Аспектно - ориентированное программирование

Лекция №3

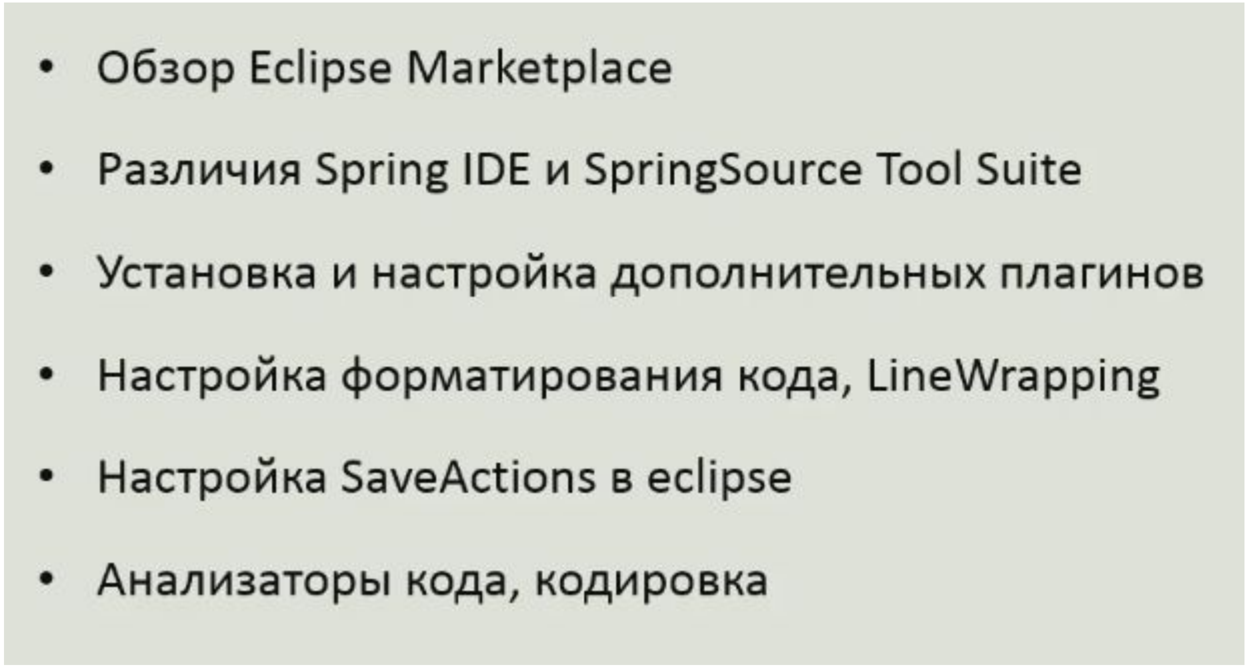
Антонов В.В.

Основные принципы ООП.

- ***1. Инкапсуляция - это объединение в единое целое данных и алгоритмов обработки этих данных***. В рамках ООП данные называются ***полями объекта (свойствами)***, а алгоритмы - ***объектными методами*** или просто ***методами***.
- ***2. Наследование - есть свойство объектов порождать своих потомков***. Объект-потомок автоматически наследует от родителя все поля и методы, может дополнять объекты новыми полями и заменять (перекрывать) методы родителя или дополнять их.
- ***3. Полиморфизм - это свойство родственных объектов (т.е. объектов, имеющих одного общего родителя) решать схожие по смыслу проблемы разными способами***. В рамках ООП поведенческие свойства объекта определяются набором входящих в него методов. Изменяя алгоритм того или иного метода в потомках объекта, программист может придавать этим потомкам отсутствующие у родителя специфические свойства. Для изменения метода необходимо ***перекрыть его в потомке***, то есть ***объявить в потомке одноименный метод и реализовать в нем нужные действия***.

Объекты
Наследование
Полиморфизм

Плагины для Eclipse



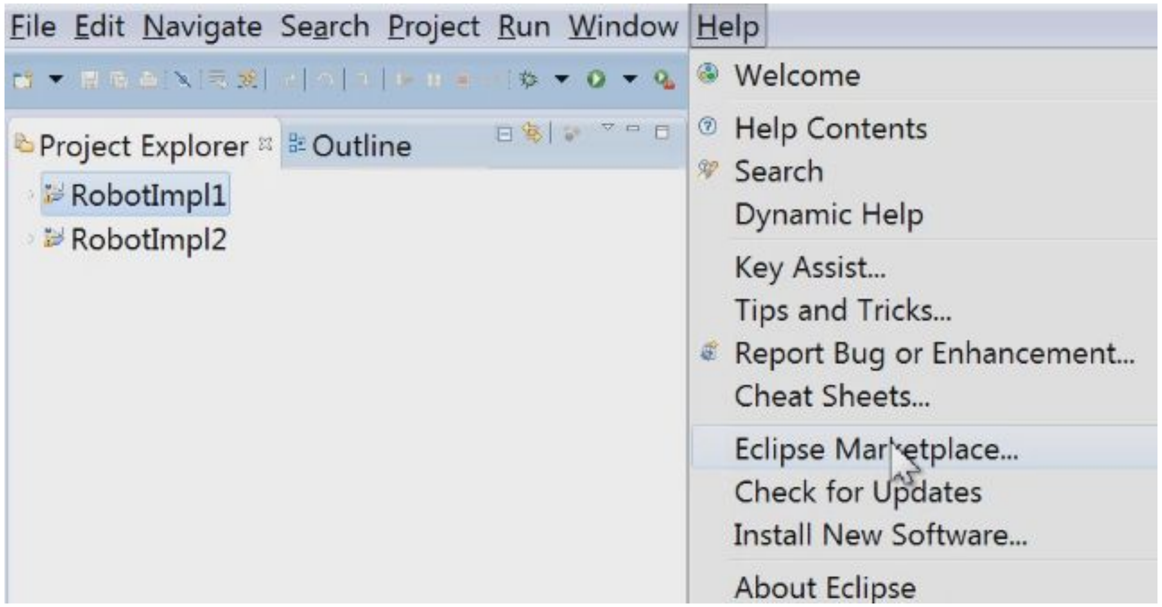
- Обзор Eclipse Marketplace
- Различия Spring IDE и SpringSource Tool Suite
- Установка и настройка дополнительных плагинов
- Настройка форматирования кода, LineWrapping
- Настройка SaveActions в eclipse
- Анализаторы кода, кодировка

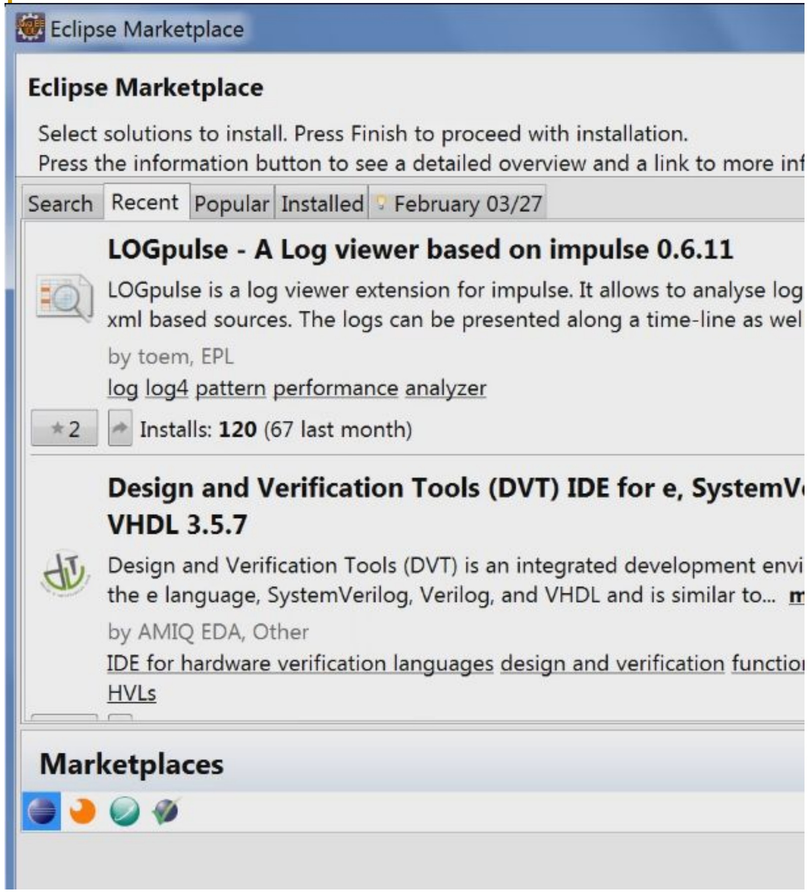
Eclipse Marketplace

- Eclipse Marketplace - Более удобный вариант поиска и установки дополнений eclipse

Два способа установки

- **Spring IDE** – набор основных плагинов
- **Spring Tool Suite** – полный пакет (включает также сам eclipse и все плагины Spring IDE)




Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more inf

Search | Recent | Popular | Installed | February 03/27

LOGpulse - A Log viewer based on impulse 0.6.11

LOGpulse is a log viewer extension for impulse. It allows to analyse log
xml based sources. The logs can be presented along a time-line as wel

by toem, EPL

log log4 pattern performance analyzer

★2 Installs: **120** (67 last month)

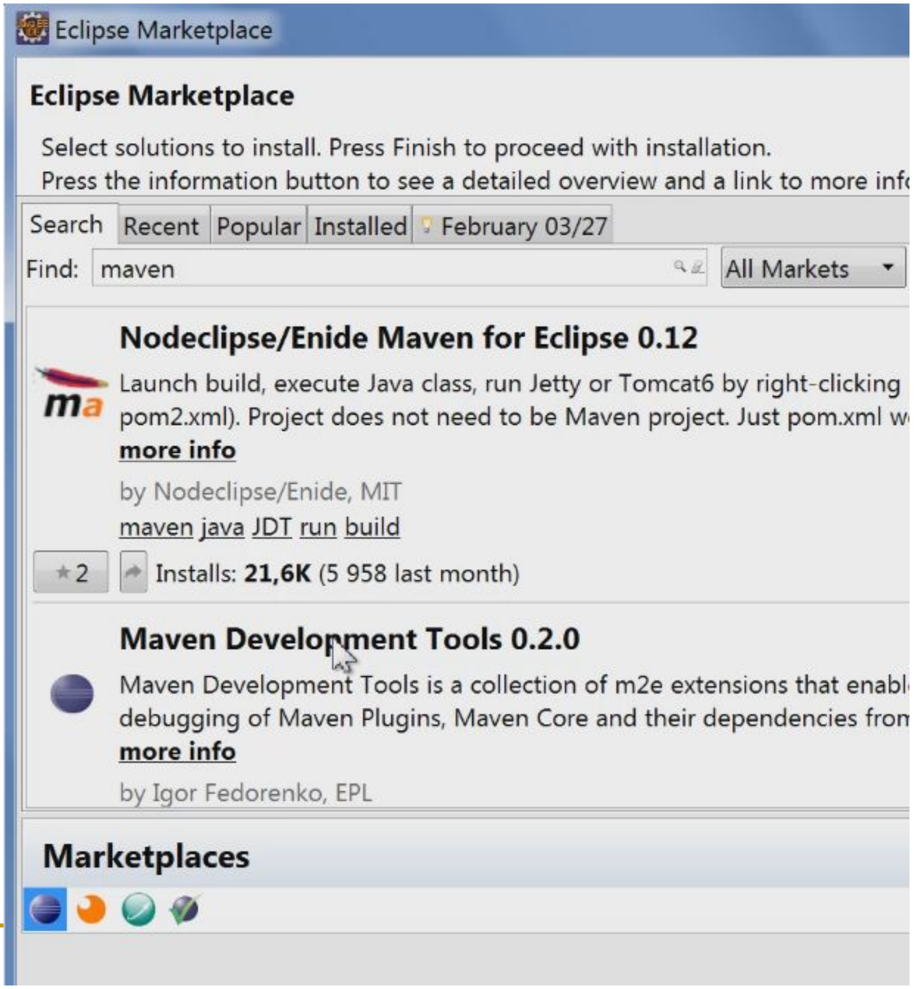
**Design and Verification Tools (DVT) IDE for e, SystemV
VHDL 3.5.7**

Design and Verification Tools (DVT) is an integrated development envi
the e language, SystemVerilog, Verilog, and VHDL and is similar to... m

by AMIQ EDA, Other

IDE for hardware verification languages design and verification functio
HVLs

Marketplaces


Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more inf

Search | Recent | Popular | Installed | February 03/27

Find: maven   All Markets

Nodeclipse/Enide Maven for Eclipse 0.12

Launch build, execute Java class, run Jetty or Tomcat6 by right-clicking
pom2.xml). Project does not need to be Maven project. Just pom.xml w
more info

by Nodeclipse/Enide, MIT

maven java JDT run build

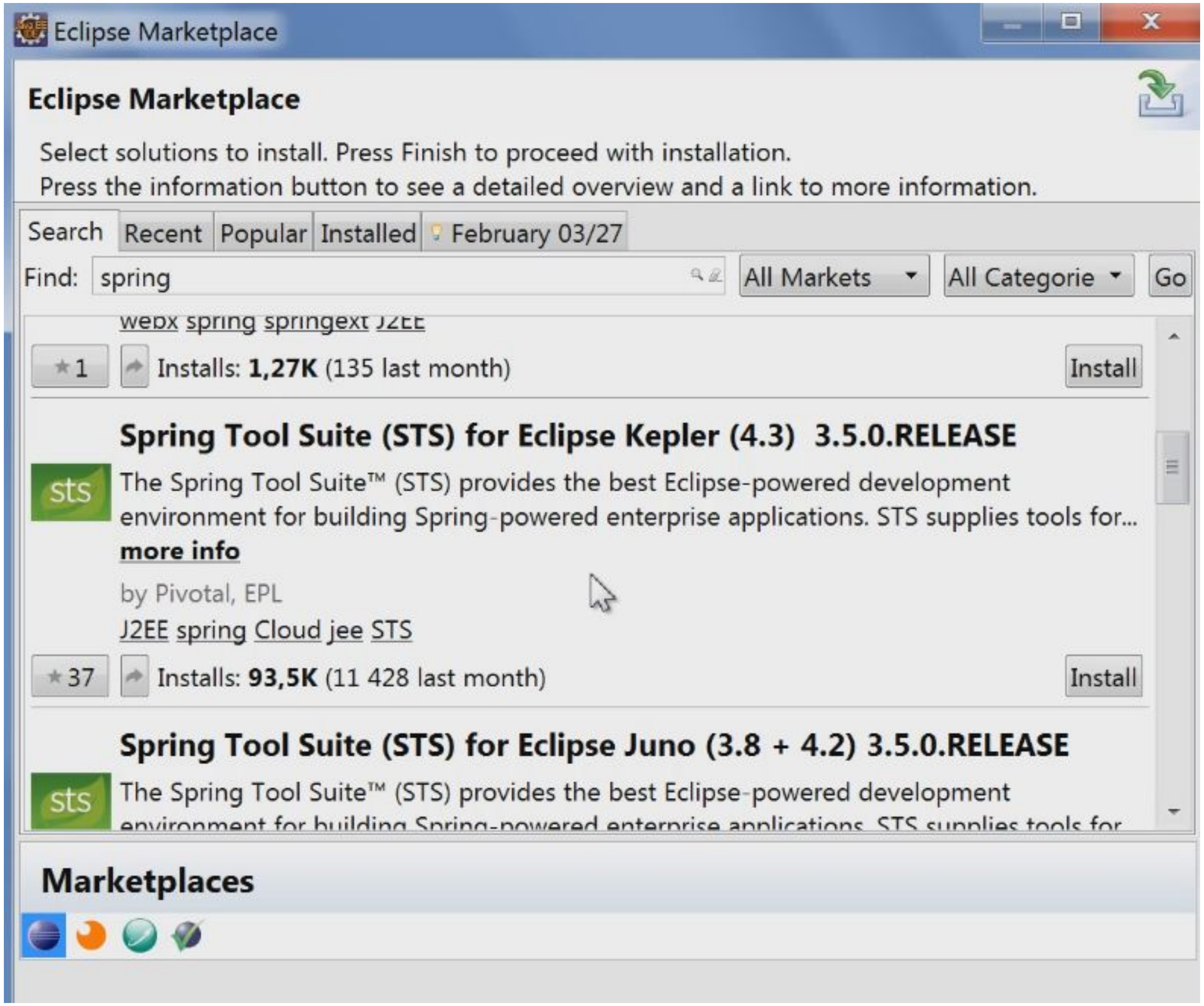
★2 Installs: **21,6K** (5 958 last month)

Maven Development Tools 0.2.0

Maven Development Tools is a collection of m2e extensions that enabl
debugging of Maven Plugins, Maven Core and their dependencies fron
more info

by Igor Fedorenko, EPL

Marketplaces


Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

Find: spring

webx spring springext J2EE

★1 Installs: **1,27K** (135 last month) Install

Spring Tool Suite (STS) for Eclipse Kepler (4.3) 3.5.0.RELEASE

The Spring Tool Suite™ (STS) provides the best Eclipse-powered development environment for building Spring-powered enterprise applications. STS supplies tools for...
more info

by Pivotal, EPL

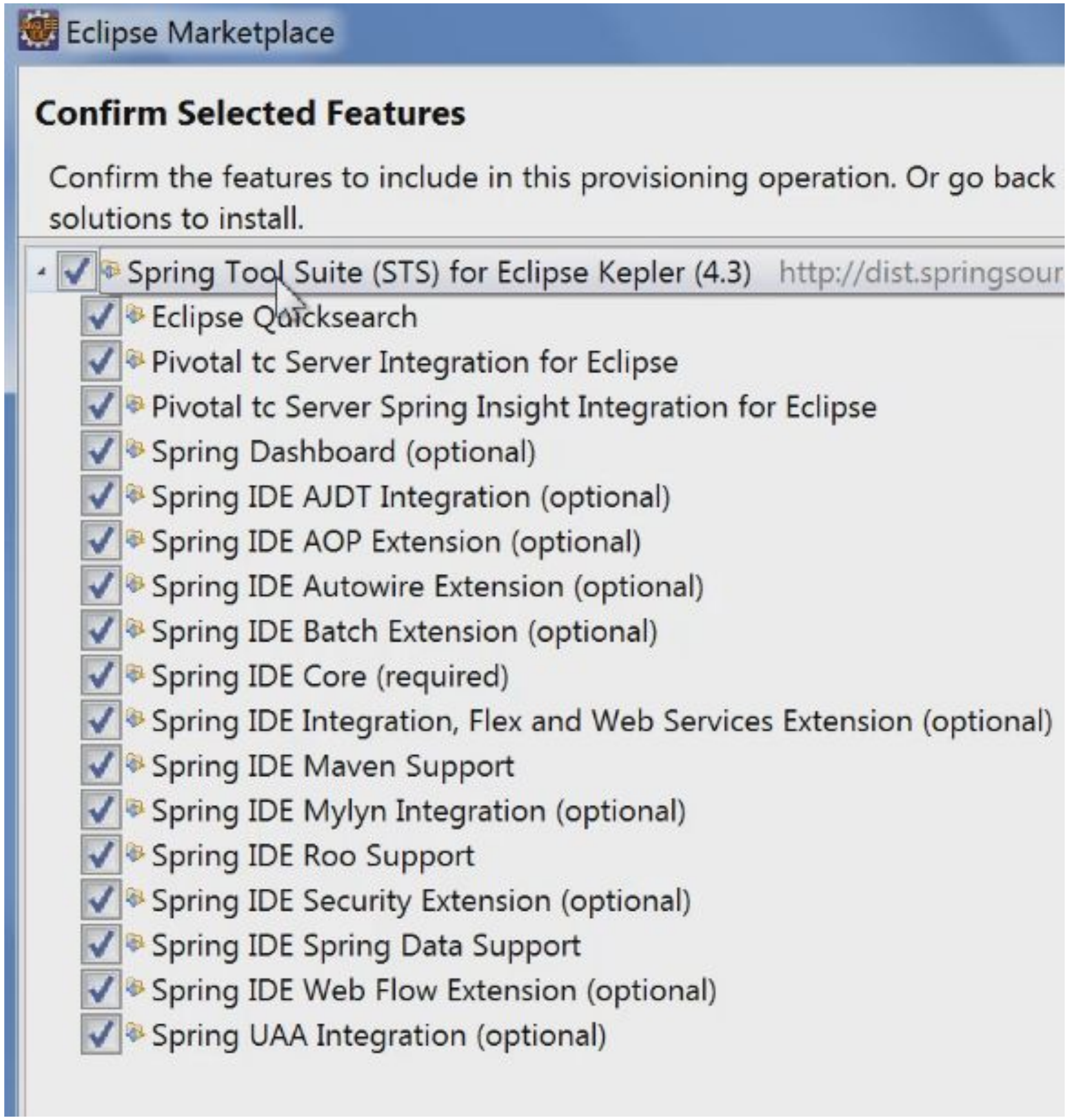
J2EE spring Cloud jee STS

★37 Installs: **93,5K** (11 428 last month) Install

Spring Tool Suite (STS) for Eclipse Juno (3.8 + 4.2) 3.5.0.RELEASE

The Spring Tool Suite™ (STS) provides the best Eclipse-powered development environment for building Spring-powered enterprise applications. STS supplies tools for

