
 - SonyHead
 - SonyLeg
 - ToshibaHead
- ▼ start
 - Robot
 - RobotManager

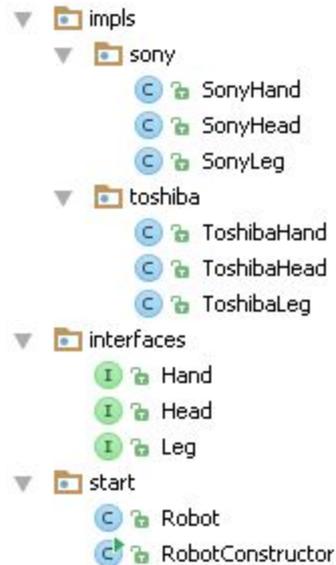
```
public class RobotManager {  
  
    public static void main(String[] args){  
        Robot robot = new Robot();  
        robot.action();  
    }  
}
```

```
public class Robot {  
  
    private SonyHand hand = new SonyHand();  
    private SonyLeg leg = new SonyLeg();  
    //private SonyHead head = new SonyHead();  
    private ToshibaHead head = new ToshibaHead();  
    public void action(){  
        head.calc();  
        hand.catchSomething();  
        leg.go();  
    }  
}
```

```
public class SonyLeg {  
    public void go(){  
        System.out.println("Go!");  
    }  
}  
  
public class SonyHead {  
    public void calc(){  
        System.out.println("Thinking...");  
    }  
}
```

```
public class ToshibaHead {  
    public void calc(){  
        System.out.println("Toshiba Thinking...");  
    }  
}
```

```
public class SonyHand {  
    public void catchSomething() { System.out.println("Caught!"); }  
}
```



```
public interface Head {  
  
    public void calc();  
}
```

```
public class ToshibaHead implements Head{  
  
    public void calc(){  
        System.out.println("Thinking about Toshiba...");  
    }  
}
```

```
public class SonyHead implements Head{  
  
    public void calc() { System.out.println("Thinking about Sony..."); }  
  
}
```

```
public class RobotConstructor {
```

```
    public static void main(String[] args){
```

```
        SonyHand sonyHand = new SonyHand();
```

```
        SonyLeg toshibaLeg = new SonyLeg();
```

```
        SonyHead sonyHead = new SonyHead();
```

```
        Robot robot = new Robot(sonyHand, toshibaLeg, sonyHead);
```

```
        robot.action();
```

```
    }
```

```
public class Robot {
```

```
    private Hand hand;
```

```
    private Leg leg;
```

```
    private Head head;
```

```
    public Robot(Hand hand, Leg leg, Head head) {
```

```
        super();
```

```
        this.hand = hand;
```

```
        this.leg = leg;
```

```
        this.head = head;
```

```
    }
```

```
    public void action(){
```

```
        head.calc();
```

```
        hand.catchSomething();
```

```
        leg.go();
```

```
    }
```

Фреймворк – что это?

- Фреймворк это программная оболочка, позволяющая упростить и ускорить решение типовых задач, характерных для данного языка программирования. Само слово `framework` означает «каркас» в переводе с английского.

Классические фреймворки:

- Javascript - jQuery – библиотека с готовыми функциями визуальных эффектов, AJAX-запросов и прочих полезных вещей.
- PHP - Yii
- Java – Spring

Фреймворк отличается от понятия библиотеки тем, что библиотека может быть использована в программном продукте просто как набор подпрограмм близкой функциональности, не влияя на архитектуру программного продукта и не накладывая на неё никаких ограничений.

В то время как фреймворк диктует правила построения архитектуры приложения, задавая на начальном этапе разработки поведение по умолчанию, каркас, который нужно будет расширять и изменять согласно указанным требованиям.

Spring

- Основная цель - упрощение разработки любых приложений на Java, разгрузка кода
 - Программный код становится проще, связь между объектами слабее
 - Каждый объект занимается свои делом (POJO)
 - Использование принципов ООП на полную мощность
 - Дополнительная логика подключается извне
 - Готовые встроенные модули (работа с БД, безопасность, транзакции, авторизация, сервисы и пр.)
 - **Spring** – огромный комплекс, который может объединить и упростить использование технологий
-

Проекты Spring

Список проектов

<http://spring.io/projects>

- **Spring Framework - основы**
- Spring Boot
- Spring XD
- Spring Data
- Spring Integration
- Spring Batch
- Spring Security
- Spring Hateoas
- Spring Social
- Spring AMQP
- Spring Mobile
- Spring for Android
- Spring WebFlow
- Spring WebServices
- Spring LDAP
- ...

Spring

IoC

Inversion of Control (IoC)

AOP

Aspect-oriented programming

Автомобиль Lexus

Летние шины Bridgestone

Бензин 92

Медиасистема Sony

- «Жесткое» связывание
- Трудно изменять код
- Трудно тестировать
- Трудно заменять объекты

Автомобиль

Летние шины

Бензин

Медиасистема

- Слабые связи (на уровне абстракций)
- Код изменять легче
- Легко тестировать
- Делегирование создания объектов контейнеру

IoC контейнер

Основной момент IoC:

- Сначала создаете абстракцию
- Потом подставляете реализацию

Контейнер

- Управляет зависимостями
- Связывает объекты между собой
- Управляет их жизненным циклом
- **Dependency Injection** – в объект внедряется ссылка на другой объект

АОП

- Разделение основного функционала и дополнительного – без перемешивание их между собой
- Кеширование, логирование, транзакции, безопасность и пр.
- Аспект - функциональность, не относящаяся напрямую к бизнес логике
- Аспекты можно использовать в любых проектах

Безопасность

Транзакции

Логирование

Кеширование

Объект 1

Объект 2

Объект 3

Spring

Основной момент:

- Создание объектов
 - Настройка взаимодействия (вне объектов)
 - Подключение дополнительных аспектов
-

```
public class ModelT1000 implements Robot {

    private Hand hand;
    private Leg leg;
    private Head head;

    public ModelT1000() {
    }

    public ModelT1000(Hand hand, Leg leg, Head head) {
        super();
        this.hand = hand;
        this.leg = leg;
        this.head = head;
    }

    @Override
    public void fire() {
        head.calc();
        hand.catchSomething();
        leg.go();
    }

    @Override
    public void dance() { System.out.println("T1000 is dancing!"); }

}
```

```
public interface Robot {
    void fire();

    void dance();
}
```

```
public interface Leg {  
    public void go();  
}
```

```
public interface Head {  
    public void calc();  
}
```

```
public interface Hand {  
    public void catchSomething();  
}
```

