Spring Framework

Spring Framework

What is Spring Framework? (1)

Light-weight yet comprehensive framework for building Java SE and Java EE applications

What is Spring Framework? (2)

- Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.
- Spring enables you to build applications from "plain old Java objects" (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.
- Examples of how you, as an application developer, can use the Spring platform advantage:
  - Make a Java method execute in a database transaction without having to deal with transaction APIs.
  - Make a local Java method a remote procedure without having to deal with remote APIs.
  - Make a local Java method a management operation without having to deal with JMX APIs.
  - Make a local Java method a message handler without having to deal with JMS APIs.

Overview Of Spring Framework

- The Spring Framework is a lightweight solution and a potential one-stop-shop for building your enterprise-ready applications.
- Spring is modular, allowing you to use only those parts that you need, without having to bring in the rest.
  - You can use the **IoC** container, with Struts on top, but you can also use only the Hibernate integration code or the JDBC abstraction layer.
- The Spring Framework supports declarative transaction management, remote access to your logic through RMI or web services, and various options for persisting your data.
- It offers a full-featured **MVC** framework, and enables you to integrate **AOP** transparently into your software.

Key Features (1)

- JavaBeans-based configuration management, applying Inversion-of-Control principles, specifically using the Dependency Injection technique
  - This aims to reduce dependencies of components on specific implementations of other components.
- A core bean factory, which is usable globally
- Generic abstraction layer for database transaction management

Overview Of Spring Framework

- Spring is designed to be non-intrusive, meaning that your domain logic code generally has no dependencies on the framework itself.
- In your integration layer (such as the data access layer), some dependencies on the data access technology and the Spring libraries will exist.
- However, it should be easy to isolate these dependencies from the rest of your code base

Key Features (2)

- Built-in generic strategies for JTA and a single JDBC DataSource
  - This removes the dependency on a Java EE environment for transaction support.
- Integration with persistence frameworks Hibernate, JDO and iBATIS.
- MVC web application framework, built on core Spring functionality, supporting many technologies for generating views, including JSP, FreeMarker, Velocity, Tiles, iText, and POI.

Why Use Spring Framework?

Why Use Spring?

- Wiring of components through Dependency Injection
  - Promotes de-coupling among the parts that make the application
- Design to interfaces
  - Insulates a user of a functionality from implementation details
- Test-Driven Development (TDD)
  - POJO classes can be tested without being tied up with the framework

Why Use Spring?

- Declarative programming through AOP
  - Easily configured aspects, esp. transaction support
- Simplify use of popular technologies
  - Abstractions insulate application from specifics, eliminate redundant code
    - Handle common error conditions
    - Underlying technology specifics still accessible

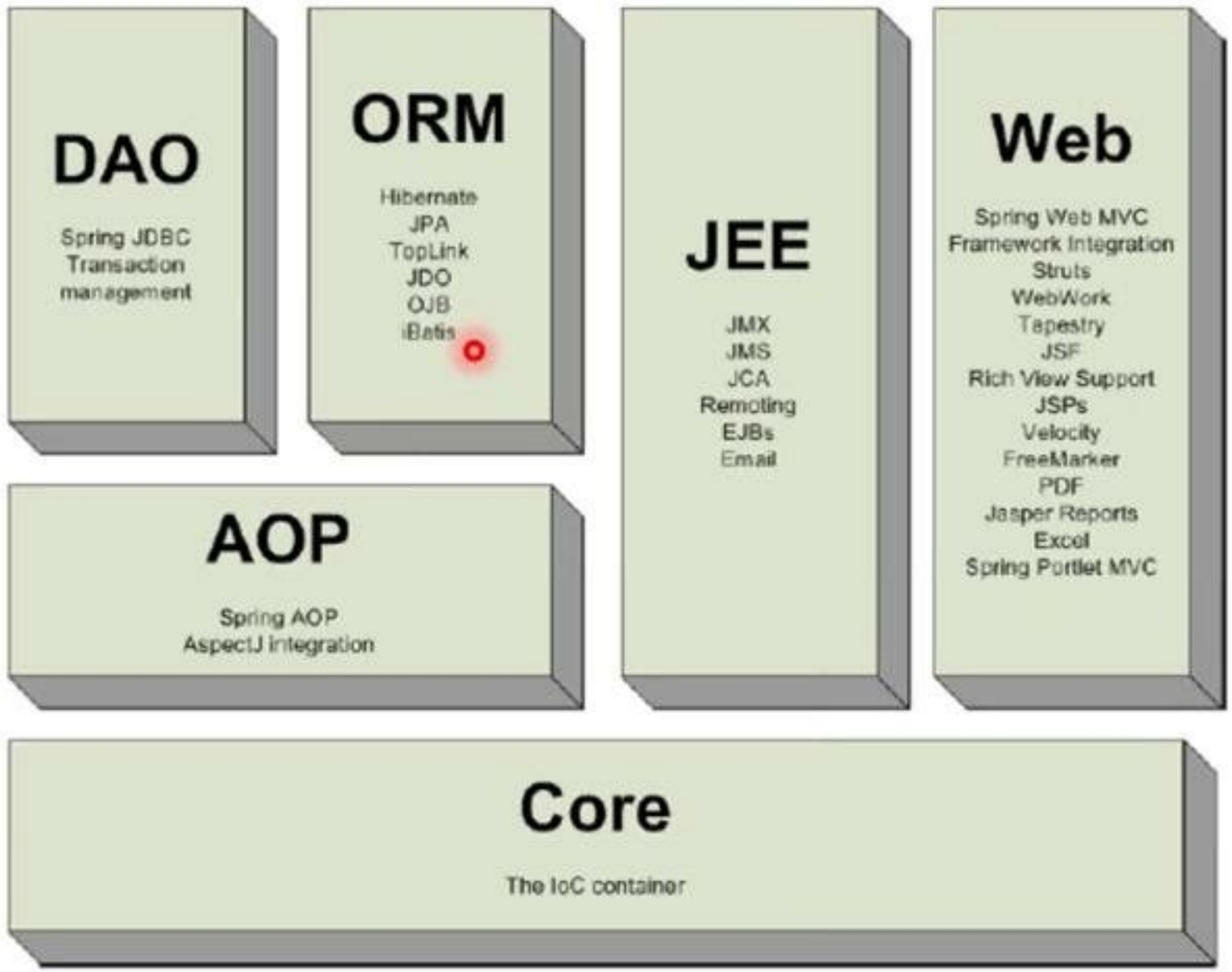
Why Use Spring?