Marketplaces



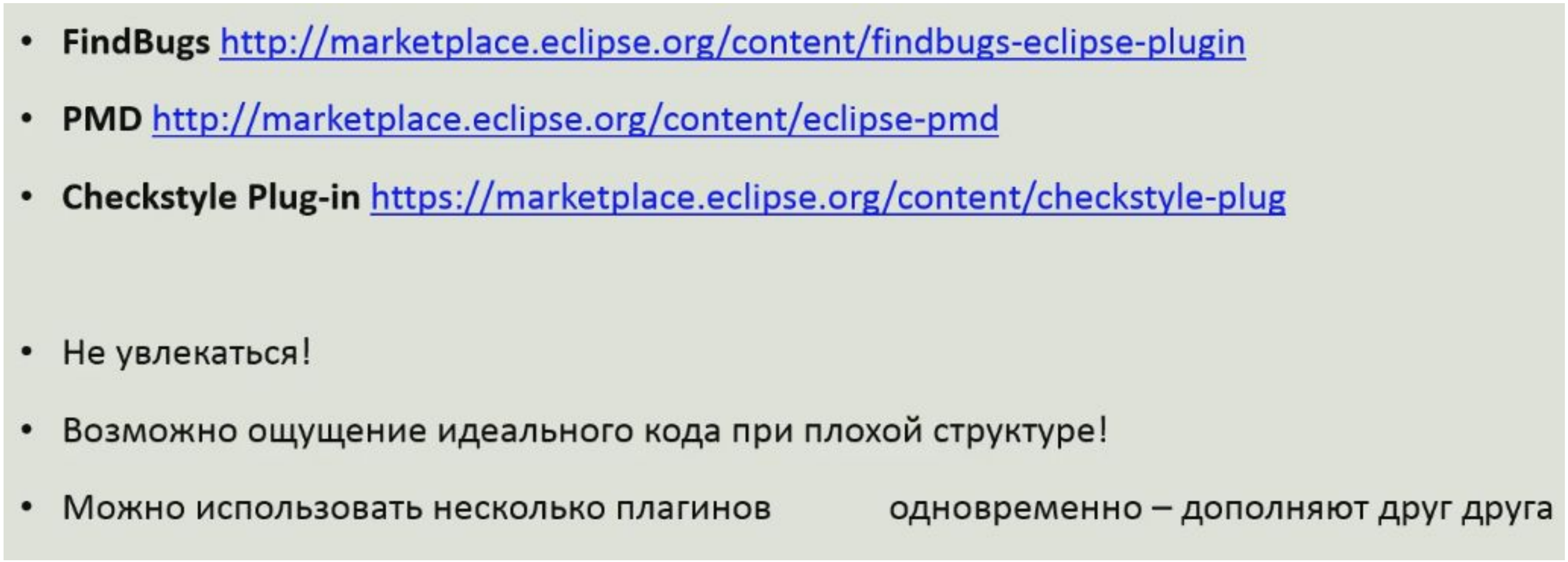
Eclipse Marketplace

Confirm Selected Features

Confirm the features to include in this provisioning operation. Or go back
solutions to install.

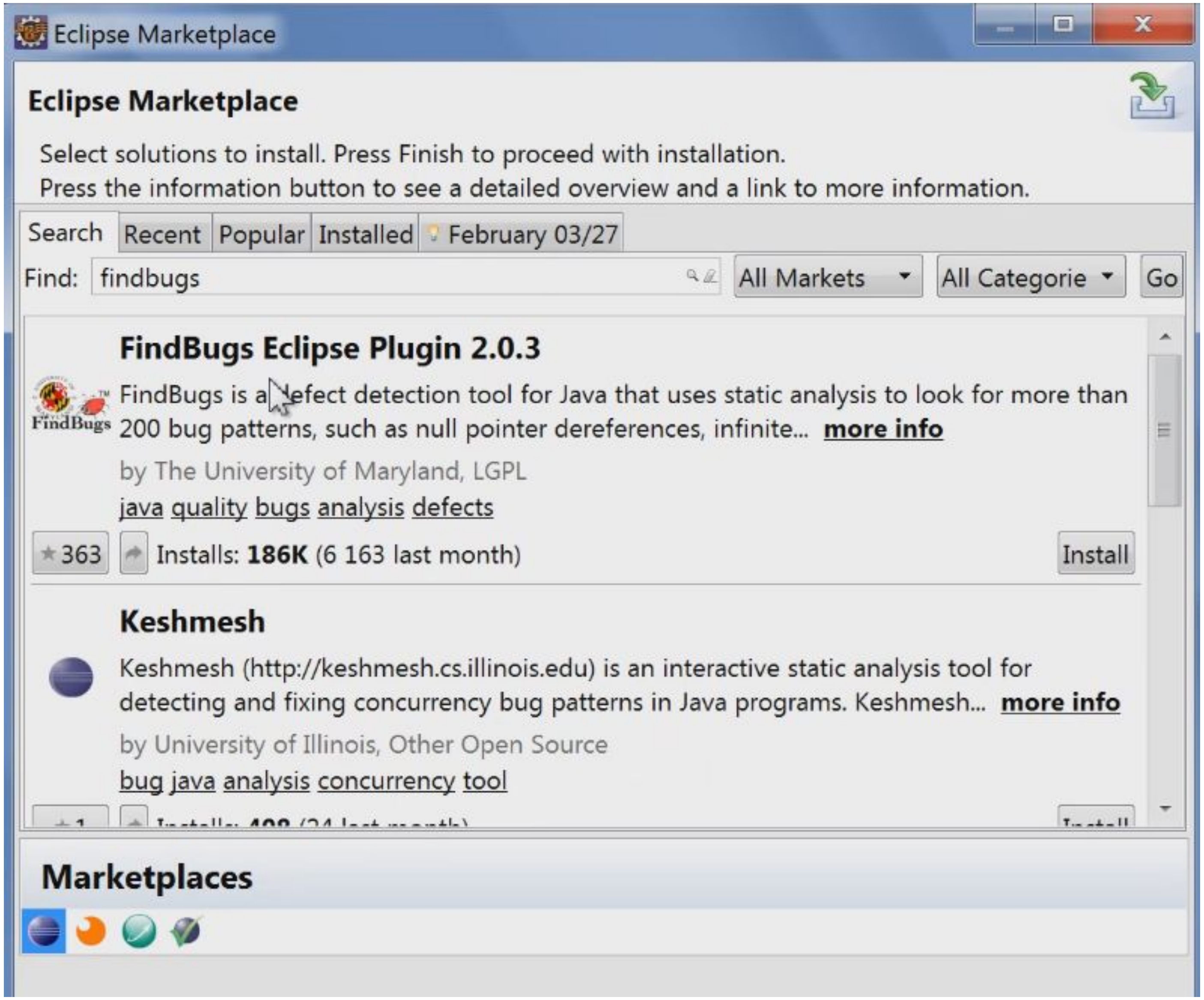
- ✓ Spring Tool Suite (STS) for Eclipse Kepler (4.3) http://dist.springsour
  - ✓ Eclipse Quicksearch
  - ✓ Pivotal tc Server Integration for Eclipse
  - ✓ Pivotal tc Server Spring Insight Integration for Eclipse
  - ✓ Spring Dashboard (optional)
  - ✓ Spring IDE AJDT Integration (optional)
  - ✓ Spring IDE AOP Extension (optional)
  - ✓ Spring IDE Autowire Extension (optional)
  - ✓ Spring IDE Batch Extension (optional)
  - ✓ Spring IDE Core (required)
  - ✓ Spring IDE Integration, Flex and Web Services Extension (optional)
  - ✓ Spring IDE Maven Support
  - ✓ Spring IDE Mylyn Integration (optional)
  - ✓ Spring IDE Roo Support
  - ✓ Spring IDE Security Extension (optional)
  - ✓ Spring IDE Spring Data Support
  - ✓ Spring IDE Web Flow Extension (optional)
  - ✓ Spring UAA Integration (optional)

Анализаторы кода



- **FindBugs** http://marketplace.eclipse.org/content/findbugs-eclipse-plugin
- **PMD** http://marketplace.eclipse.org/content/eclipse-pmd
- **Checkstyle Plug-in** https://marketplace.eclipse.org/content/checkstyle-plug

- Не увлекаться!
- Возможно ощущение идеального кода при плохой структуре!
- Можно использовать несколько плагинов одновременно – дополняют друг друга


Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

FindBugs Eclipse Plugin 2.0.3

FindBugs is a defect detection tool for Java that uses static analysis to look for more than 200 bug patterns, such as null pointer dereferences, infinite... **more info**

by The University of Maryland, LGPL

java quality bugs analysis defects

363 Installs: **186K** (6 163 last month) Install

Keshmesh

Keshmesh (http://keshmesh.cs.illinois.edu) is an interactive static analysis tool for detecting and fixing concurrency bug patterns in Java programs. Keshmesh... **more info**

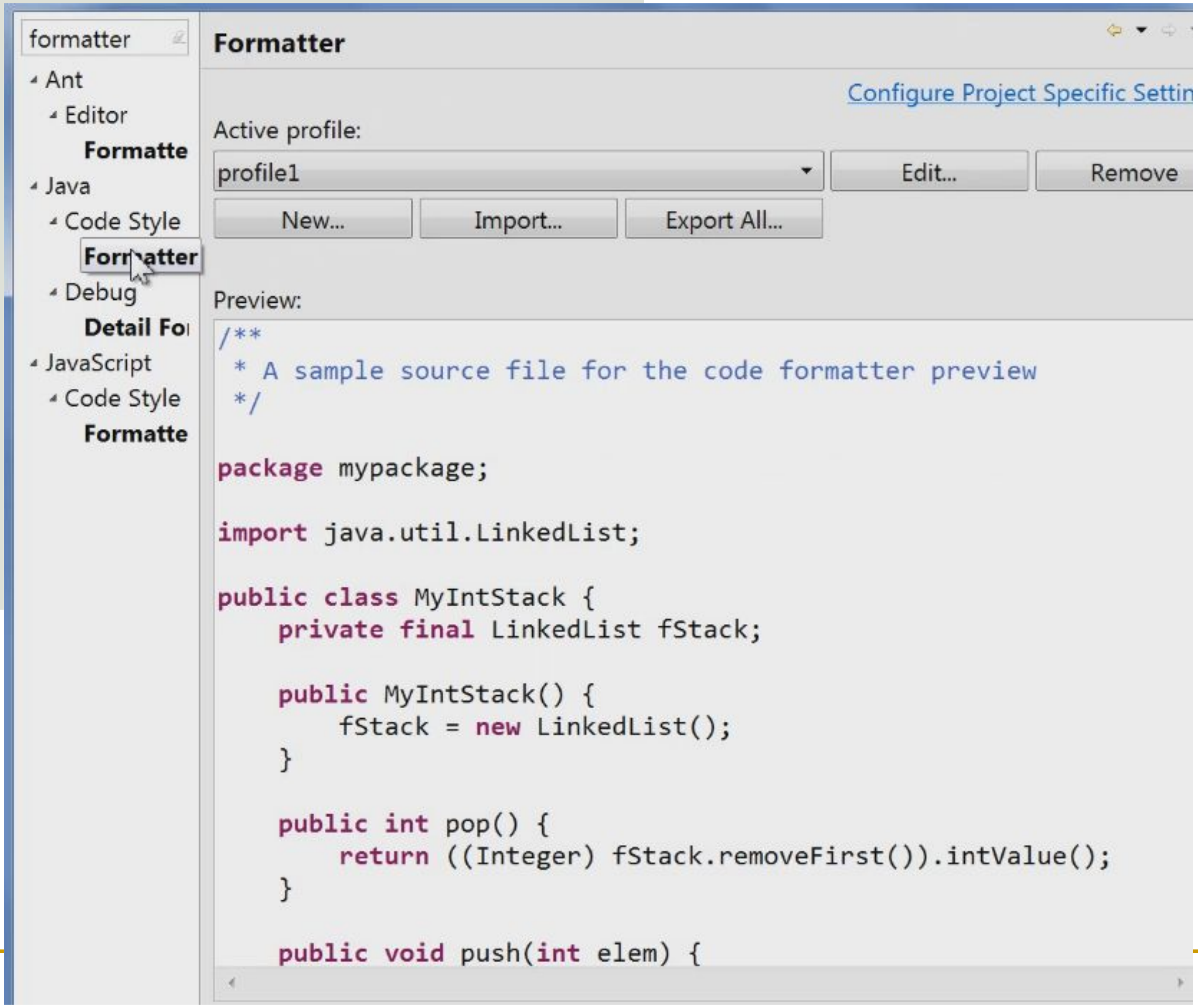
by University of Illinois, Other Open Source

bug java analysis concurrency tool

Marketplaces

Форматирование кода

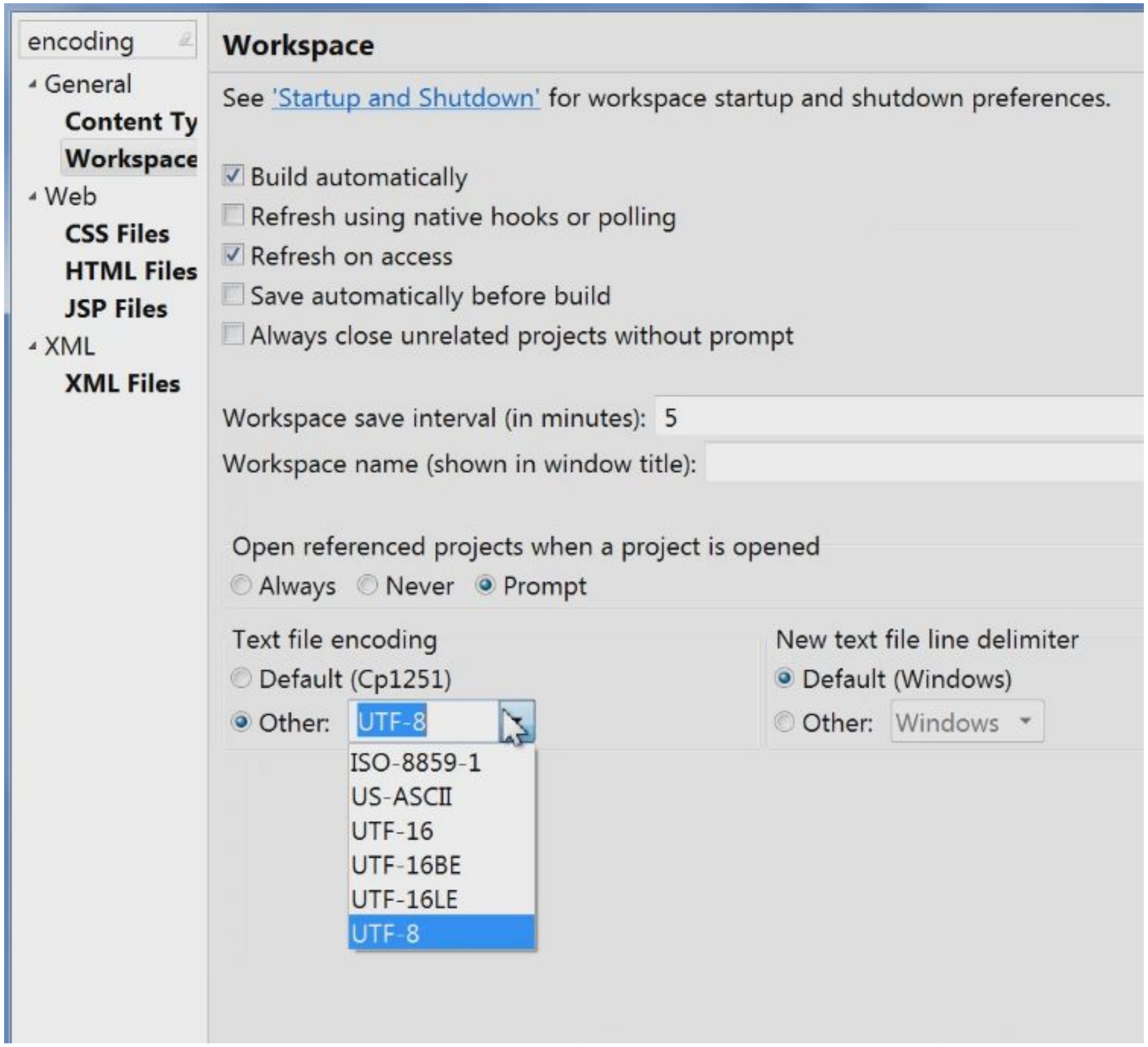
- Настроить максимальную длину LineWrapping



Active profile: profile1

Preview:

```
/**
 * A sample source file for the code formatter preview
 */

package mypackage;

import java.util.LinkedList;

public class MyIntStack {
    private final LinkedList fStack;

    public MyIntStack() {
        fStack = new LinkedList();
    }

    public int pop() {
        return ((Integer) fStack.removeFirst()).intValue();
    }

    public void push(int elem) {
```



encoding

- General
  - Content Ty
  - Workspace
- Web
  - CSS Files
  - HTML Files
  - JSP Files
- XML
  - XML Files

Workspace

See 'Startup and Shutdown' for workspace startup and preferences.