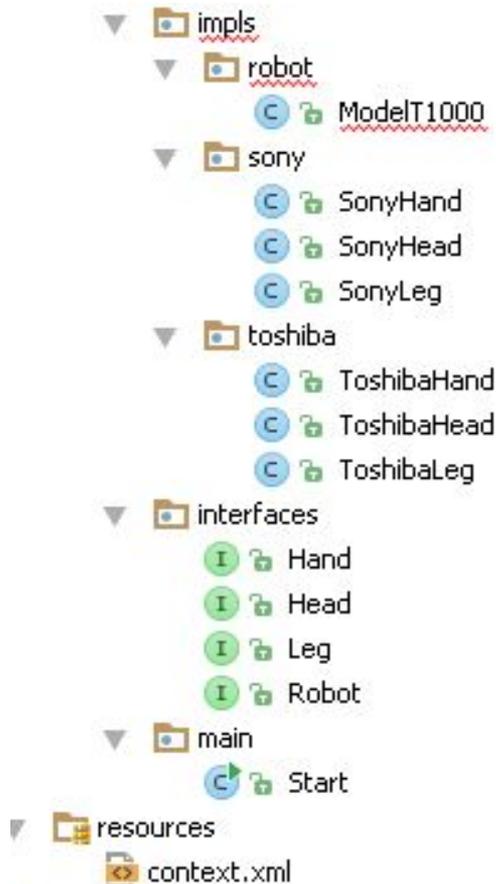
```
public class SonyHand implements Hand{  
    public void catchSomething(){  
        System.out.println("Caught from Sony!!");  
    }  
}
```

```
public class ToshibaHand implements Hand{  
    public void catchSomething(){  
        System.out.println("Caught from Toshiba!");  
    }  
}
```

```
public class SonyHead implements Head{  
    public void calc(){  
        System.out.println("Thinking about Sony...");  
    }  
}
```

```
public class SonyLeg implements Leg {  
    public void go(){  
        System.out.println("Go to Sony!");  
    }  
}
```

Проект робот в Spring



```
public class Start {  
    public static void main(String[] args) {  
        ApplicationContext context = new ClassPathXmlApplicationContext("context.xml");  
        ModelT1000 t1000 = (ModelT1000) context.getBean("t1000");  
        t1000.fire();  
    }  
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://www.springframework.org/schema/beans http  
  
    <bean id="t1000" class="ru.spring.impls.robot.ModelT1000" />  
    <!--bean id="t1000" class="ru.spring.impls.robot.ModelT1000New"/-->  
  
</beans>
```

-
- Установка eclipse – среда разработки
 - Установка Maven – инструмент сборки
 - Установка Spring – core библиотеки
 - Создание тестового проекта – проверка работоспособности
-

Инструменты

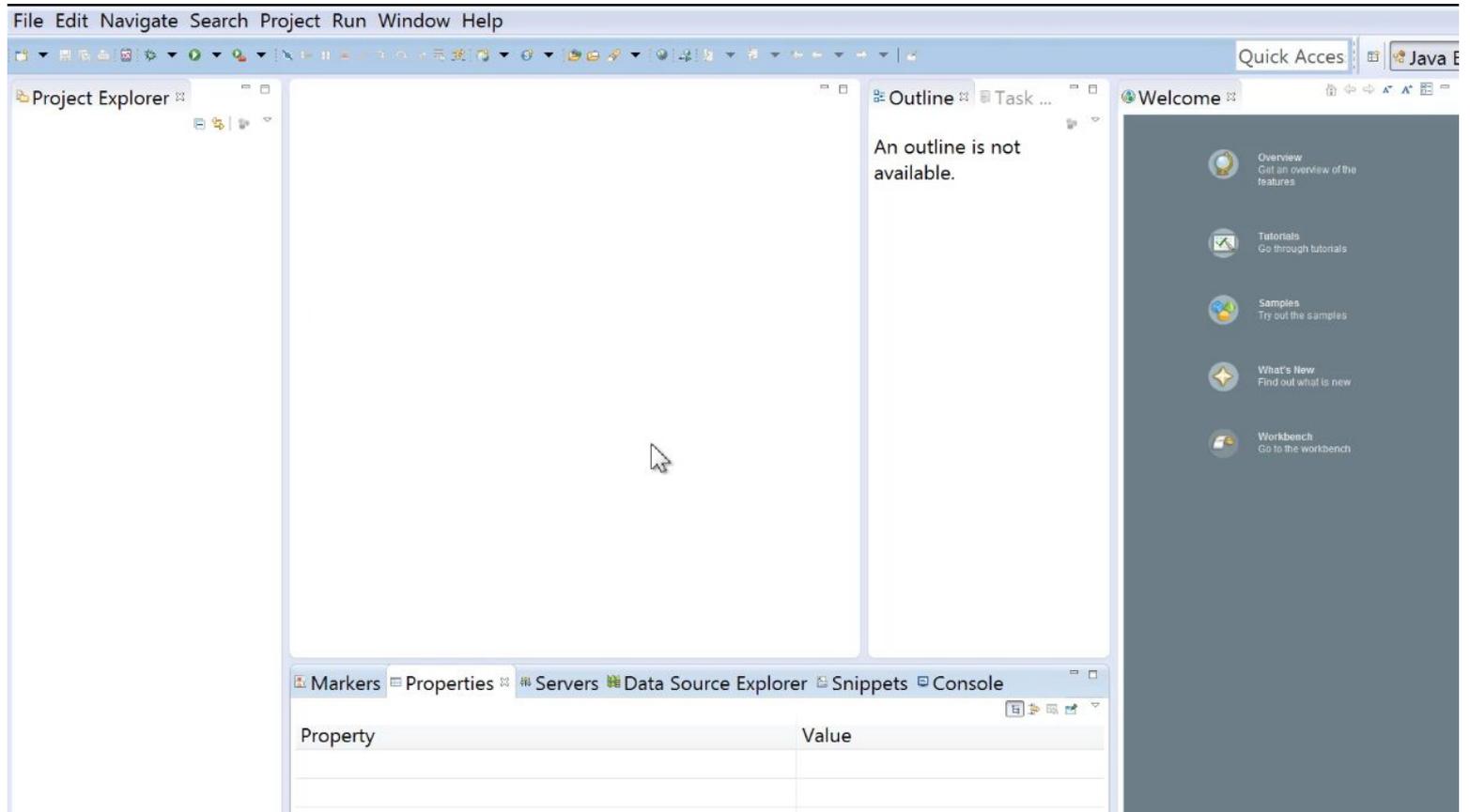
- Eclipse IDE for Java EE Developers <https://www.eclipse.org/downloads/>