- Conversion of checked exceptions to unchecked
- Extremely modular and flexible
- Well designed
  - Easy to extend
  - Many reusable classes

Why Use Spring?

- Integration with other technologies
  - EJB for J2EE
  - Hibernate, iBates, JDBC (for data access)
  - Velocity (for presentation)
  - Struts and WebWork (For web)

Spring Framework



Core Package

- Core package is the most fundamental part of the framework and provides the IoC and Dependency Injection features
- The basic concept here is the BeanFactory, which provides a sophisticated implementation of the factory pattern which removes the need for programmatic singletons and allows you to decouple the configuration and specification of dependencies from your actual program logic

The IOC Container and Dependency Injection

Dependency Injection and IOC Container

- Java applications -- a loose term that runs the gamut from constrained applets to n-tier server-side enterprise applications -- typically consist of objects that collaborate to form the application proper.
- Thus the objects in an application have *dependencies* on each other
- Although the Java platform provides a wealth of application development functionality, it lacks the means to organize the basic building blocks into a coherent whole, leaving that task to architects and developers

Dependency Injection and IOC Container

- Architects and Developers can use design patterns such as *Factory*, *Abstract Factory*, *Builder*, *Decorator*, and *Service Locator* to compose the various classes and object instances that make up an application
- Patterns are formalized best practices that *you must implement yourself* in your application.
- The Spring Framework *Inversion of Control* (IoC) component addresses this concern by providing a formalized means of composing disparate components into a fully working application ready for use

Dependency Injection (DI):
Basic concept

Spring Dependency Injection

- A kind of Inversion of Control (IoC)
- "Hollywood Principle"
  - Don't call me, I'll call you
- "Container" resolves (injects) dependencies of components by setting implementation object (push)
  - As opposed to component instantiating or Service Locator pattern where component locates implementation (pull)
- Martin Fowler calls Dependency Injection

Benefits of Dependency Injection

- Flexible
  - Avoid adding lookup code in business logic
- Testable
  - No need to depend on external resources or containers for testing
- Maintainable
  - Allows reuse in different application environments by changing configuration files instead of code
  - Promotes a consistent approach across all applications and teams

Two Dependency Injection Variants

- Constructor dependency Injection
  - Dependencies are provided through the constructors of the component
- Setter dependency injection
  - Dependencies are provided through the JavaBean style setter methods of the component
  - More popular than Constructor dependency injection

Constructor Dependency Injection

```
public class ConstructorInjection {

    private Dependency dep;

    public ConstructorInjection(Dependency dep) {

            this.dep = dep;

    }

}
```