- [✓] Build automatically
- [ ] Refresh using native hooks or polling
- [✓] Refresh on access
- [ ] Save automatically before build
- [ ] Always close unrelated projects without prompt

Workspace save interval (in minutes): 5

Workspace name (shown in window title):

Open referenced projects when a project is opened

Always Never [✓ Prompt]

Text file encoding

Default (Cp1251)

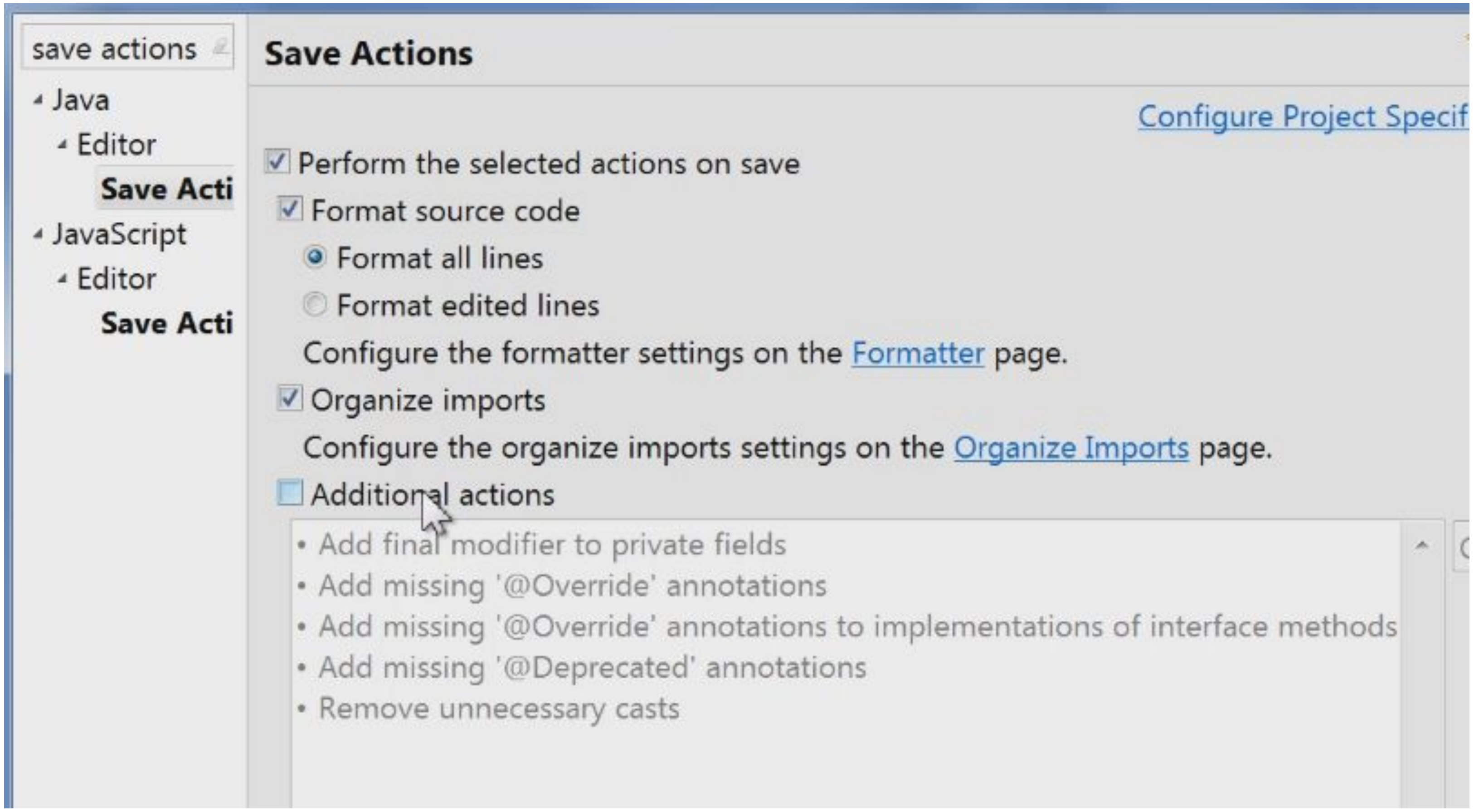
[✓ Other:] UTF-8

ISO-8859-1
US-ASCII
UTF-16
UTF-16BE
UTF-16LE
UTF-8

New text file line delimiter

[✓ Default (Windows)]

Other: Windows



Save Actions

[✓] Perform the selected actions on save

[✓] Format source code

[● Format all lines]

○ Format edited lines

Configure the formatter settings on the Formatter page.

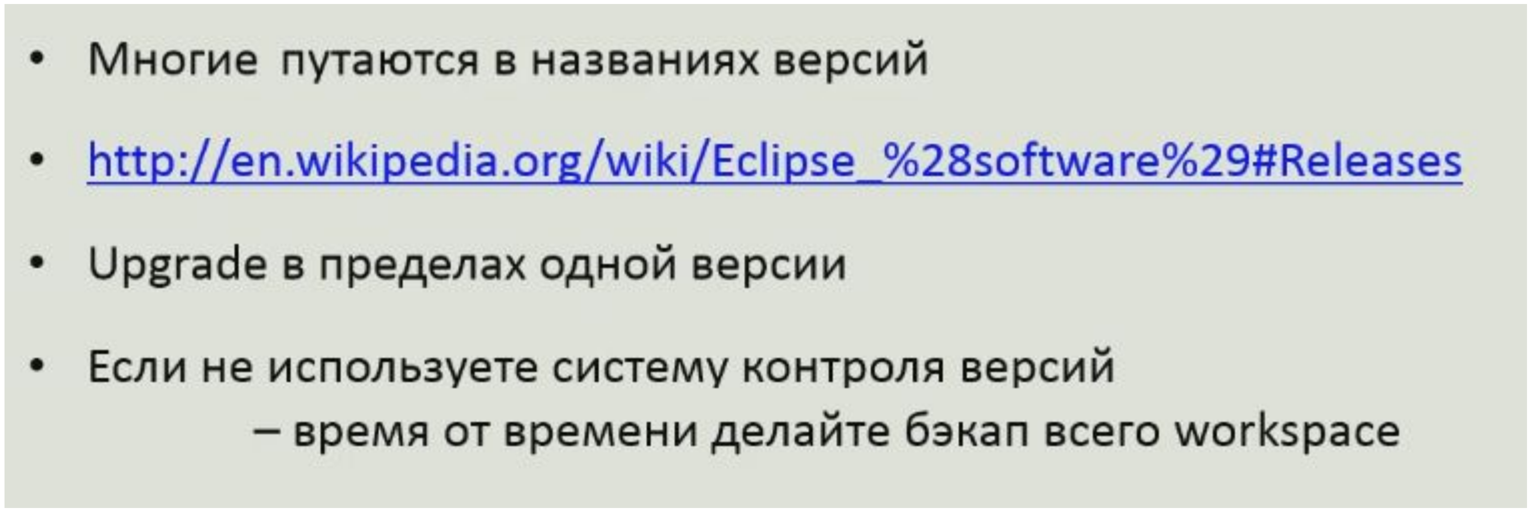
[✓] Organize imports

Configure the organize imports settings on the Organize Imports page.

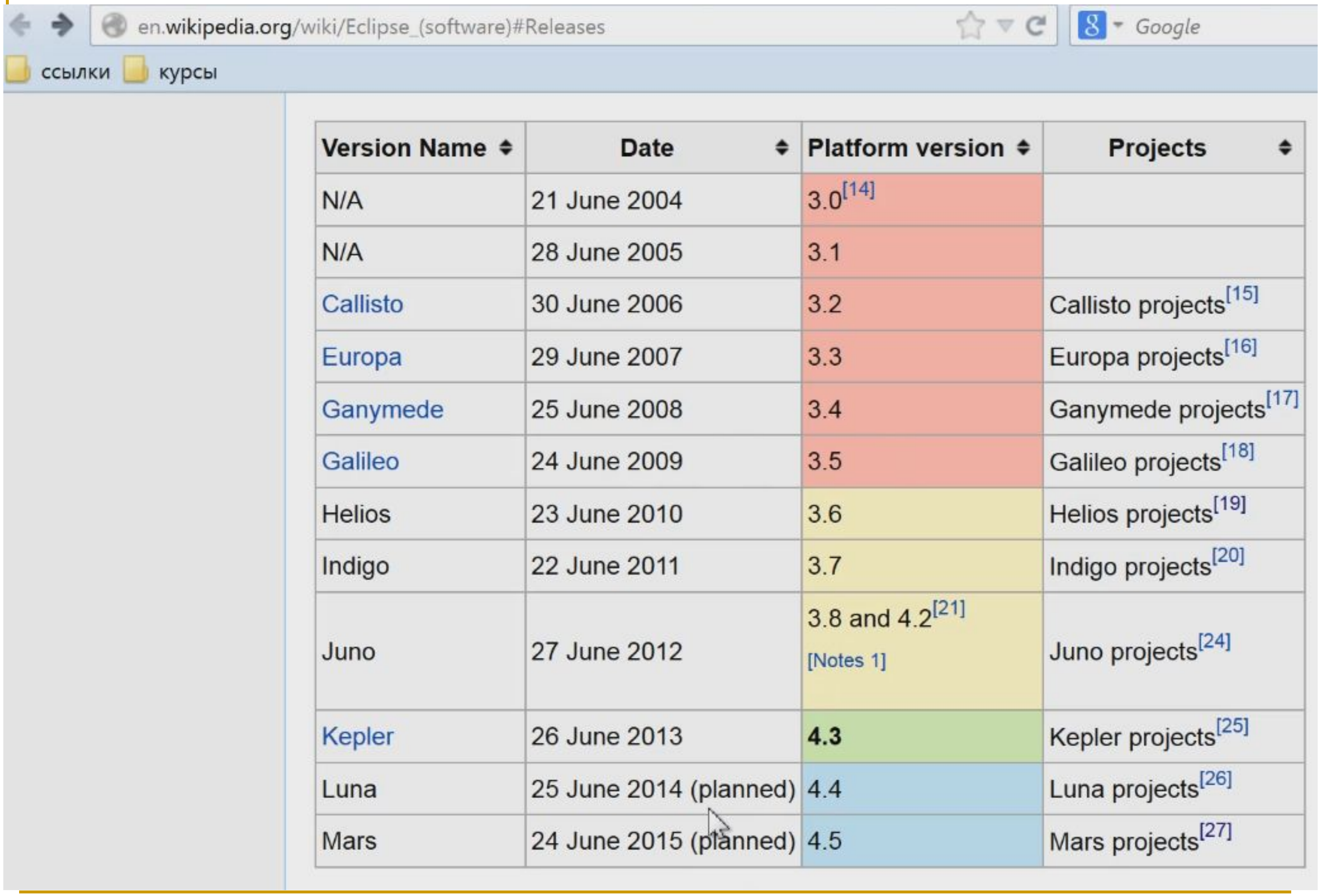
[ ] Additional actions

- Add final modifier to private fields
- Add missing '@Override' annotations
- Add missing '@Override' annotations to implementations of interface methods
- Add missing '@Deprecated' annotations
- Remove unnecessary casts

Версии eclipse

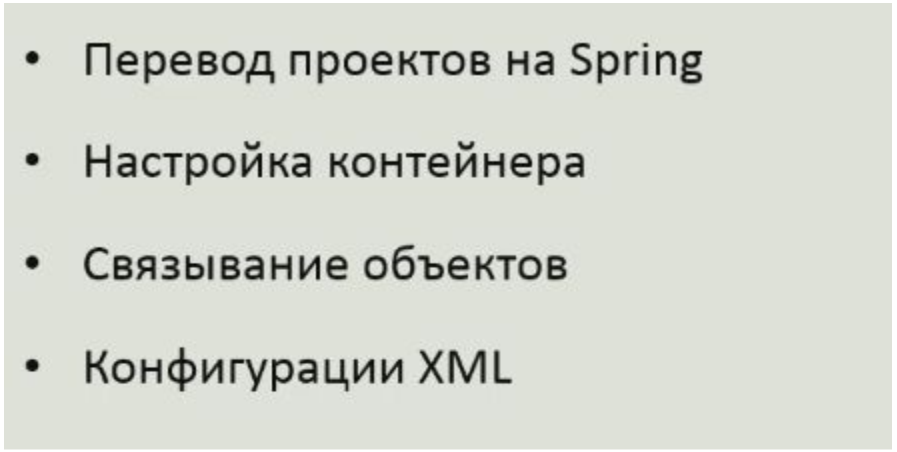


- Многие путаются в названиях версий
- http://en.wikipedia.org/wiki/Eclipse_%28software%29#Releases
- Upgrade в пределах одной версии
- Если не используете систему контроля версий
  – время от времени делайте бэкап всего workspace



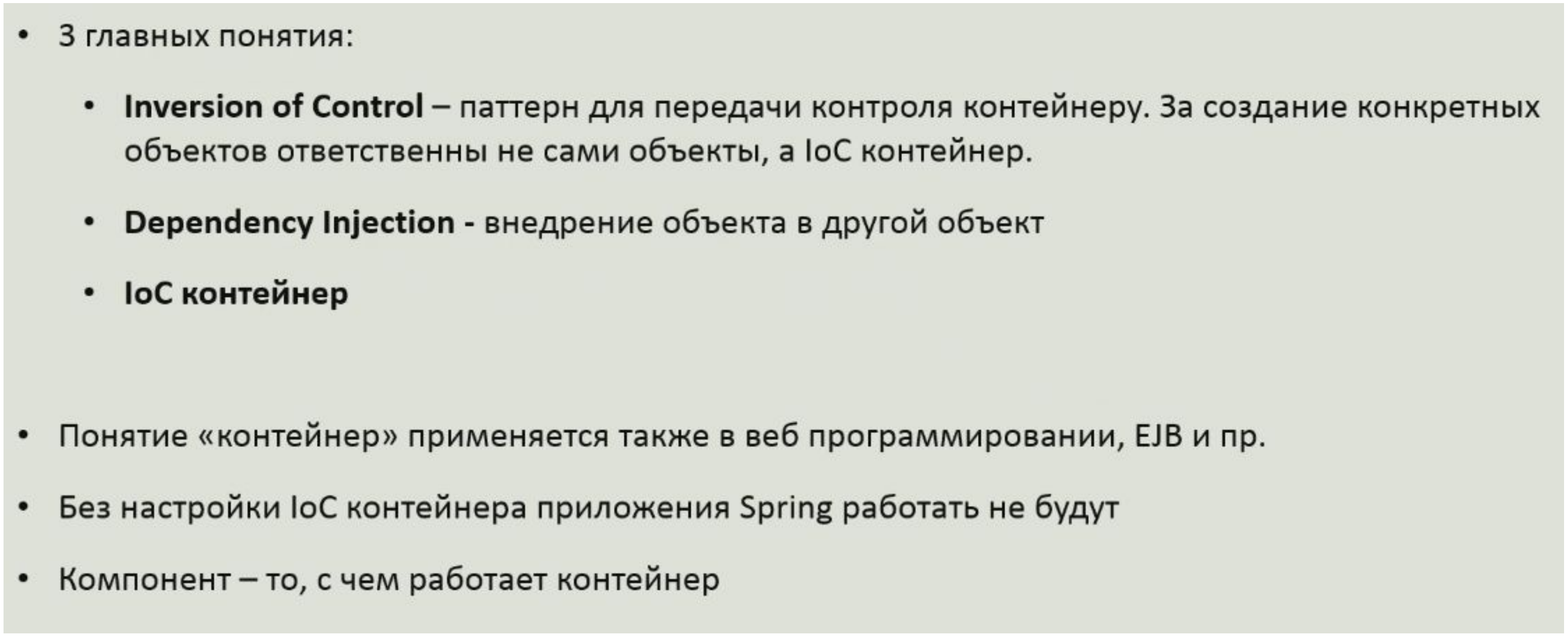
Version Name	Date	Platform version	Projects
N/A	21 June 2004	3.0[14]	
N/A	28 June 2005	3.1	
Callisto	30 June 2006	3.2	Callisto projects[15]
Europa	29 June 2007	3.3	Europa projects[16]
Ganymede	25 June 2008	3.4	Ganymede projects[17]
Galileo	24 June 2009	3.5	Galileo projects[18]
Helios	23 June 2010	3.6	Helios projects[19]
Indigo	22 June 2011	3.7	Indigo projects[20]
Juno	27 June 2012	3.8 and 4.2[21] [Notes 1]	Juno projects[24]
Kepler	26 June 2013	**4.3**	Kepler projects[25]
Luna	25 June 2014 (planned)	4.4	Luna projects[26]
Mars	24 June 2015 (planned)	4.5	Mars projects[27]

Пример на eclipse



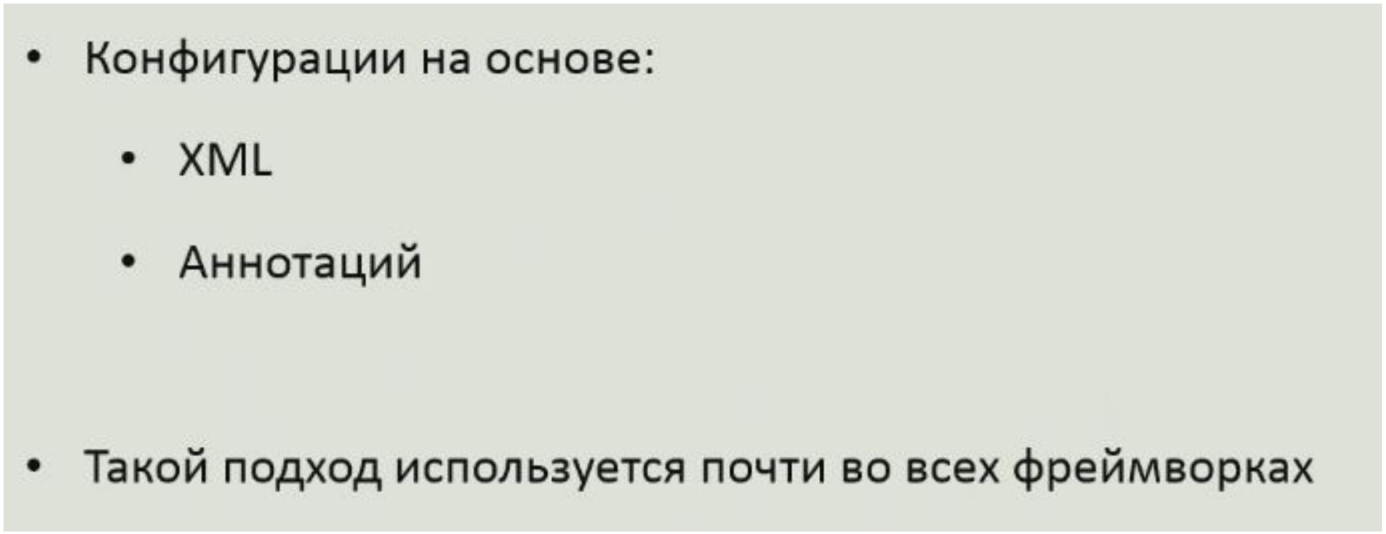
- Перевод проектов на Spring
- Настройка контейнера
- Связывание объектов
- Конфигурации XML

Понятия

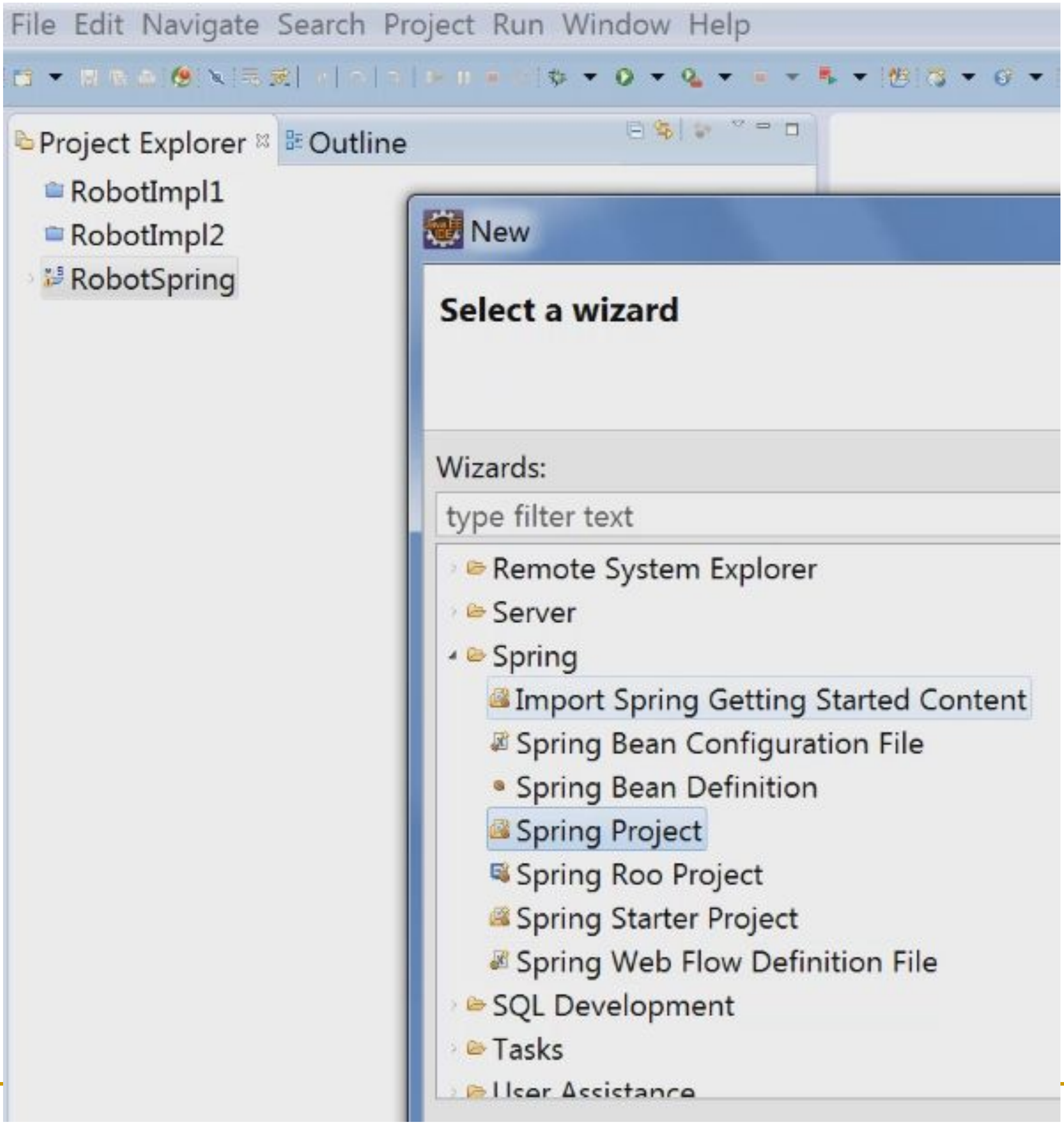


- 3 главных понятия:
  - **Inversion of Control** – паттерн для передачи контроля контейнеру. За создание конкретных объектов ответственны не сами объекты, а IoC контейнер.
  - **Dependency Injection -** внедрение объекта в другой объект
  - **IoC контейнер**
- Понятие «контейнер» применяется также в веб программировании, EJB и пр.
- Без настройки IoC контейнера приложения Spring работать не будут
- Компонент – то, с чем работает контейнер