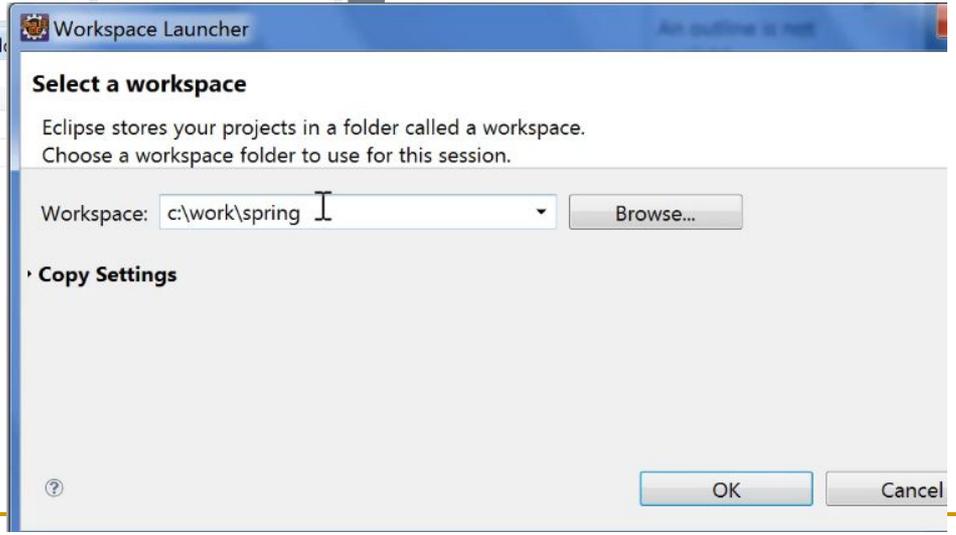
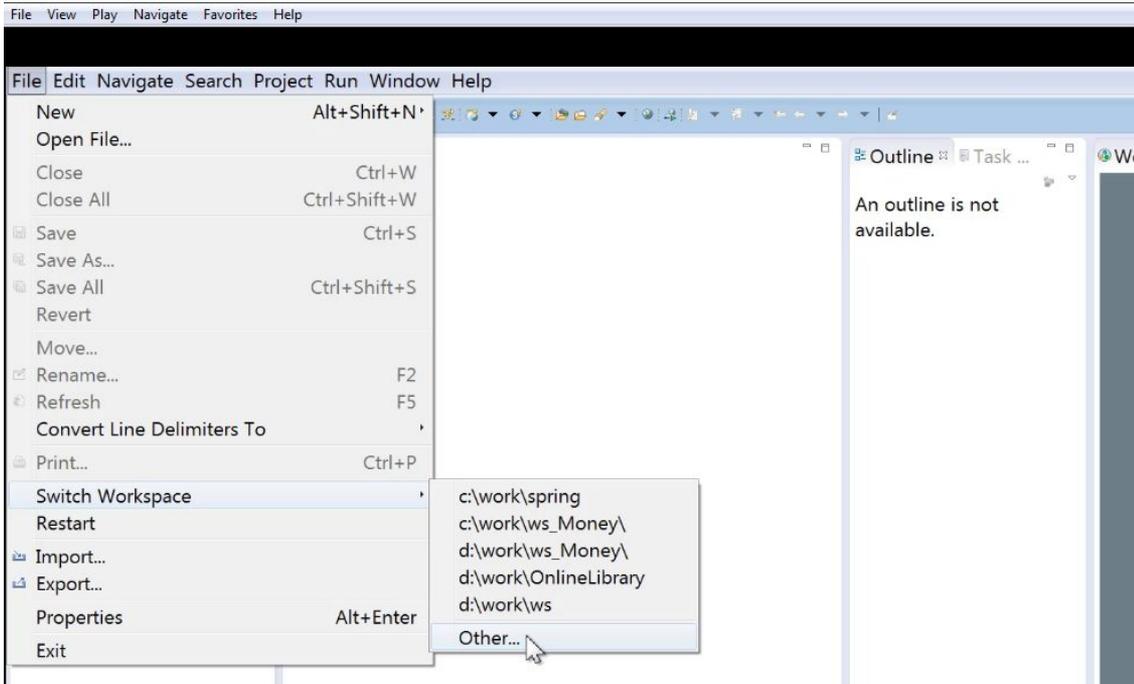
The screenshot shows the Eclipse Downloads website. The browser address bar displays <https://www.eclipse.org/downloads/>. The page features a navigation menu with links for Home, Downloads, Users, Members, Committers, Resources, Projects, and About Us. A search bar for Google Custom Search is also present. The main heading is "Eclipse Downloads". Below this, there are tabs for Packages, Java™ 8 Support, and Developer Builds. A dropdown menu shows "Eclipse Kepler (4.3.2) SR2 Packages for Windows". The main content area lists several packages:

- Eclipse Standard 4.3.2**, 200 MB
Downloaded 2,723,281 Times [Other Downloads](#)
The Eclipse Platform, and all the tools needed to develop and debug it: Java and Plug-in Development Tooling, Git and CVS...
- Eclipse IDE for Java EE Developers**, 250 MB
Downloaded 1,500,095 Times
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...
- Eclipse IDE for Java Developers**, 153 MB
Downloaded 640,326 Times

Each package listing includes a download icon and links for Windows 32 Bit and Windows 64 Bit. A "Filter Packages" button is visible. On the right side, there is a "Related Links" section with links like "Compare & Co", "Eclipse Indigo", "Eclipse Juno", "Install Guide", "Documentation", "Updating Eclip", "Forums", and "Older Version". A "Hint" section at the bottom right states: "You will need a [Java environment \(JRE\)](#) SE 6 or greater is required. Downloads are provided for Windows, Linux, and Mac OS X."

Eclipse





File Edit Navigate Search Project Run Window Help

Project Explorer

- New Window
- New Editor
- Hide Toolbar
- Open Perspective
- Show View
- Customize Perspective...
- Save Perspective As...
- Reset Perspective...
- Close Perspective
- Close All Perspectives

- Navigation
- Web Browser
- Preferences
- Genymobi
- Help
- Install/Upd
- Java
- Java EE
- Java Persis
- JavaScript
- Maven
 - Archety
 - Discover
 - Installati
 - Lifecycle
 - Templat
 - User Inte
 - User Set
 - Warning
- Mylyn
- Plug-in De
- Remote Sy
- Run/Debu
- Server
- Team
- Terminal
- Validation
- Web
- Web Servi

Installations

Select the installation used to launch Maven:

- Embedded (3.0.4/1.4.0.201305)

Add... Edit... Remove

Note: Embedded runtime is always used for dependency resolution, but does not use global settings when it is used to launch Maven. To learn more, visit the [Maven](#) web page.

Global settings for embedded installation: Browse...

Outline

An outline is no available.

Data Source Explorer Snippets Console

Value

ИНСТРУМЕНТЫ

- Maven <http://maven.apache.org/download.cgi>
- Maven Eclipse Integration <https://www.eclipse.org/m2e/>



→ Main

Welcome

→ Get Maven

Download

Releases History
Release Notes (3.2.1)
Release Notes (3.1.1)
Release Notes (3.0.5)
License
Security

→ IDE Integration

Eclipse
NetBeans

→ About Maven

What is Maven?
Features
FAQ (official)
FAQ (unofficial)

Download Apache Maven 3.2.1

Maven is distributed in several formats for your convenience. Use a source Maven yourself. Otherwise, simply pick a ready-made binary distribution instructions given at the end of this document.

You will be prompted for a mirror - if the file is not found on yours, please hours to reach all mirrors.

In order to guard against corrupted downloads/installations, it is highly recommended to check the [signature](#) of the release bundles against the public [KEYS](#) used by the Apache Project.

Maven is distributed under the [Apache License, version 2.0](#).

We **strongly** encourage our users to configure a Maven repository please read [How to Use Mirrors for Repositories](#).