Spring



- Конфигурации на основе:
  - XML
  - Аннотаций
- Такой подход используется почти во всех фреймворках

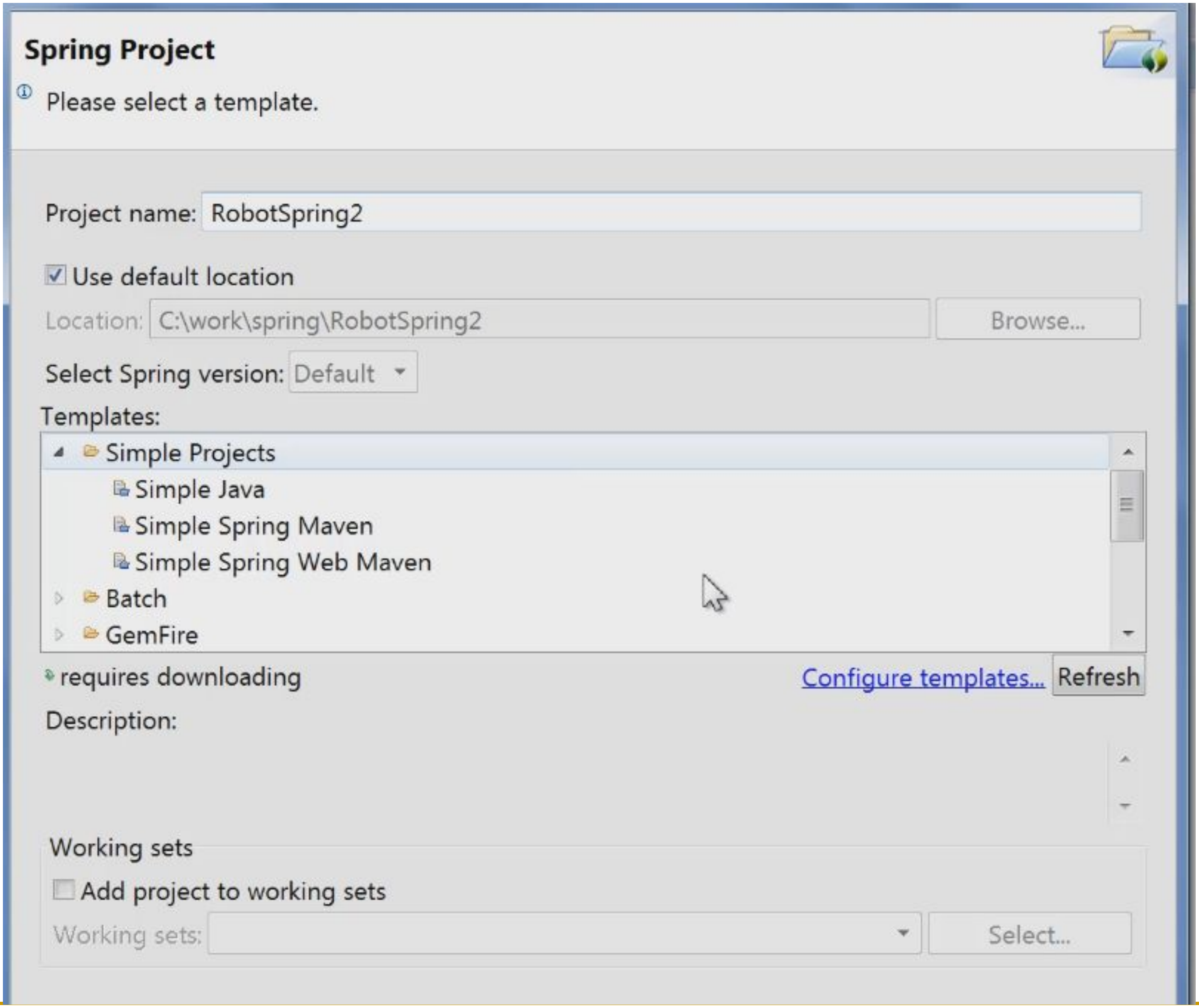


Select a wizard

Wizards:

type filter text

- Remote System Explorer
- Server
- Spring
  - Import Spring Getting Started Content
  - Spring Bean Configuration File
  - Spring Bean Definition
  - Spring Project
  - Spring Roo Project
  - Spring Starter Project
  - Spring Web Flow Definition File
- SQL Development
- Tasks
- User Assistance



Spring Project

Please select a template.

Project name: RobotSpring2

[✓] Use default location

Location: C:\work\spring\RobotSpring2

Select Spring version: Default

Templates:

- Simple Projects
  - Simple Java
  - Simple Spring Maven
  - Simple Spring Web Maven
- Batch
- GemFire

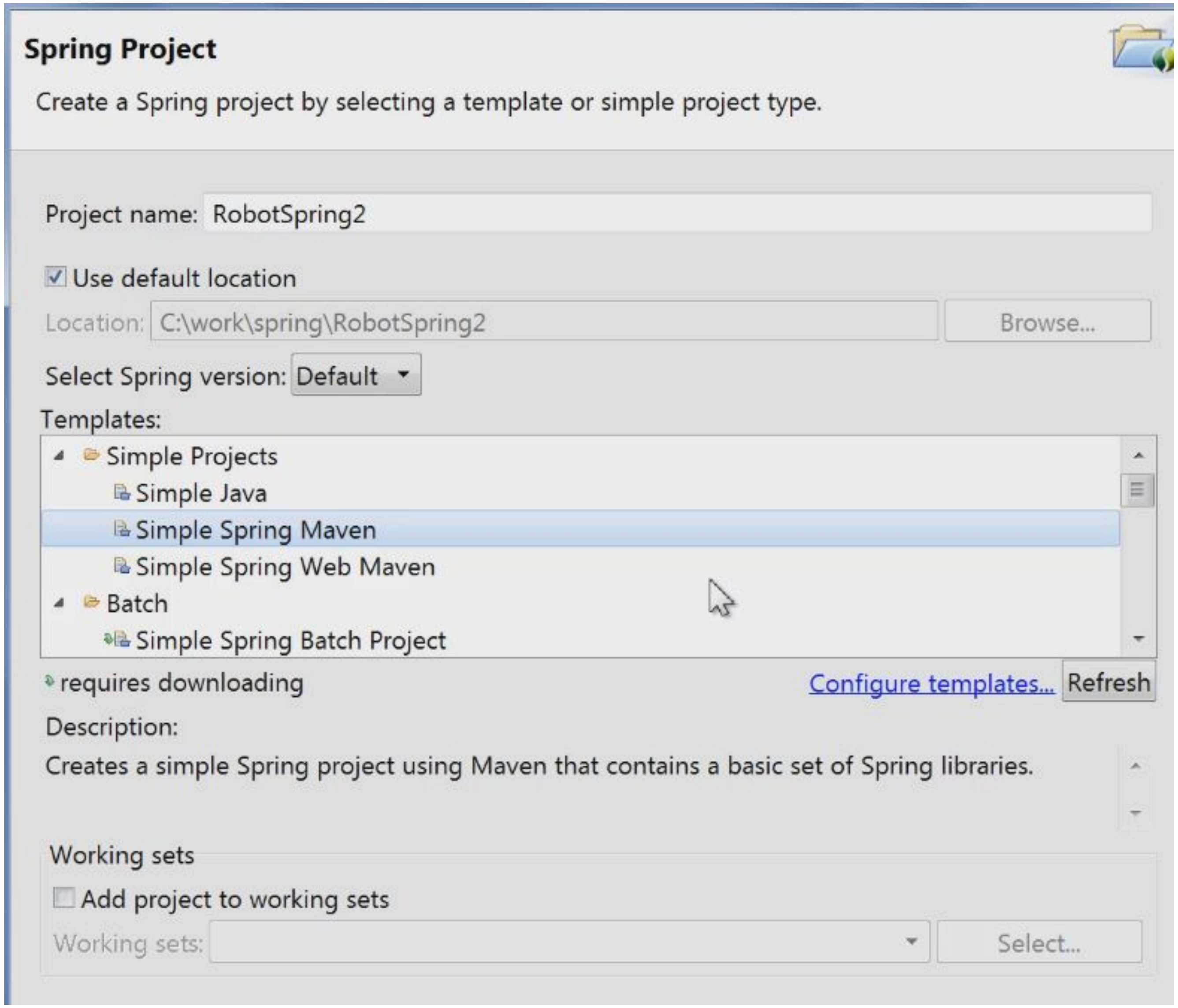
requires downloading

Description:

Working sets

[ ] Add project to working sets

Working sets:


Spring Project

Create a Spring project by selecting a template or simple project type.

Project name: RobotSpring2

☑ Use default location

Location: C:\work\spring\RobotSpring2

Select Spring version: Default

Templates:

- Simple Projects
  - Simple Java
  - Simple Spring Maven
  - Simple Spring Web Maven
- Batch
  - Simple Spring Batch Project

requires downloading

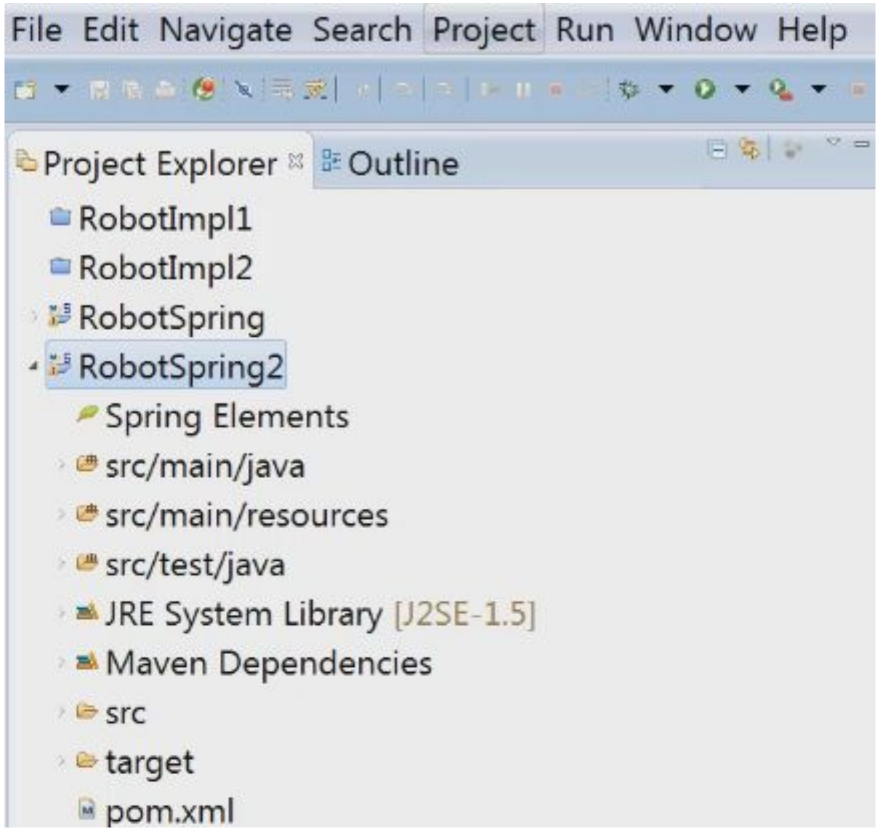
Description:

Creates a simple Spring project using Maven that contains a basic set of Spring libraries.

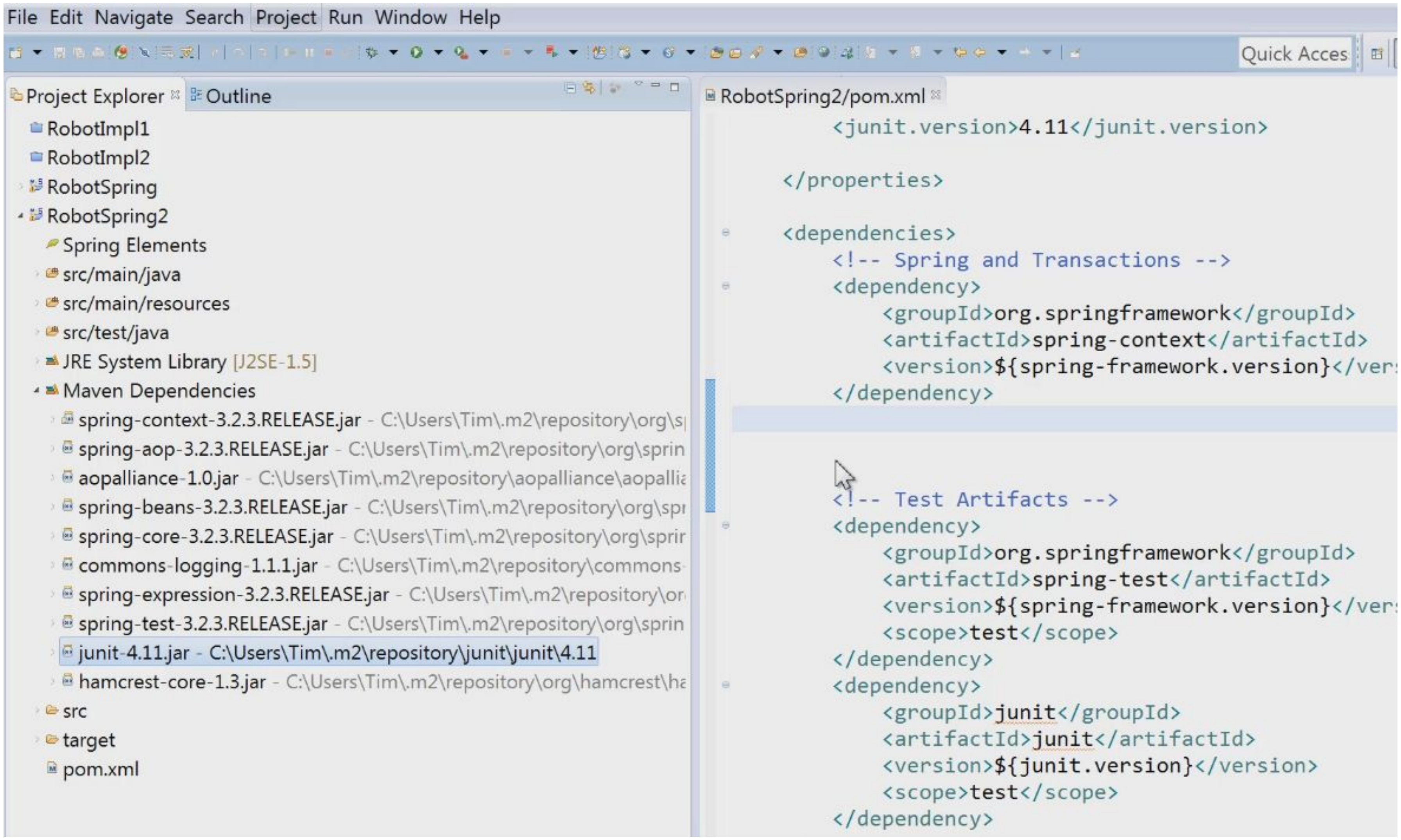
Working sets

☐ Add project to working sets

Working sets:

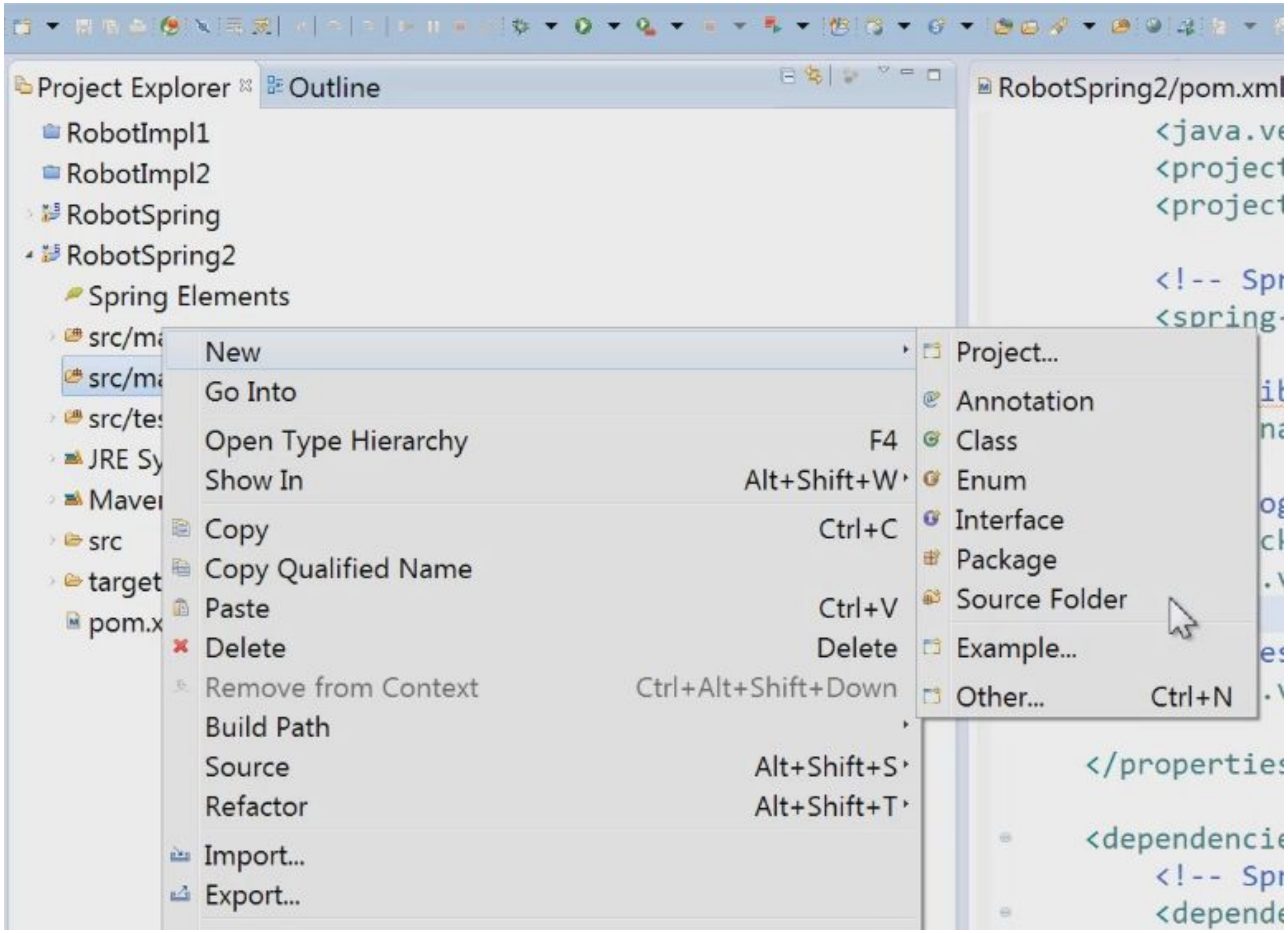


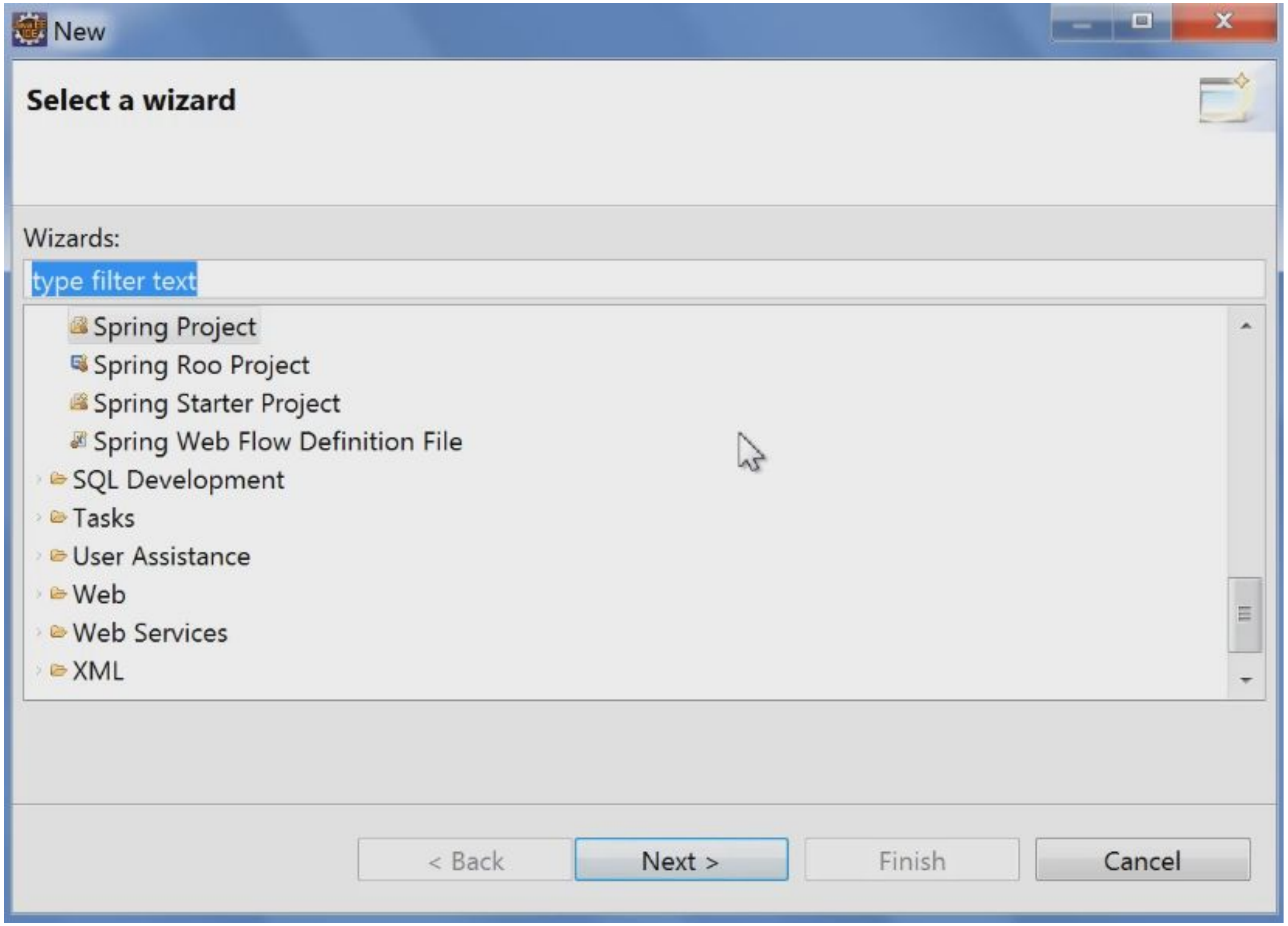
RobotImpl1
RobotImpl2
RobotSpring
RobotSpring2
Spring Elements
src/main/java
src/main/resources
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

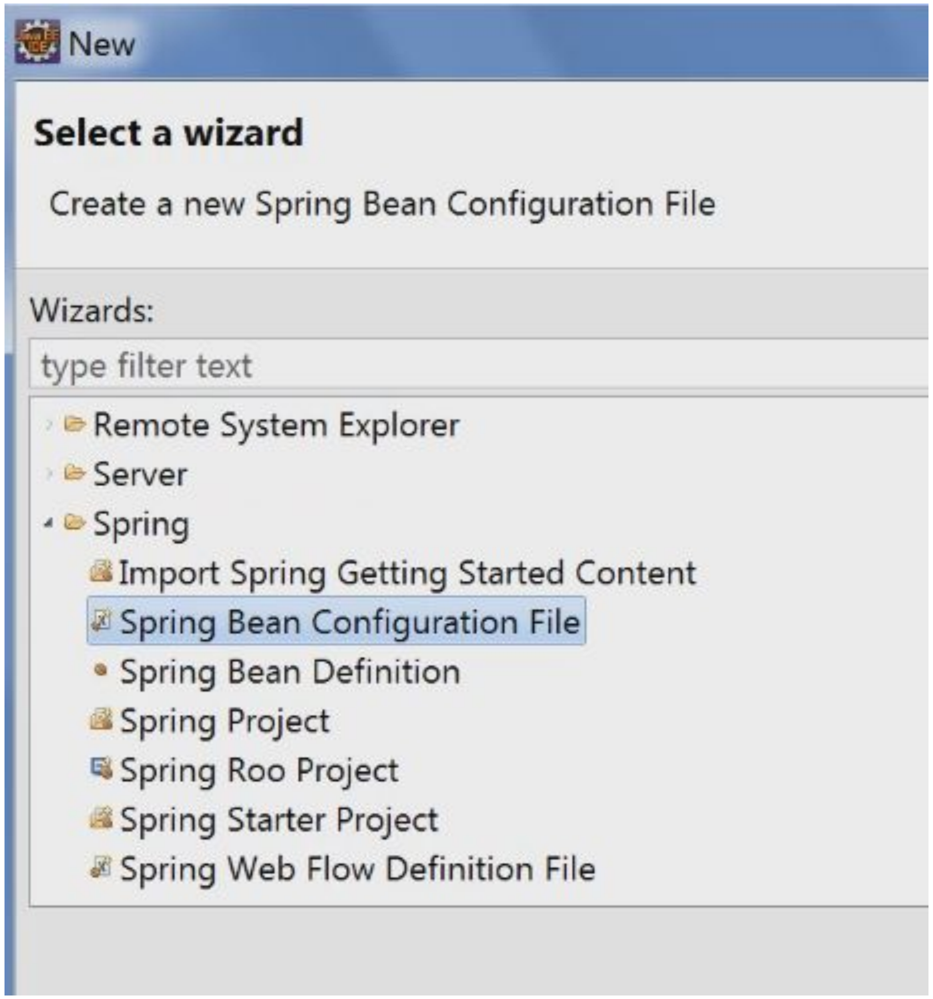


```
RobotImpl1
RobotImpl2
RobotSpring
RobotSpring2
  Spring Elements
  src/main/java
  src/main/resources
  src/test/java
  JRE System Library [J2SE-1.5]
  Maven Dependencies
    spring-context-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\org\s
    spring-aop-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\org\sprin
    aopalliance-1.0.jar - C:\Users\Tim\.m2\repository\aopalliance\aopalli
    spring-beans-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\org\spr
    spring-core-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\org\sprir
    commons-logging-1.1.1.jar - C:\Users\Tim\.m2\repository\commons-
    spring-expression-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\or
    spring-test-3.2.3.RELEASE.jar - C:\Users\Tim\.m2\repository\org\sprin
    junit-4.11.jar - C:\Users\Tim\.m2\repository\junit\junit\4.11
    hamcrest-core-1.3.jar - C:\Users\Tim\.m2\repository\org\hamcrest\ha
  src
  target
  pom.xml
```

```xml
        <junit.version>4.11</junit.version>

    </properties>

    <dependencies>
        <!-- Spring and Transactions -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${spring-framework.version}</ver
        </dependency>


        <!-- Test Artifacts -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-test</artifactId>
            <version>${spring-framework.version}</ver
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>
```


Project Explorer
Outline
RobotSpring2/pom.xml
RobotImpl1
RobotImpl2
RobotSpring
RobotSpring2
Spring Elements
src/ma
src/ma
src/te
JRE Sy
Mave
src
target
pom.x
New
Go Into
Open Type Hierarchy F4
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
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Project...
Annotation
Class
Enum
Interface
Package
Source Folder
Example...
Other... Ctrl+N
<java.ve
<project
<project
<!-- Spr
<spring-
</properties
<dependencie
<!-- Spr
<depende


New
Select a wizard
Wizards:
type filter text
Spring Project
Spring Roo Project
Spring Starter Project
Spring Web Flow Definition File
SQL Development
Tasks
User Assistance
Web
Web Services
XML
< Back
Next >
Finish
Cancel



New

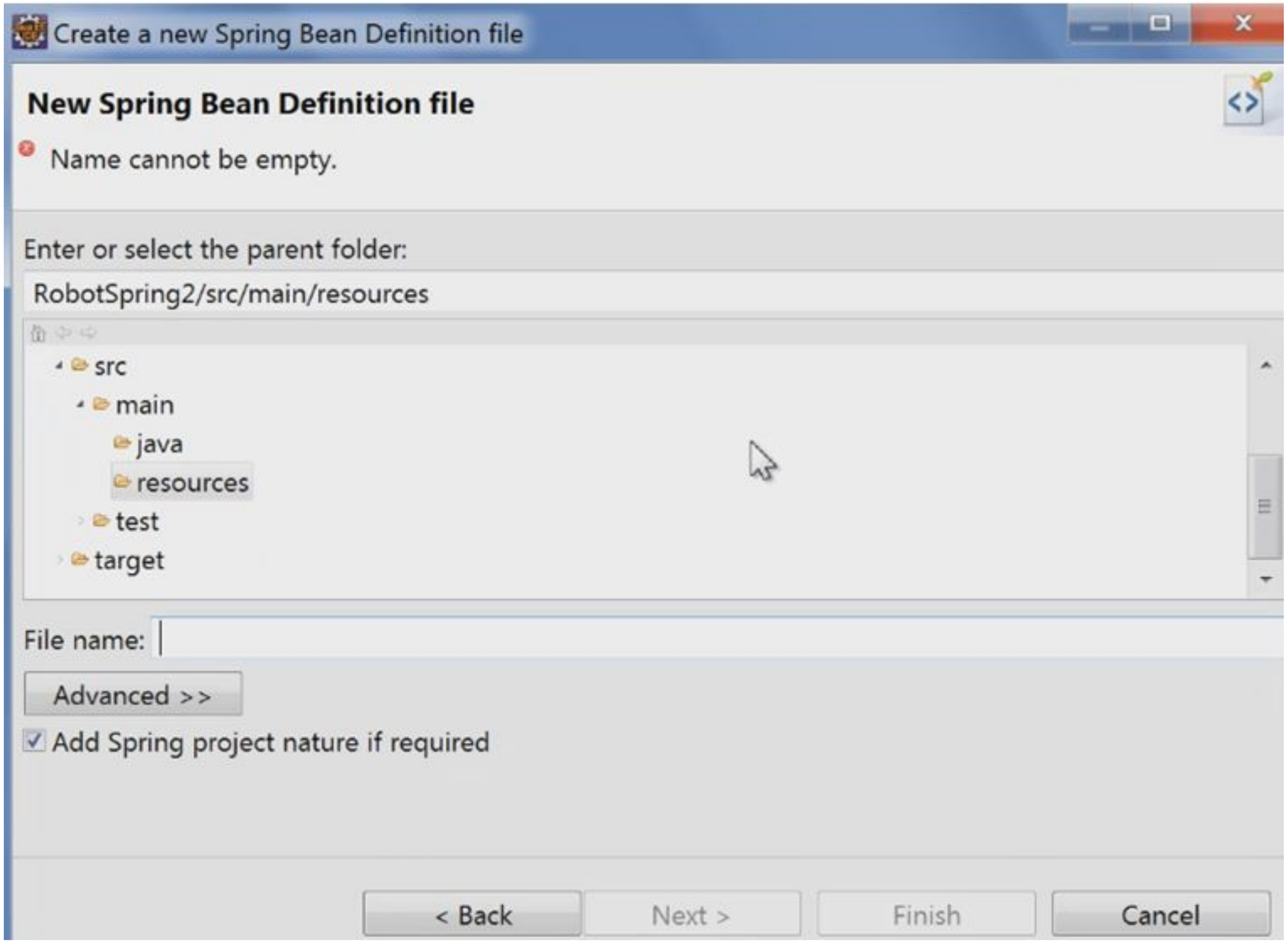
Select a wizard

Create a new Spring Bean Configuration File

Wizards:

type filter text

- Remote System Explorer
- Server
- Spring
  - Import Spring Getting Started Content
  - Spring Bean Configuration File
  - Spring Bean Definition
  - Spring Project
  - Spring Roo Project
  - Spring Starter Project
  - Spring Web Flow Definition File


Create a new Spring Bean Definition file

New Spring Bean Definition file

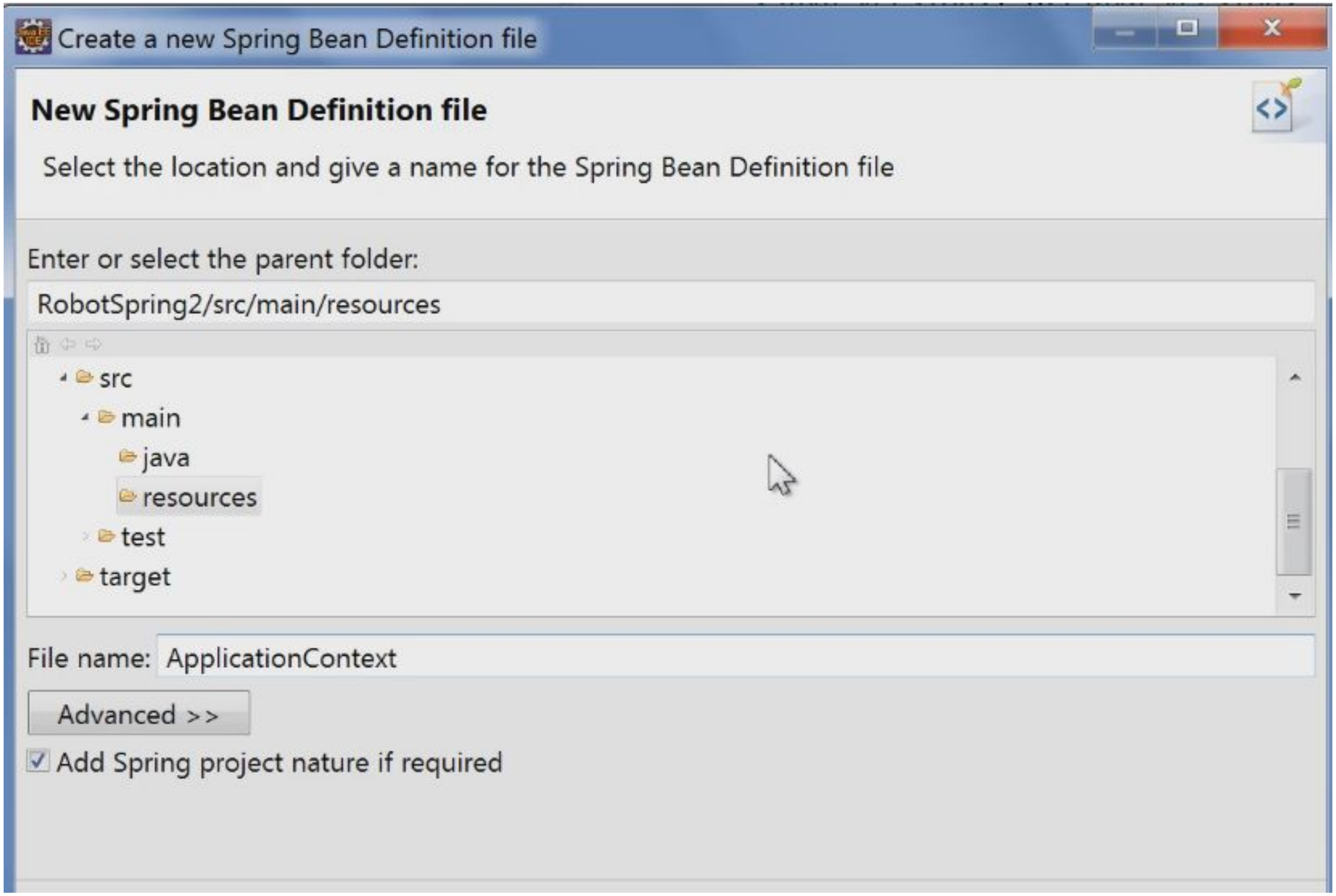
Name cannot be empty.

Enter or select the parent folder:

RobotSpring2/src/main/resources

File name:

[✓] Add Spring project nature if required



Create a new Spring Bean Definition file

New Spring Bean Definition file

Select the location and give a name for the Spring Bean Definition file

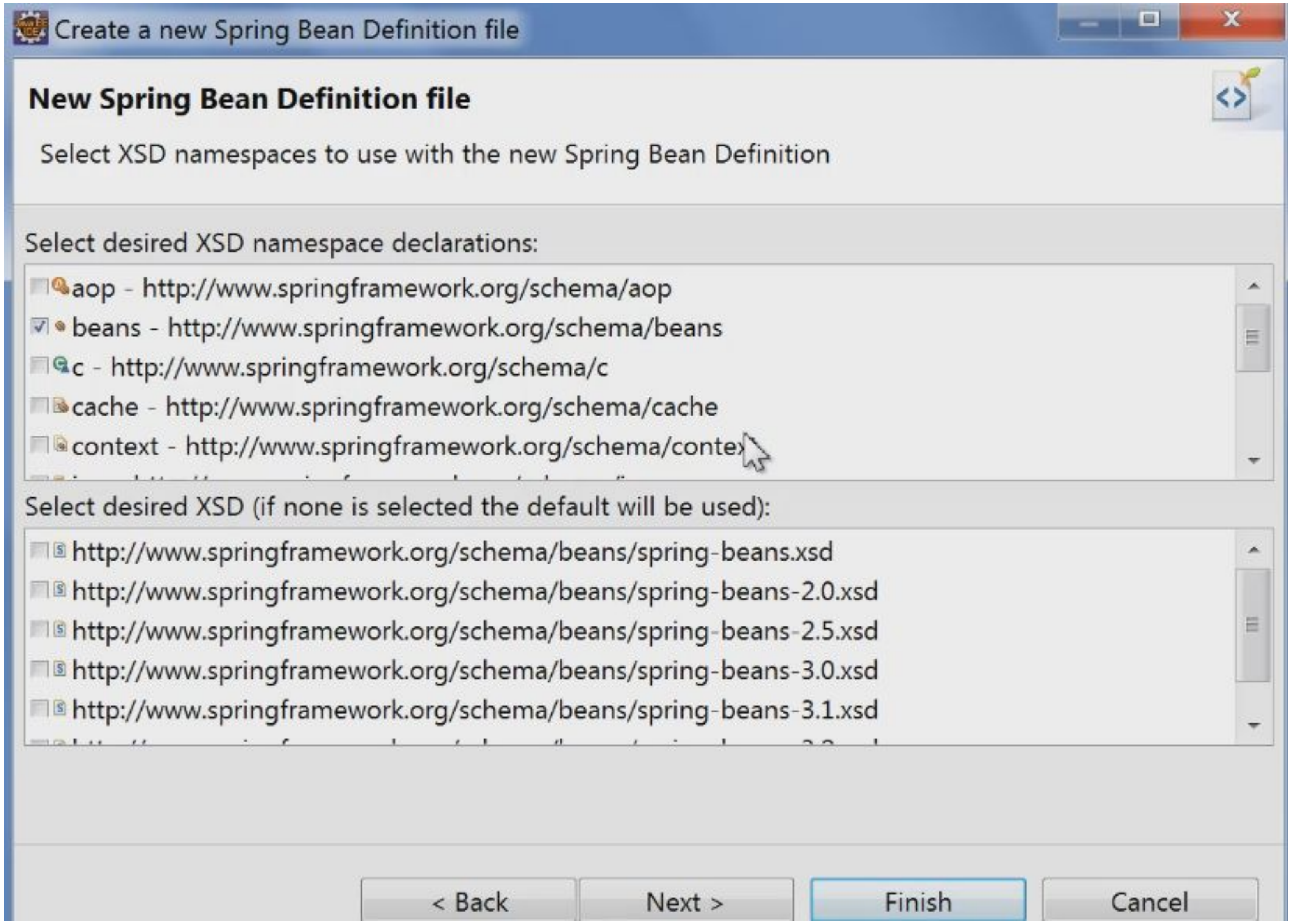
Enter or select the parent folder:

RobotSpring2/src/main/resources

- src
  - main
    - java
    - resources
  - test
- target

File name: ApplicationContext

[✓] Add Spring project nature if required


Create a new Spring Bean Definition file

New Spring Bean Definition file

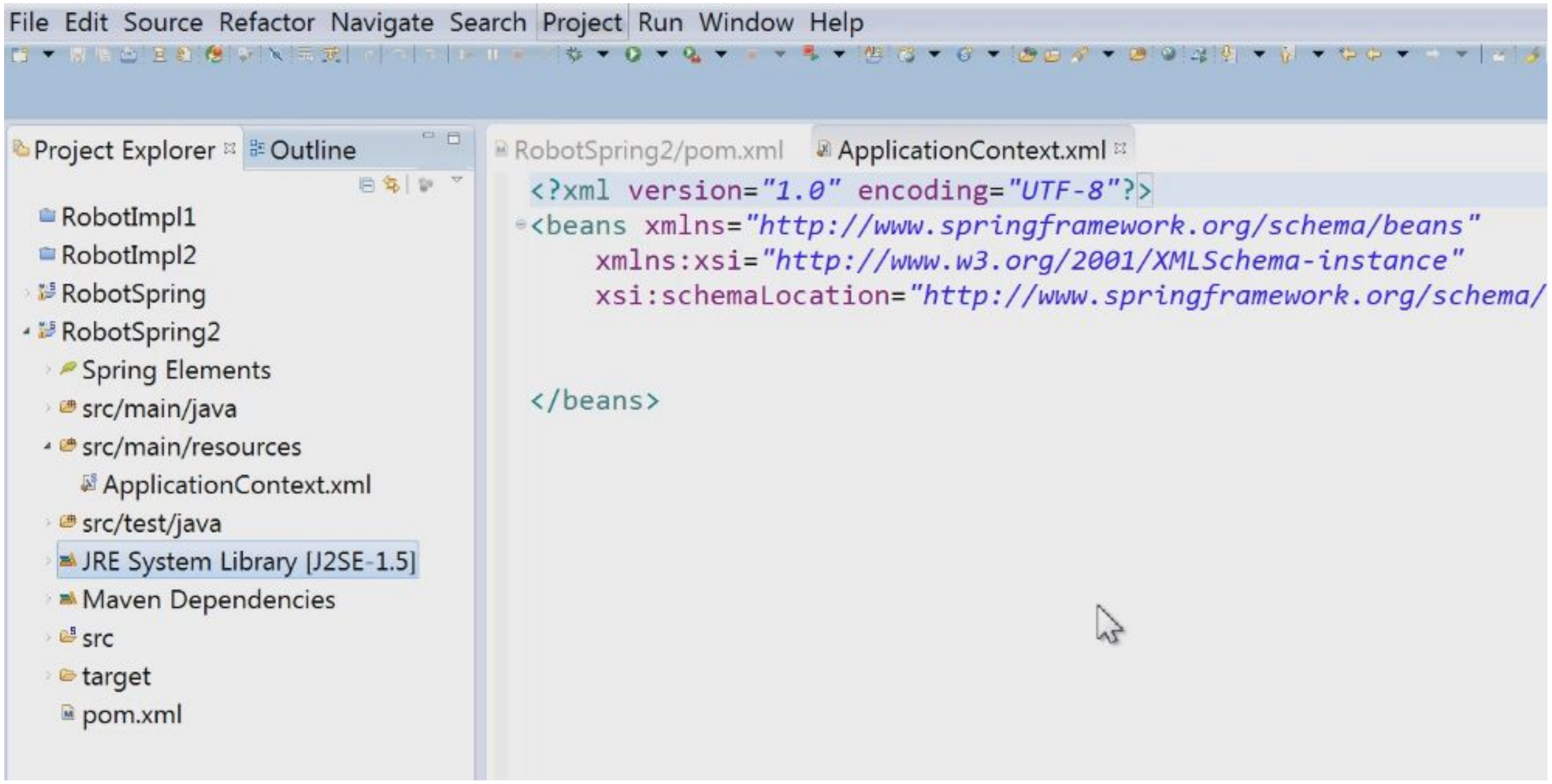
Select XSD namespaces to use with the new Spring Bean Definition

Select desired XSD namespace declarations:

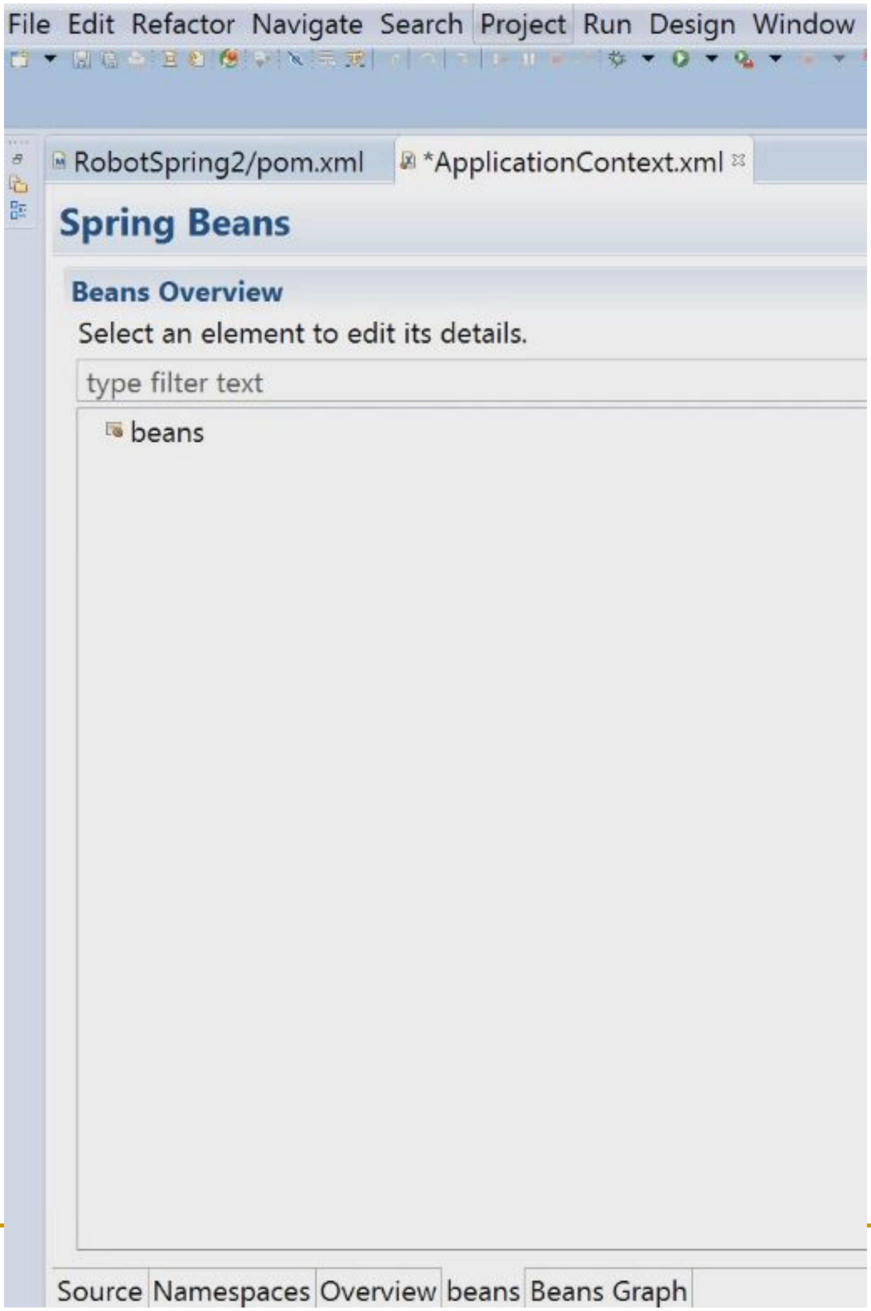
- [ ] aop - http://www.springframework.org/schema/aop
- [✓] beans - http://www.springframework.org/schema/beans
- [ ] c - http://www.springframework.org/schema/c
- [ ] cache - http://www.springframework.org/schema/cache
- [ ] context - http://www.springframework.org/schema/conte

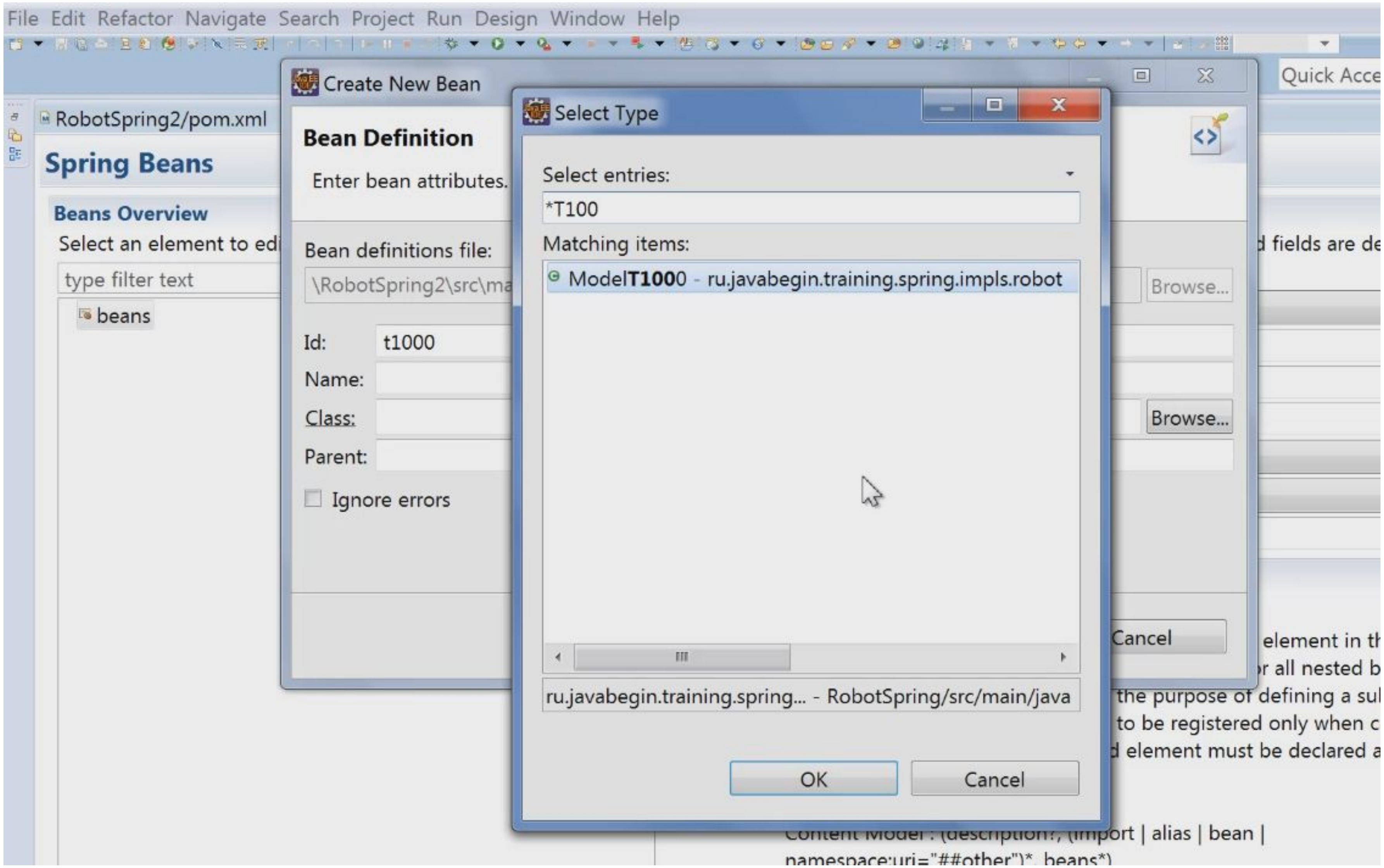
Select desired XSD (if none is selected the default will be used):

- [ ] http://www.springframework.org/schema/beans/spring-beans.xsd
- [ ] http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
- [ ] http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
- [ ] http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
- [ ] http://www.springframework.org/schema/beans/spring-beans-3.1.xsd



```xml
<?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>
```


File Edit Refactor Navigate Search Project Run Design Window
RobotSpring2/pom.xml
*ApplicationContext.xml
Spring Beans
Beans Overview
Select an element to edit its details.
type filter text
beans
Source Namespaces Overview beans Beans Graph



Bean Definition

Enter bean attributes.

Bean definitions file: \RobotSpring2\src\ma

Id: t1000

Name:

Class:

Parent:

Ignore errors

Select Type

Select entries:

*T100

Matching items:

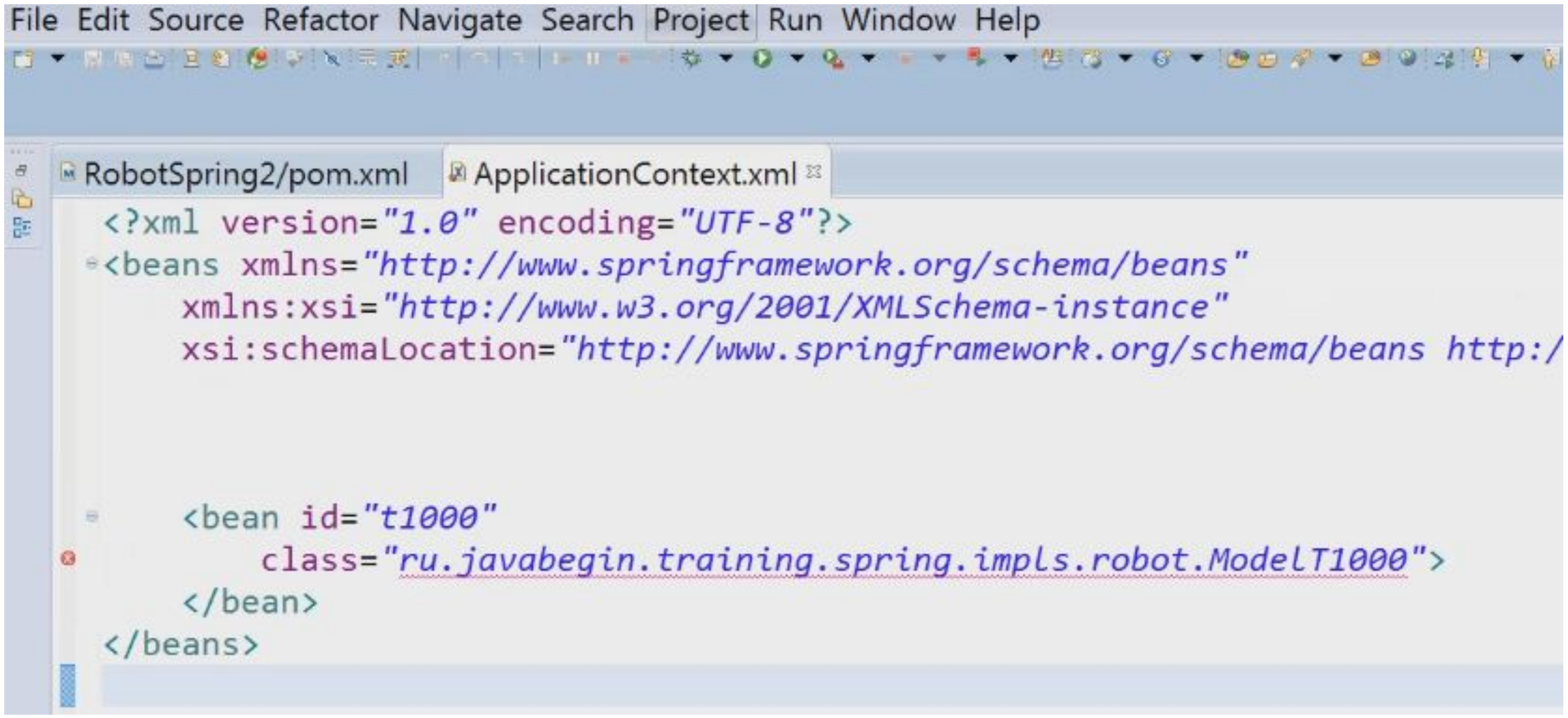
ModelT1000 - ru.javabegin.training.spring.impls.robot

ru.javabegin.training.spring... - RobotSpring/src/main/java

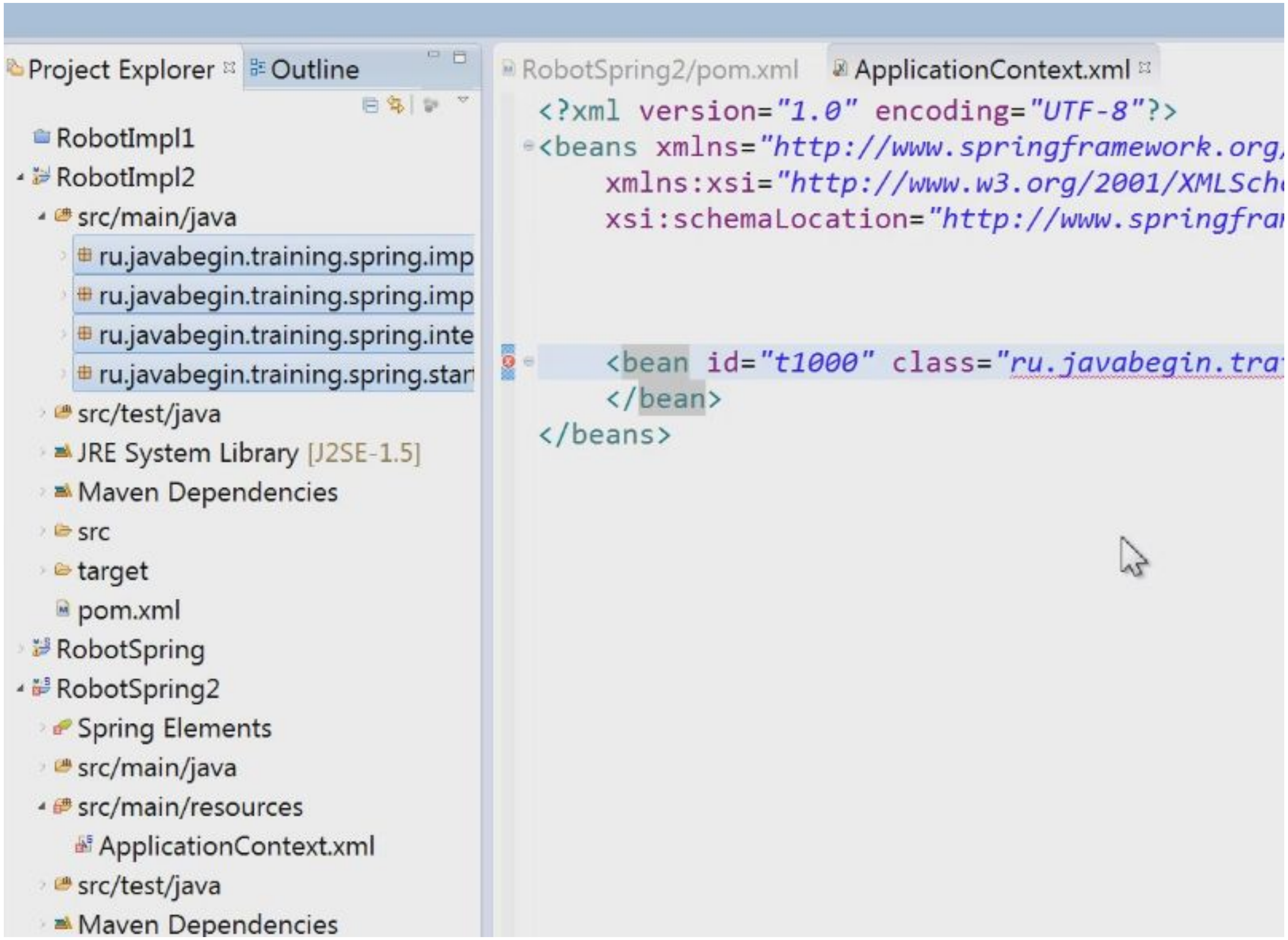
fields are de

element in th
r all nested b
the purpose of defining a sul
to be registered only when c
element must be declared a