Be sure to check the [compatibility notes](#) before using this version to avoid being backward-compatible with Maven 2.x to the extent possible, there

Files Mark Commands Net Show Configuration Start



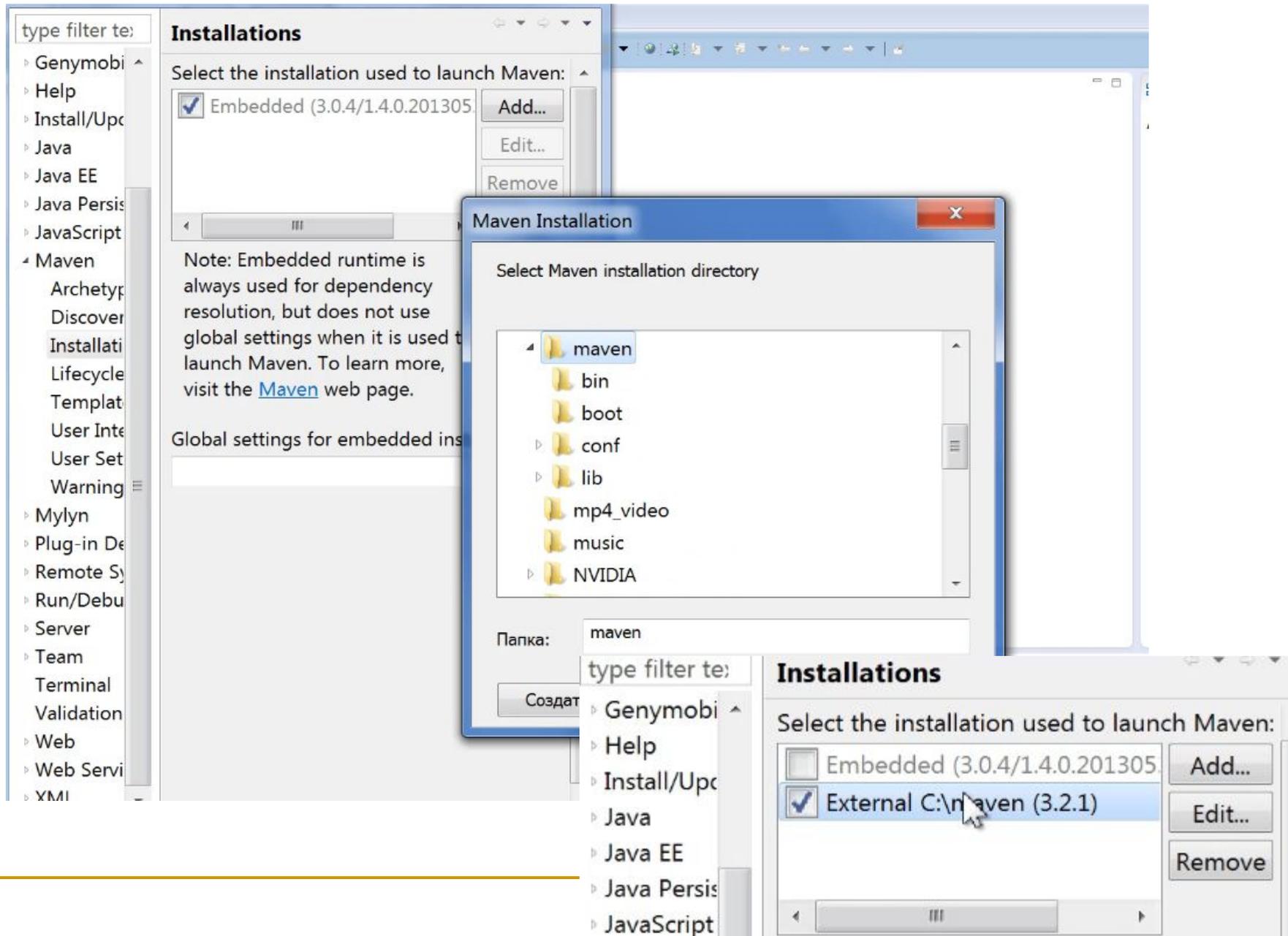
c \ [bootcamp] 2 589 476 k of 326 171 644 k free

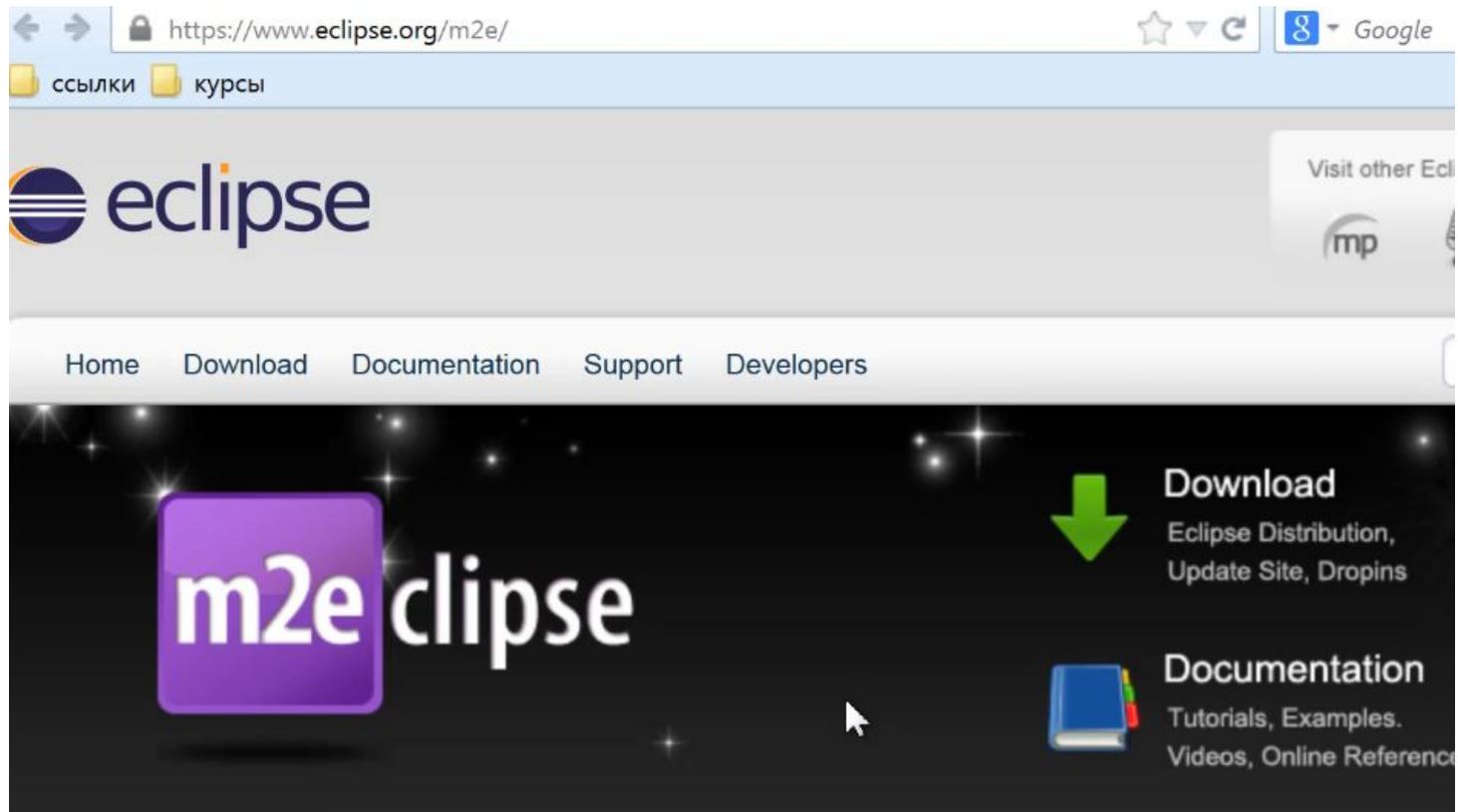
\ .. c \ [bootcamp] 2 590 324 k of 326 171 644 k free