Content Model : (description?, (import | alias | bean |
namespace:uri="##other")*, beans*)

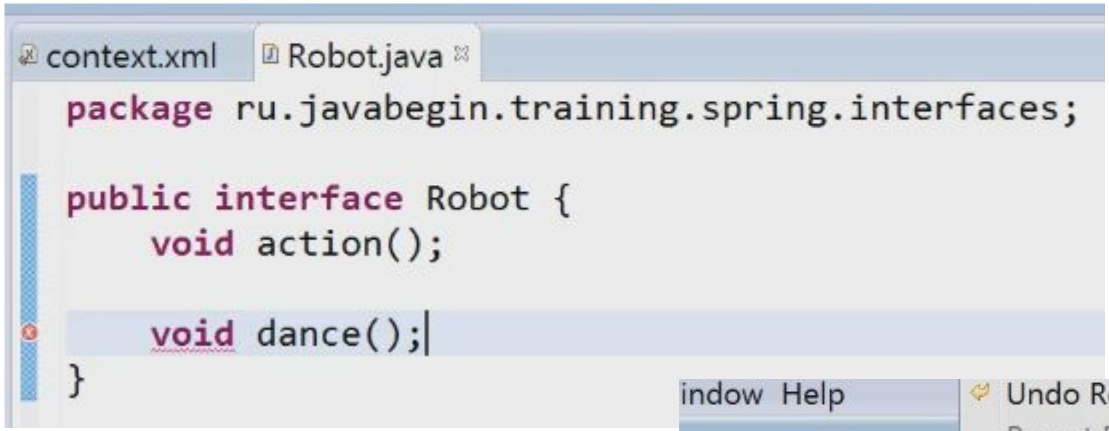

File Edit Source Refactor Navigate Search Project Run Window Help
RobotSpring2/pom.xml
ApplicationContext.xml

```
<?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.javabegin.training.spring.impls.robot.ModelT1000">
    </bean>
</beans>
```



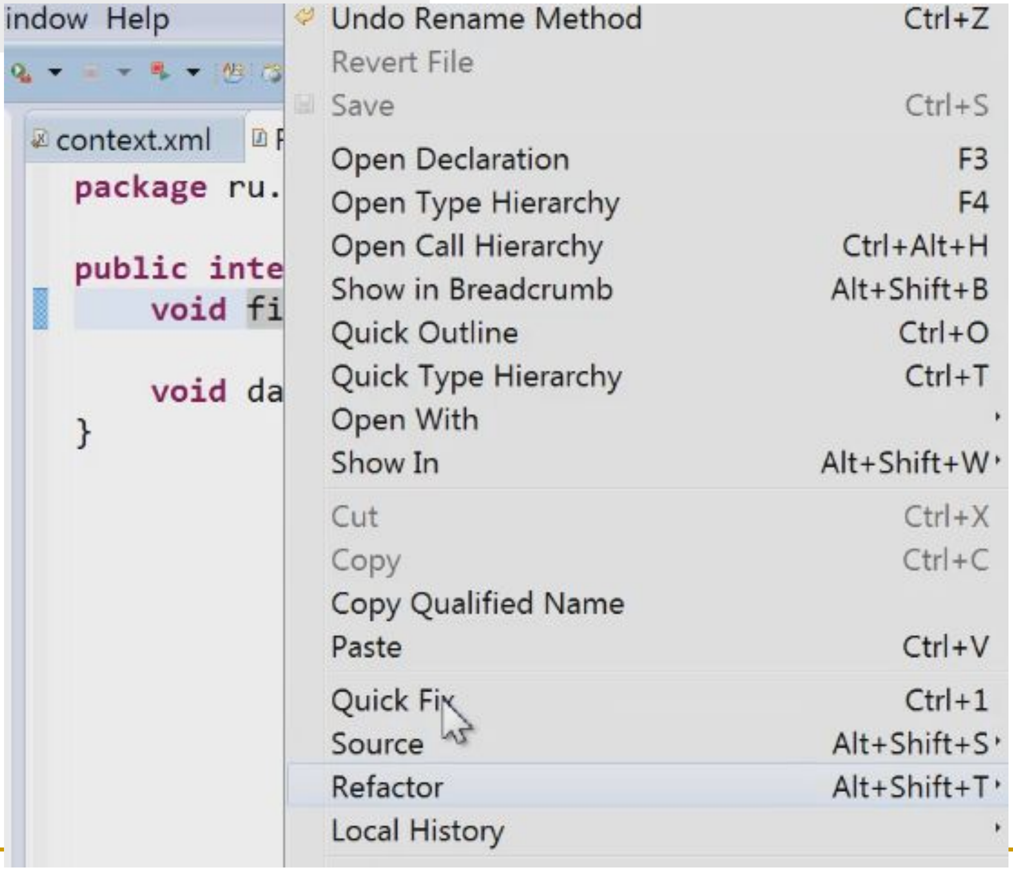
RobotImpl1
RobotImpl2
src/main/java
ru.javabegin.training.spring.imp
ru.javabegin.training.spring.imp
ru.javabegin.training.spring.inte
ru.javabegin.training.spring.star
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
RobotSpring
RobotSpring2
Spring Elements
src/main/java
src/main/resources
ApplicationContext.xml
src/test/java
Maven Dependencies

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org
    xmlns:xsi="http://www.w3.org/2001/XMLSch
    xsi:schemaLocation="http://www.springfra


    <bean id="t1000" class="ru.javabegin.tra
    </bean>
</beans>
```



context.xml | Robot.java

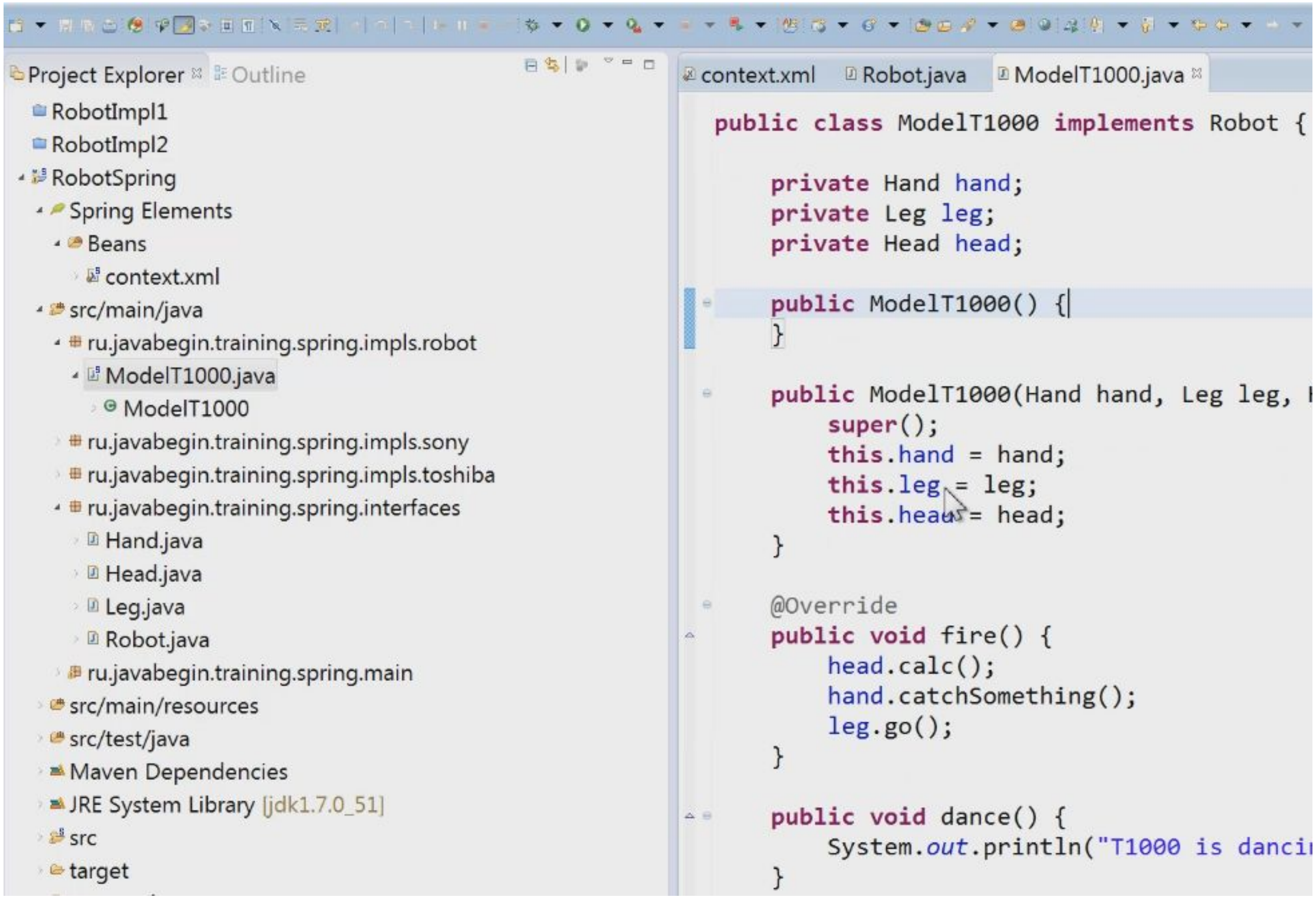
```
package ru.javabegin.training.spring.interfaces;

public interface Robot {
    void action();

    void dance();
}
```



indow Help

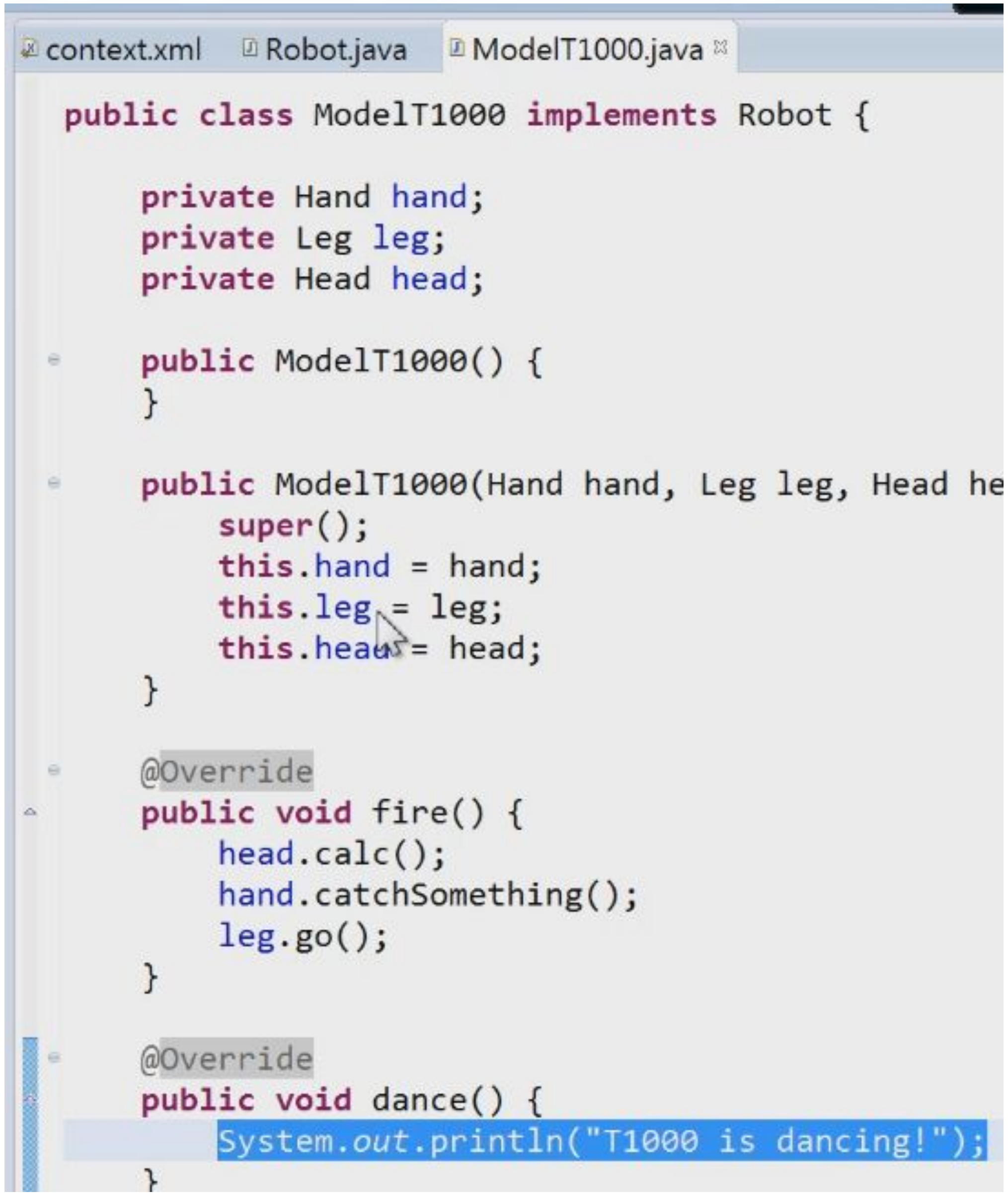
context.xml | R

```
package ru.

public inte
    void fi

    void da
}
```

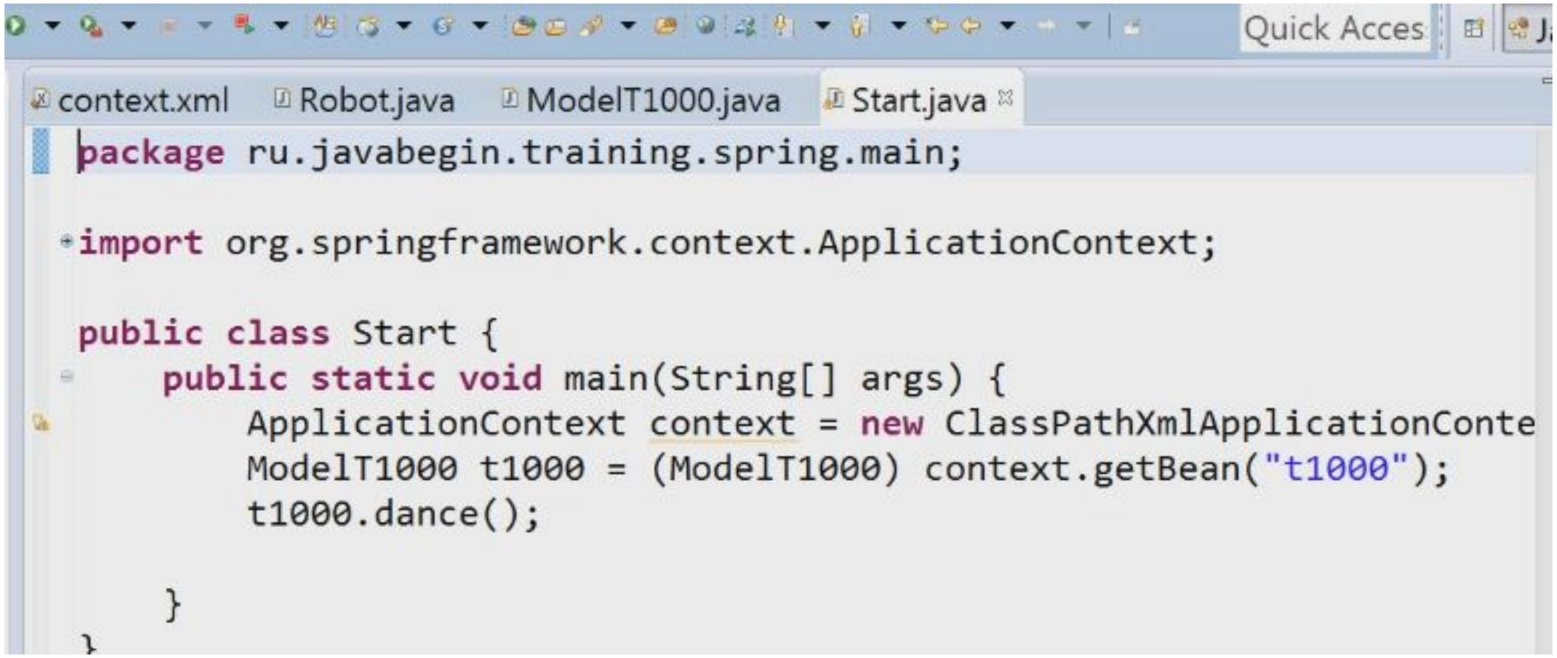


```
RobotImpl1
RobotImpl2
RobotSpring
  Spring Elements
    Beans
      context.xml
  src/main/java
    ru.javabegin.training.spring.impls.robot
      ModelT1000.java
        ModelT1000
    ru.javabegin.training.spring.impls.sony
    ru.javabegin.training.spring.impls.toshiba
    ru.javabegin.training.spring.interfaces
      Hand.java
      Head.java
      Leg.java
      Robot.java
    ru.javabegin.training.spring.main
  src/main/resources
  src/test/java
  Maven Dependencies
  JRE System Library [jdk1.7.0_51]
  src
  target
```

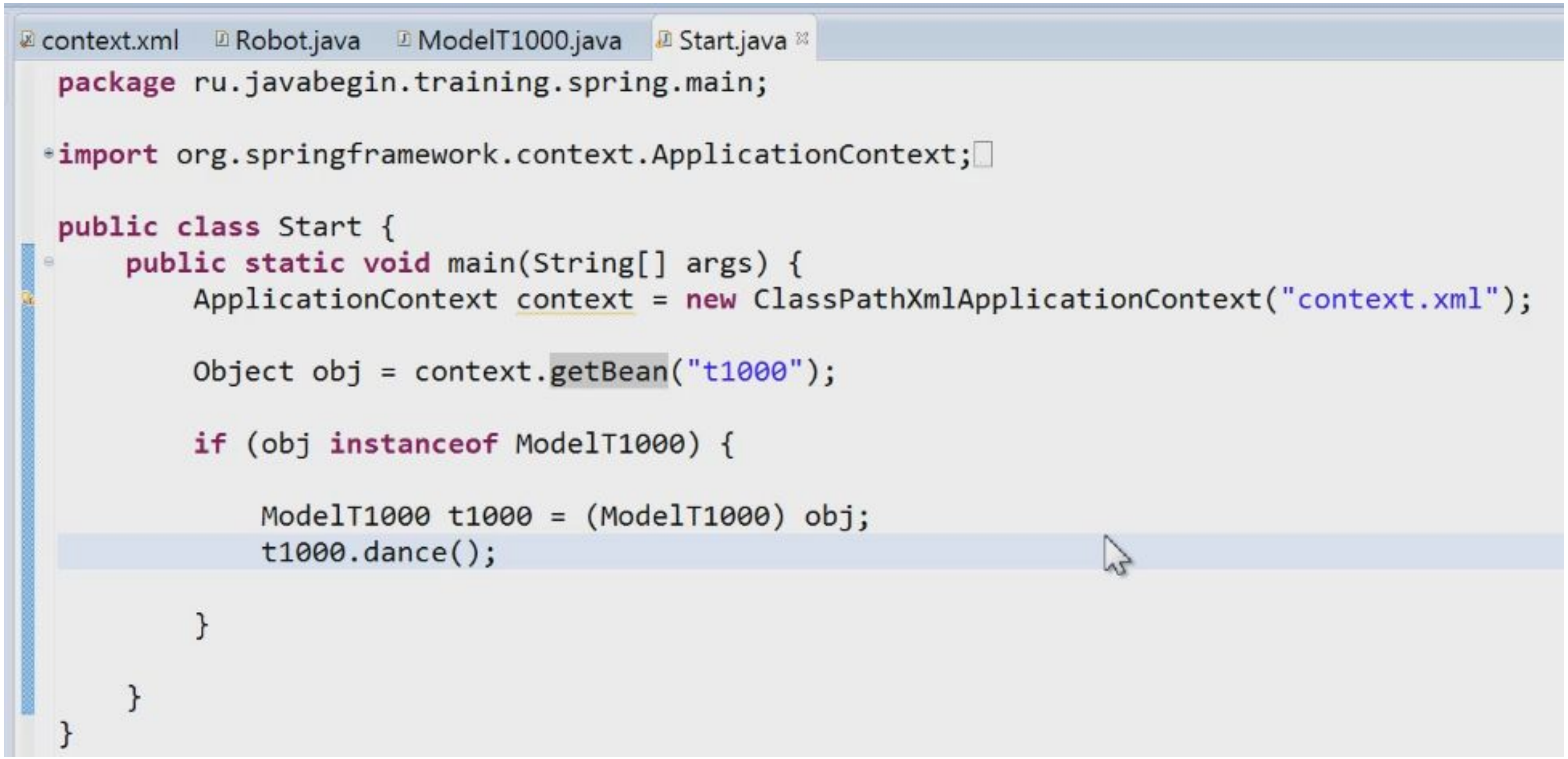
```java
public class ModelT1000 implements Robot {

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

    public ModelT1000() {
    }

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

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

    public void dance() {
        System.out.println("T1000 is dancin
    }
```



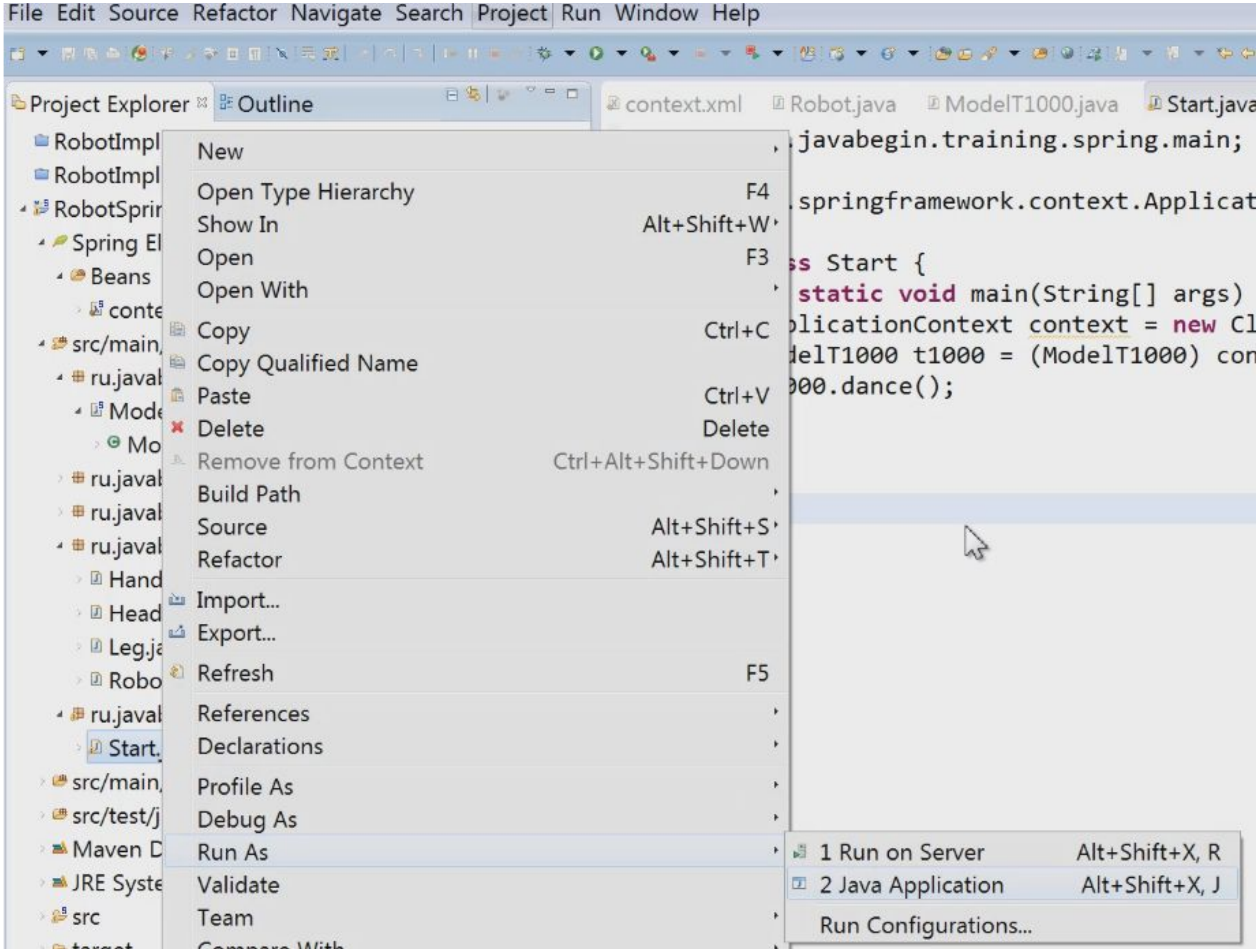
```
public class ModelT1000 implements Robot {

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

    public ModelT1000() {
    }

    public ModelT1000(Hand hand, Leg leg, Head he
        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!");
    }
```



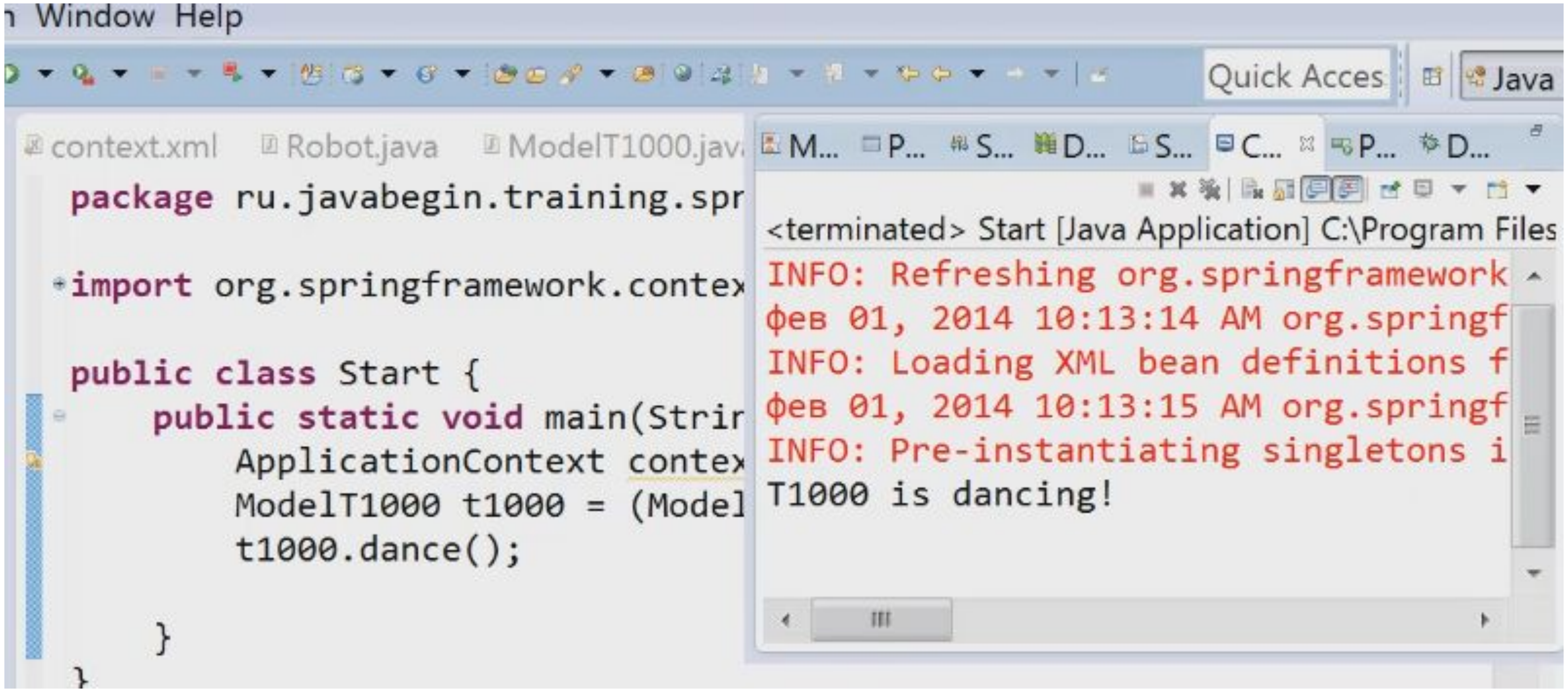
```
package ru.javabegin.training.spring.main;

import org.springframework.context.ApplicationContext;

public class Start {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationConte
        ModelT1000 t1000 = (ModelT1000) context.getBean("t1000");
        t1000.dance();

    }
}
```



```
package ru.javabegin.training.spring.main;

import org.springframework.context.ApplicationContext;

public class Start {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("context.xml");

        Object obj = context.getBean("t1000");

        if (obj instanceof ModelT1000) {

            ModelT1000 t1000 = (ModelT1000) obj;
            t1000.dance();

        }

    }
}
```

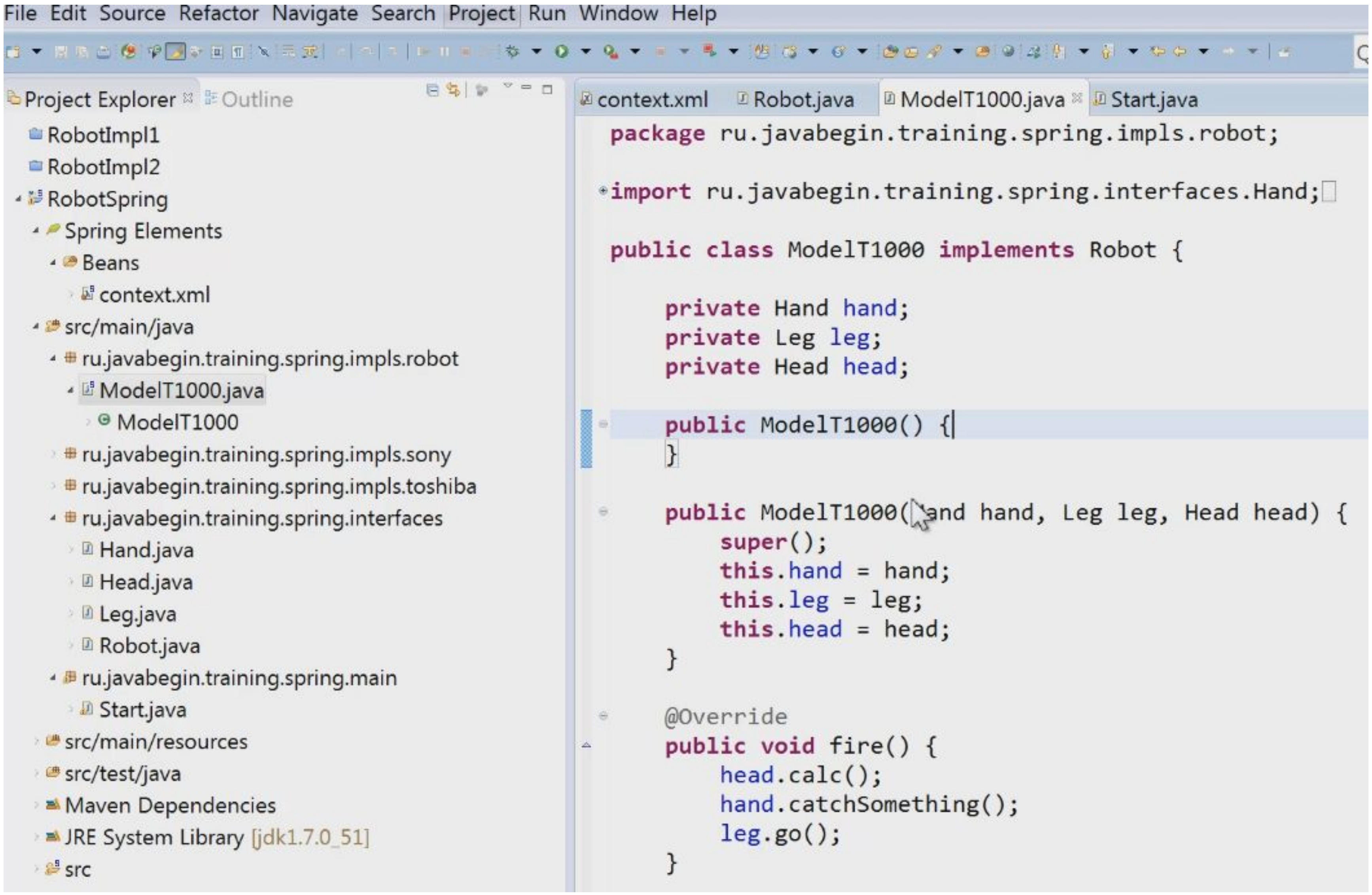


```
.javabegin.training.spring.main;

.springframework.context.Applicat

ss Start {
 static void main(String[] args)
plicationContext context = new Cl
delT1000 t1000 = (ModelT1000) con
000.dance();
```



```
package ru.javabegin.training.spr

import org.springframework.contex

public class Start {
    public static void main(Strir
        ApplicationContext contex
        ModelT1000 t1000 = (Model
        t1000.dance();

    }
}
```

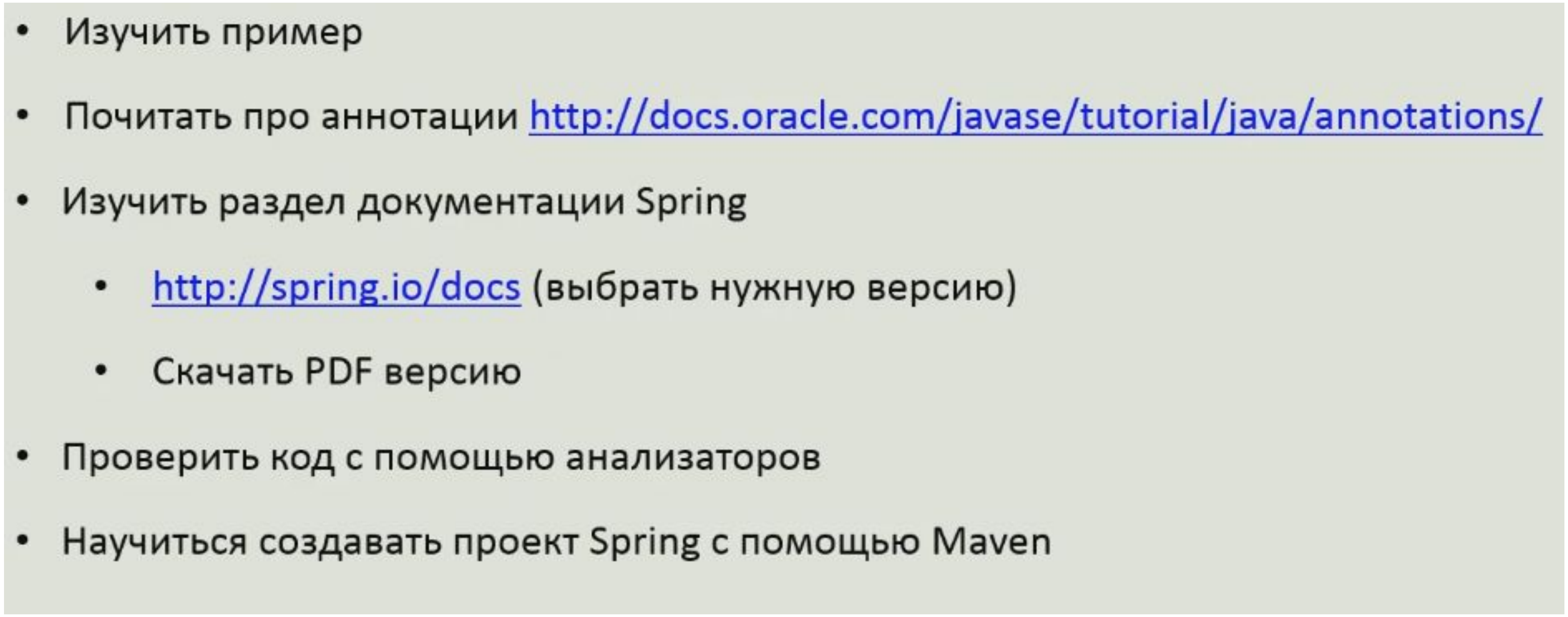
```
<terminated> Start [Java Application] C:\Program Files
INFO: Refreshing org.springframework
фев 01, 2014 10:13:14 AM org.springf
INFO: Loading XML bean definitions f
фев 01, 2014 10:13:15 AM org.springf
INFO: Pre-instantiating singletons i
T1000 is dancing!
```



```
RobotImpl1
RobotImpl2
RobotSpring
  Spring Elements
    Beans
      context.xml
  src/main/java
    ru.javabegin.training.spring.impls.robot
      ModelT1000.java
        ModelT1000
    ru.javabegin.training.spring.impls.sony
    ru.javabegin.training.spring.impls.toshiba
    ru.javabegin.training.spring.interfaces
      Hand.java
      Head.java
      Leg.java
      Robot.java
    ru.javabegin.training.spring.main
      Start.java
  src/main/resources
  src/test/java
  Maven Dependencies
  JRE System Library [jdk1.7.0_51]
  src
```

```java
package ru.javabegin.training.spring.impls.robot;

import ru.javabegin.training.spring.interfaces.Hand;

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();
    }
```

Домашнее задание



- Изучить пример
- Почитать про аннотации http://docs.oracle.com/javase/tutorial/java/annotations/
- Изучить раздел документации Spring
  - http://spring.io/docs (выбрать нужную версию)
  - Скачать PDF версию
- Проверить код с помощью анализаторов
- Научиться создавать проект Spring с помощью Maven