c:\maven*.*

c:\maven*.*

Name	↑Ext	Size	Date	↑Name
[..]		<DIR>	11.04.	[..]
[bin]		<DIR>	11.04.	[bin]
[boot]		<DIR>	11.04.	[boot]
[conf]		<DIR>	11.04.	[conf]
[lib]		<DIR>	11.04.	[lib]
LICENSE		14 865	14.02.	LICENSE
NOTICE		182	14.02.	NOTICE
README	txt	2 513	14.02.	README





Maven Integration (m2e)

The goal of the m2e project is to provide a first-class [Apache Maven](#) support in the Eclipse IDE, making it easier to edit Maven's pom.xml, run a build from the IDE and much more. For Java developers, the very tight integration with JDT greatly simplifies the consumption of Java artifacts, either being hosted on open source repositories such as [Maven Central](#) or in

- ссылки  курсы
- Download
- Documentation
- Support
- Getting Involved

Install

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

m2e is tested against Eclipse 4.2 (Juno) and 4.3 (Kepler).

See http://wiki.eclipse.org/M2E_updatesite_and_gittags for detailed information at available builds and m2e build repository layout.

m2e 1.3 and earlier version have been removed from the main m2e update site. Old releases are still available and can be installed from repositories documented in http://wiki.eclipse.org/M2E_updatesite_and_gittags

Please note that links below point at Eclipse **p2 repositories**; you must access them from Eclipse ([see how](#)).

Update Sites

Latest m2e release (recommended)

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

m2e milestone builds towards version 1.5

<http://download.eclipse.org/technology/m2e/milestones/1.5>

Latest m2e 1.5 SNAPSHOT build (not tested, not hosted at eclipse.org)

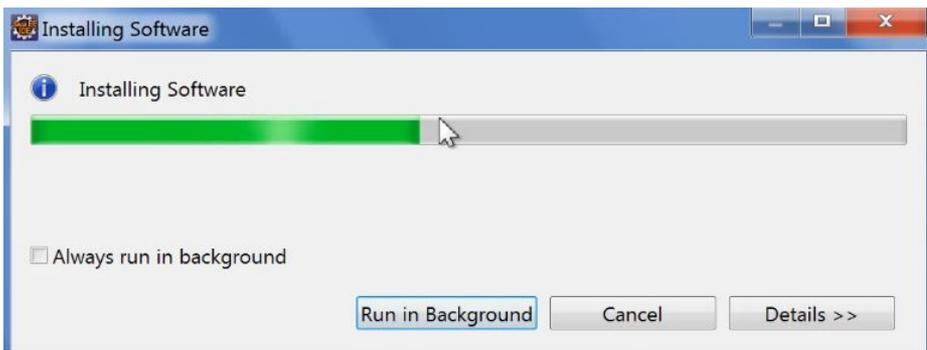
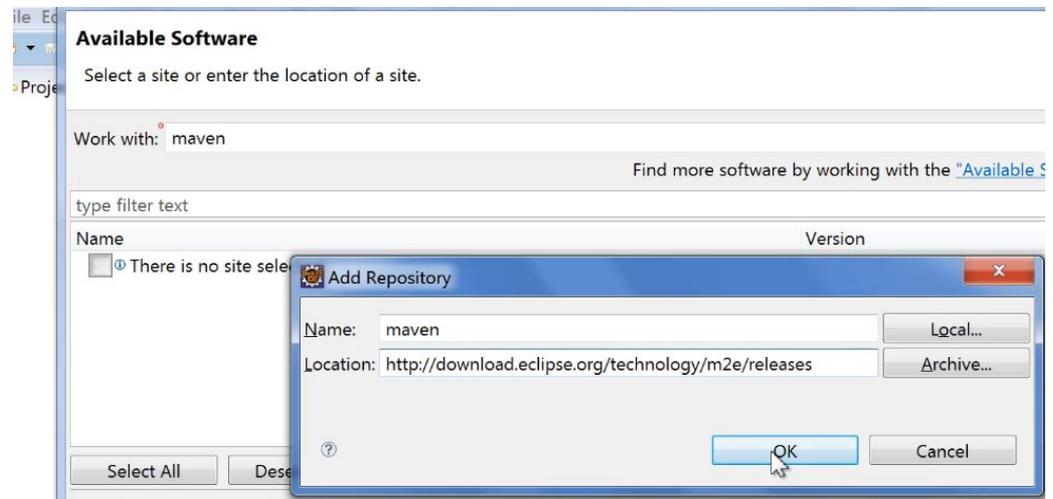
File Edit Navigate Search Project Run Window Help



Project Explorer

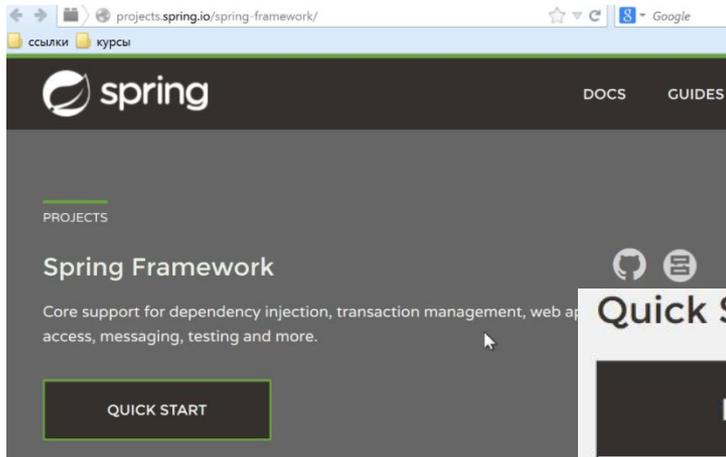


- Welcome
- Help Contents
- Search
- Dynamic Help
- Key Assist... Ctrl+Shift+L
- Tips and Tricks...
- Report Bug or Enhancement...
- Cheat Sheets...
- Eclipse Marketplace...
- Check for Updates
- Install New Software...
- About Eclipse



Инструменты

- Spring Framework <http://projects.spring.io/spring-framework/>



Quick Start

Download

4.0.3 **CURRENT** 

MAVEN GRADLE

The recommended way to get started using `spring-framework` in your project is with a dependency management system – the snippet below can be copied and pasted into your build. Need help? See our getting started guides on building with [Maven](#) and [Gradle](#).

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>4.0.3.RELEASE</version>
  </dependency>
</dependencies>
```

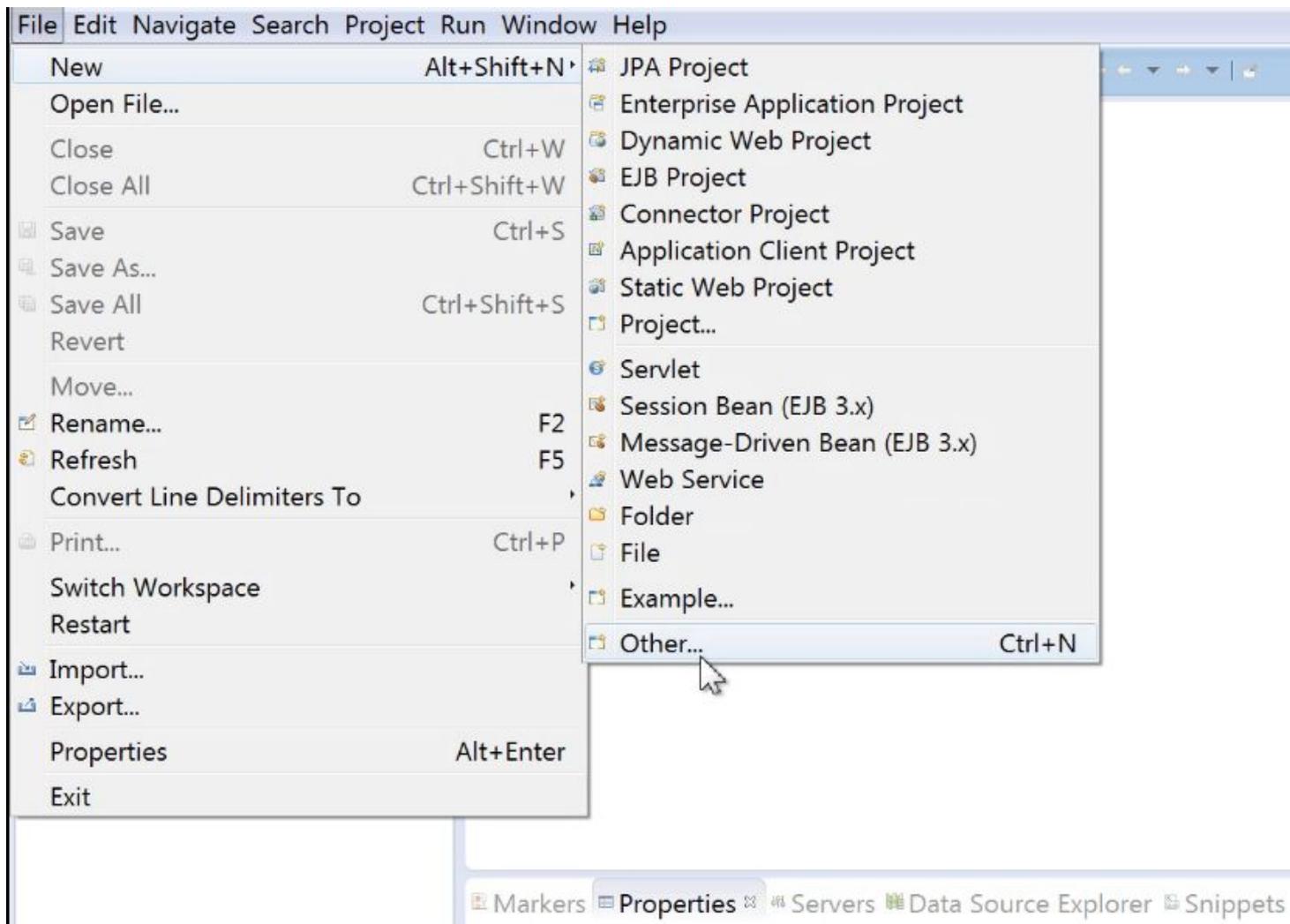


```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>4.0.3.RELEASE</version>
  </dependency>
</dependencies>
```

- Копировать
- Выделить всё
- Искать «<dependency> <g...>» в Google
- Перевести выбранное с помощью Переводчика Google
- Исходный код выделенного фрагмента
- Исследовать элемент
-  Read in Reader

Spring Framework includes... which provides core functionality... options.

ng spi
ght fo



New

Select a wizard

Create a Maven Project

Wizards:

type filter text

- Maven
 - Checkout Maven Projects from SCM
 - Maven Module
 - Maven Project**
- Plug-in Development
- Remote System Explorer
- Server
- SQL Development
- Tasks
- User Assistance
- Web
- Web Services

[?](#) [< Back](#) [Next >](#)

New Maven Project

New Maven project

Select an Archetype

Catalog: All Catalogs [Configure...](#)

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-portlet	RELEASE
org.apache.maven.archetypes	maven-archetype-profiles	RELEASE
org.apache.maven.archetypes	maven-archetype-quickstart	RELEASE
org.apache.maven.archetypes	maven-archetype-site	RELEASE
org.apache.maven.archetypes	maven-archetype-site-simple	RELEASE
org.apache.maven.archetypes	maven-archetype-webapp	RELEASE

Show the last version of Archetype only Include snapshot archetypes [Add Archetype...](#)

Advanced

New Maven Project

New Maven project

Specify Archetype parameters

Group Id: test

Artifact Id: test

Version: 0.0.1-SNAPSHOT

Package: test.test

Properties available from

Name	Value

Project Explorer

test

src/main/java

src/test/java

JRE System Library [J2SE-1.5]

Maven Dependencies

src

target

pom.xml

pom.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/xsi:schemaLocation" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>test</groupId>
  <artifactId>test</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
```

```
  <name>test</name>
  <url>http://maven.apache.org</url>
```

```
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
```

```
  <dependencies>
    <dependency>
```

pom.xml

Markers Properties Servers Database

Property

pom.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/xsi:schemaLocation" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>test</groupId>
  <artifactId>test</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
```

```
  <name>test</name>
  <url>http://maven.apache.org</url>
```

```
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
```

```
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

```
</project>
```

```
File Edit Source Navigate Search Project Run Window Help
pom.xml
<xsi:schemaLocation="http://maven.apache.org/POM
<modelVersion>4.0.0</modelVersion>

<groupId>test</groupId>
<artifactId>test</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>test</name>
<url>http://maven.apache.org</url>

<properties>
  <project.build.sourceEncoding>UTF-8</project.
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>4.0.3.RELEASE</version>
```

File Edit Navigate Search Project Run Window Help

Project Explorer

- test
 - src/main/java
 - src/test/java
 - JRE System Library [J2SE-1.5]
 - Maven Dependencies
 - junit-3.8.1.jar - C:\Users\Tim\
 - spring-context-4.0.3.RELEASE
 - spring-aop-4.0.3.RELEASE.jar
 - aopalliance-1.0.jar - C:\Users\
 - spring-beans-4.0.3.RELEASE.jar
 - spring-core-4.0.3.RELEASE.jar
 - commons-logging-1.1.3.jar -
 - spring-expression-4.0.3.RELE
 - src
 - target
 - pom.xml

pom.xml

```
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 h
<modelVersion>4.0.0</modelVersion>

<groupId>test</groupId>
<artifactId>test</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>test</name>
<url>http://maven.apache.org</url>

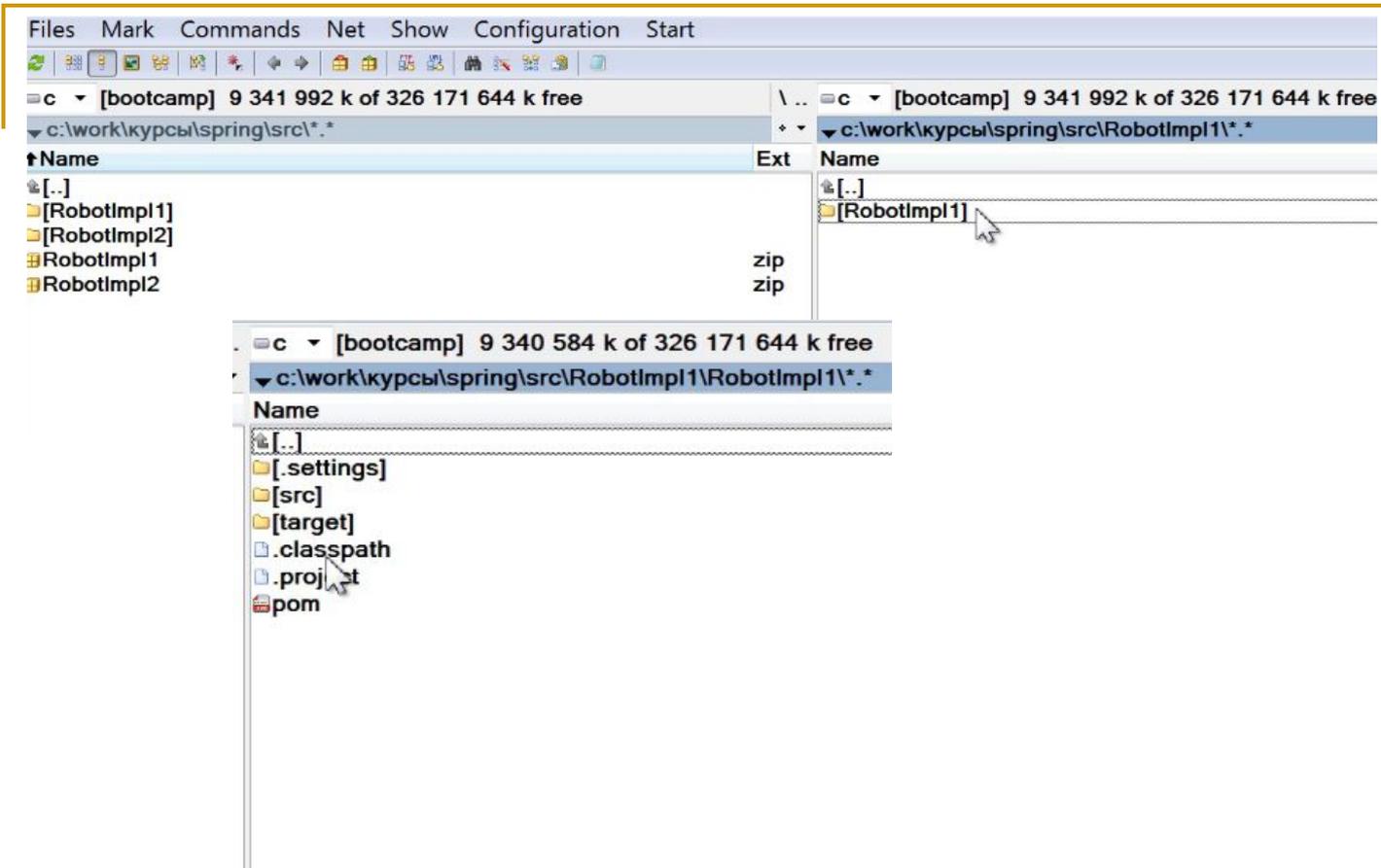
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sou
</properties>

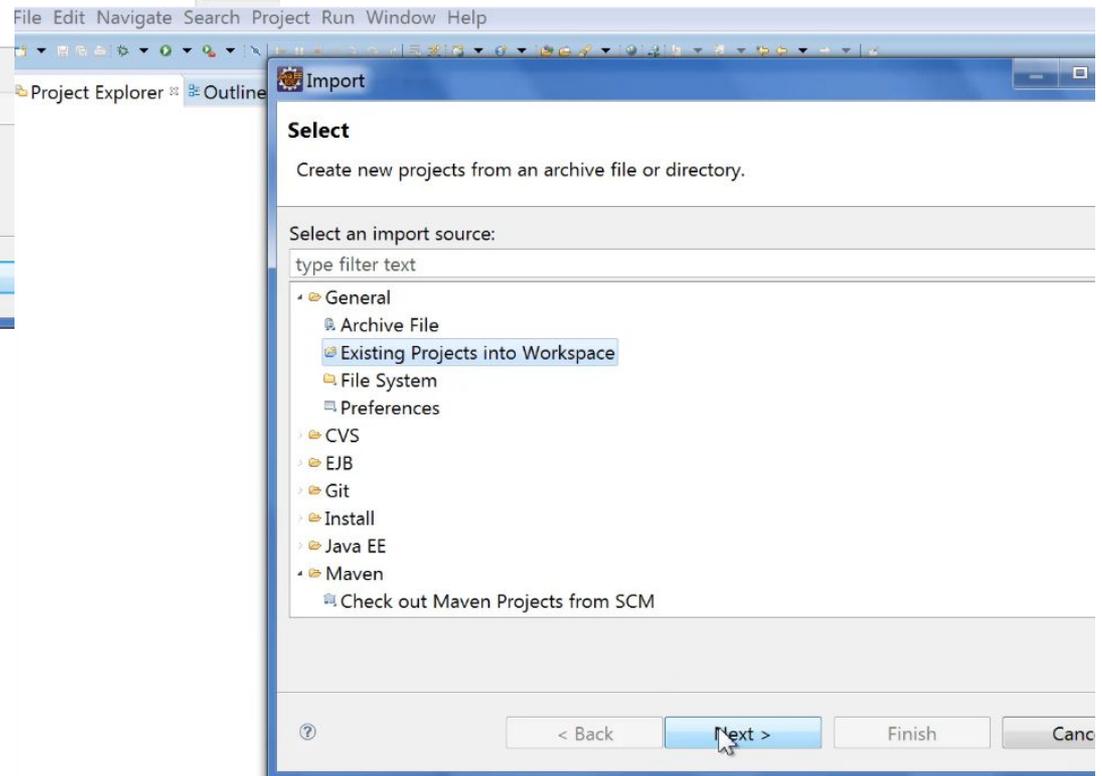
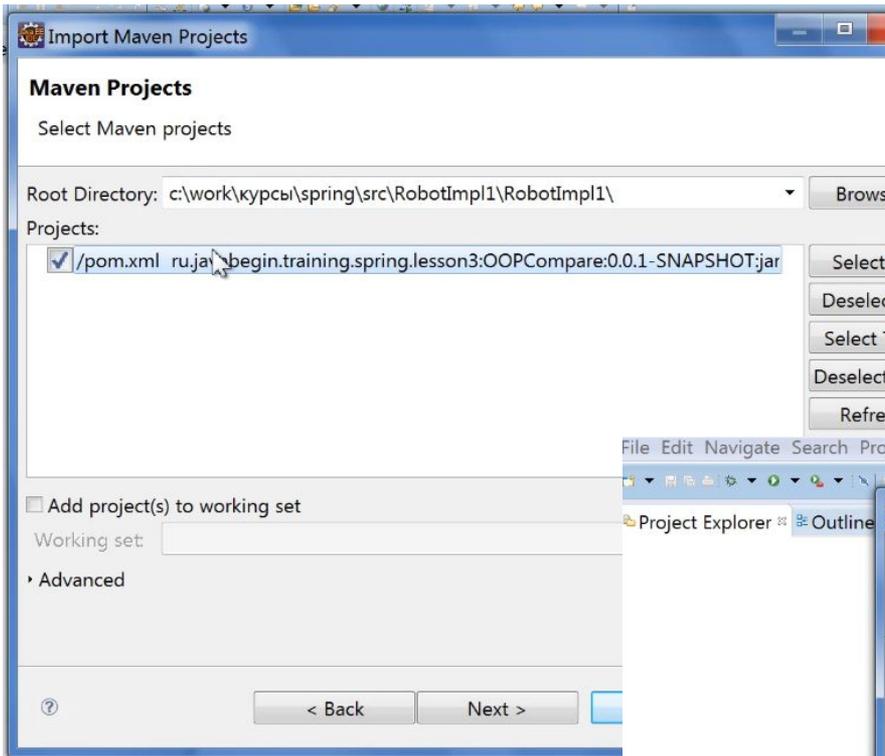
<dependencies>
  <dependency>
    <groupId>junit</groupId>
```

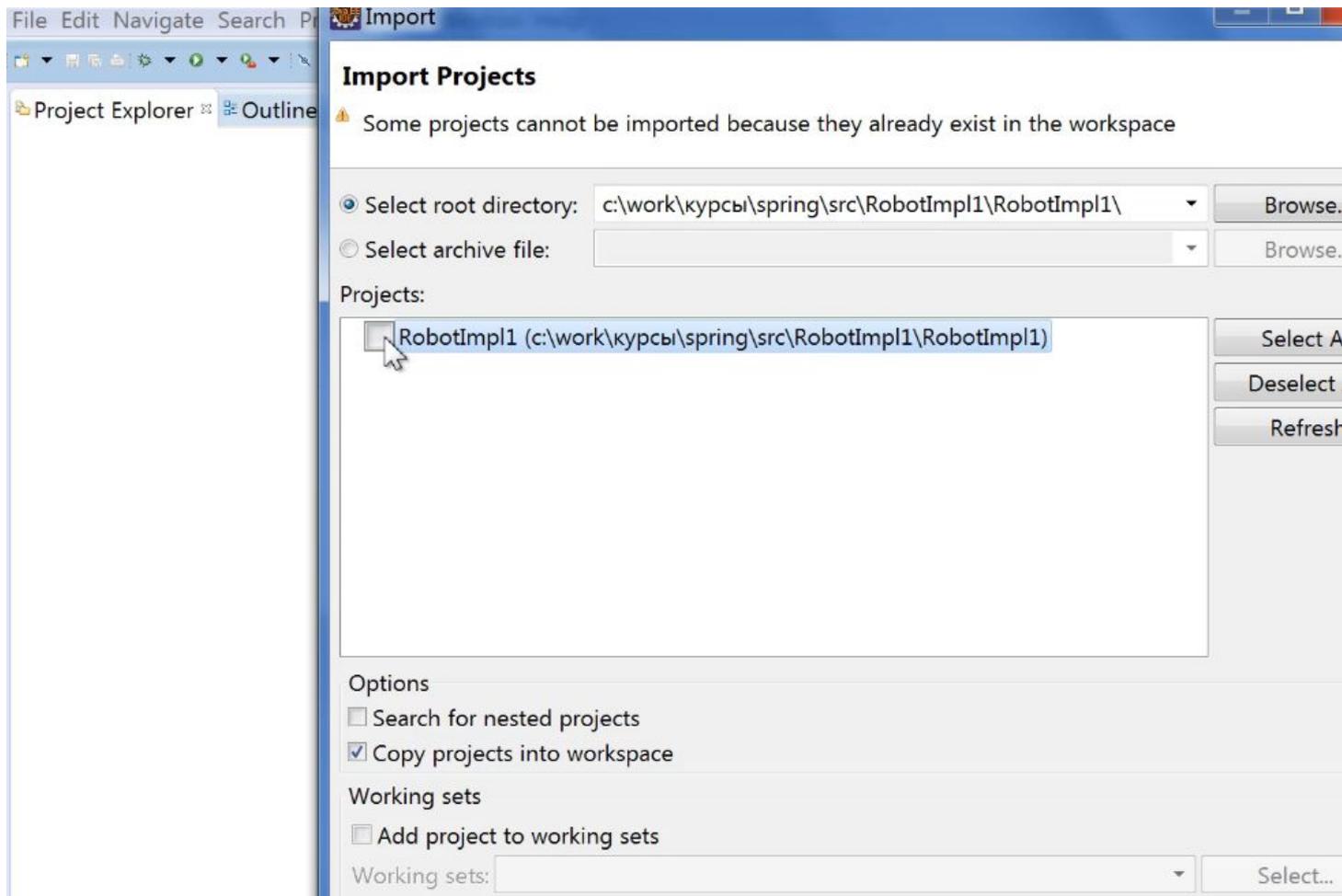
Markers Properties Servers Data Source Explorer Snippets Console Progress

Updating indexes

Updating index central|http://repo.maven.apache.org/maven2:







RobotImpl1

- New
- Go Into
- Show In Alt+Shift+W
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path
- Refactor Alt+Shift+T
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Projects
- Validate
- Show in Remote Systems view
- Profile As
- Debug As
- Run As
- Team
- Compare With

- 1 Java Applet
- 2 Java Application
- 3 JUnit Test
- 4 Maven build
- 5 Maven build...
- 6 Maven clean
- 7 Maven generate-sources
- 8 Maven install
- 9 Maven test
- Run Configurations...

Select Java Application

Select type (? = any character, * = any String, T = ...)

Matching items:

- RobotConstructor - ru.javabegin.training.spr
- TestRunner - junit.awtui

ru.javabegin.training.spring.start

OK Cancel

Quick Acces Java

```
Thinking about Sony...  
Caught from Sony!!  
Go to Toshiba!
```