WEB_INTELLECT7 - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer | JUnit

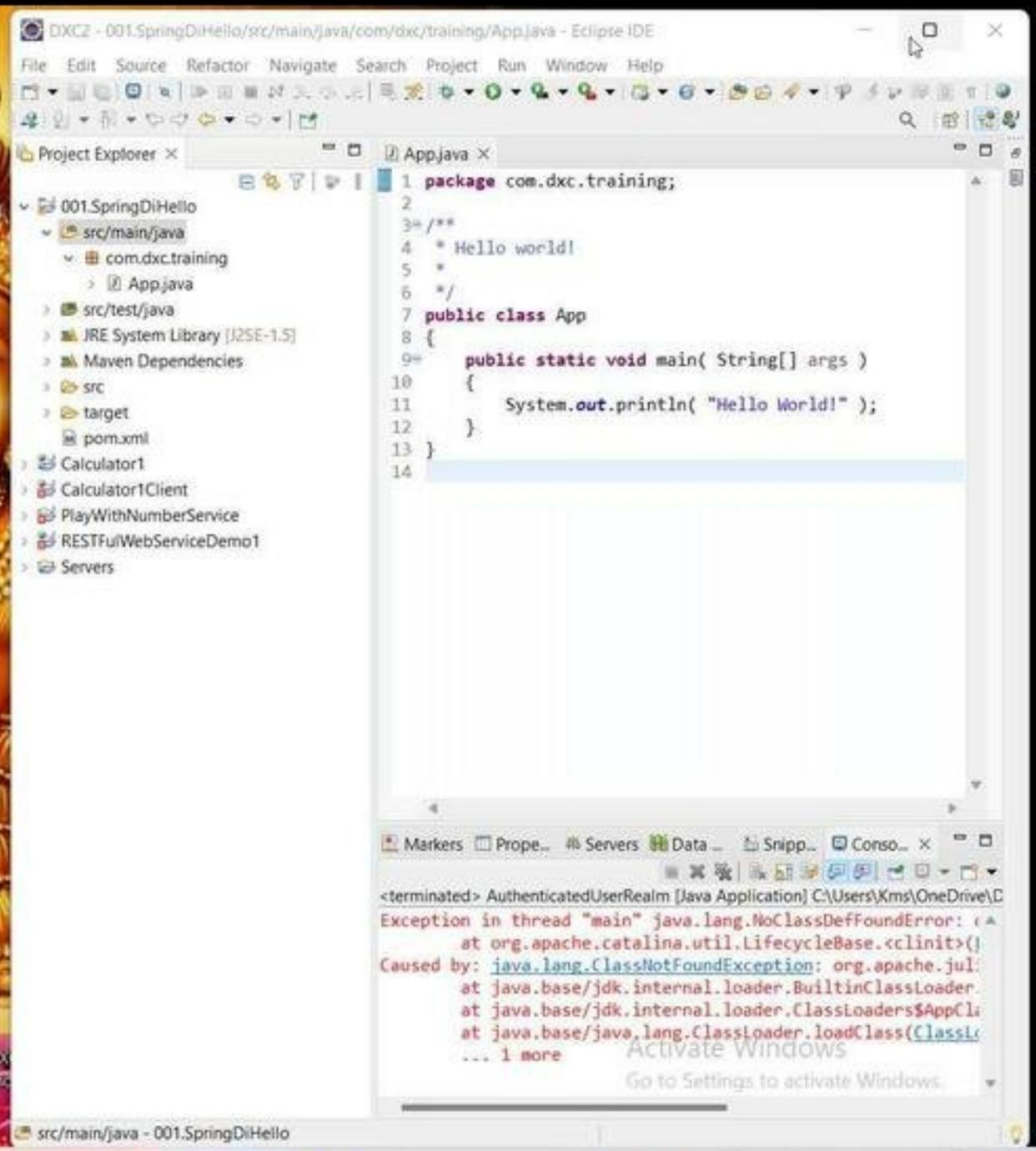
- 004.CarringMultiValuedData1
- 005.CarringMultiValuedData2
- 006.UsingServletConfigAndServletContext
- 007.DbAccessFromServlet
- 008.SessionManagementByURLReWriting
- 009.SessionManagemtByHiddenFields
- 010.SessionManagementByCookies
- 011.SessionManagementByHttpSession
- 012.HttpSessionAndAttributListener
- 013.JspBasicTagDemo
- 014.JspForwardAndInclude
- 015.JspBeanHandling
- 016.JspResponseAndExceptionHandling
- 017.JspBeansAndExceptionHandling
- 018.JspDatabaseHandling
- 019.JunitTestingPurseManger
- 020.JunitTestingCalculator
- Complaint-management-system
- EmployeeProject
- PlayWithNumberService
- PlayWithNumberServiceClient
- RESTFulService
- RESTFulServiceClient
- RESTfulWebservice2
  - JRE System Library [JavaSE-1.8]
  - src/main/java
    - com.dxc.rs
      - UnitConversionService.java
  - Server Runtime [Apache Tomcat v9.0]
  - Web App Libraries
  - build
  - src

Javadoc | Declaration | Console | Progress

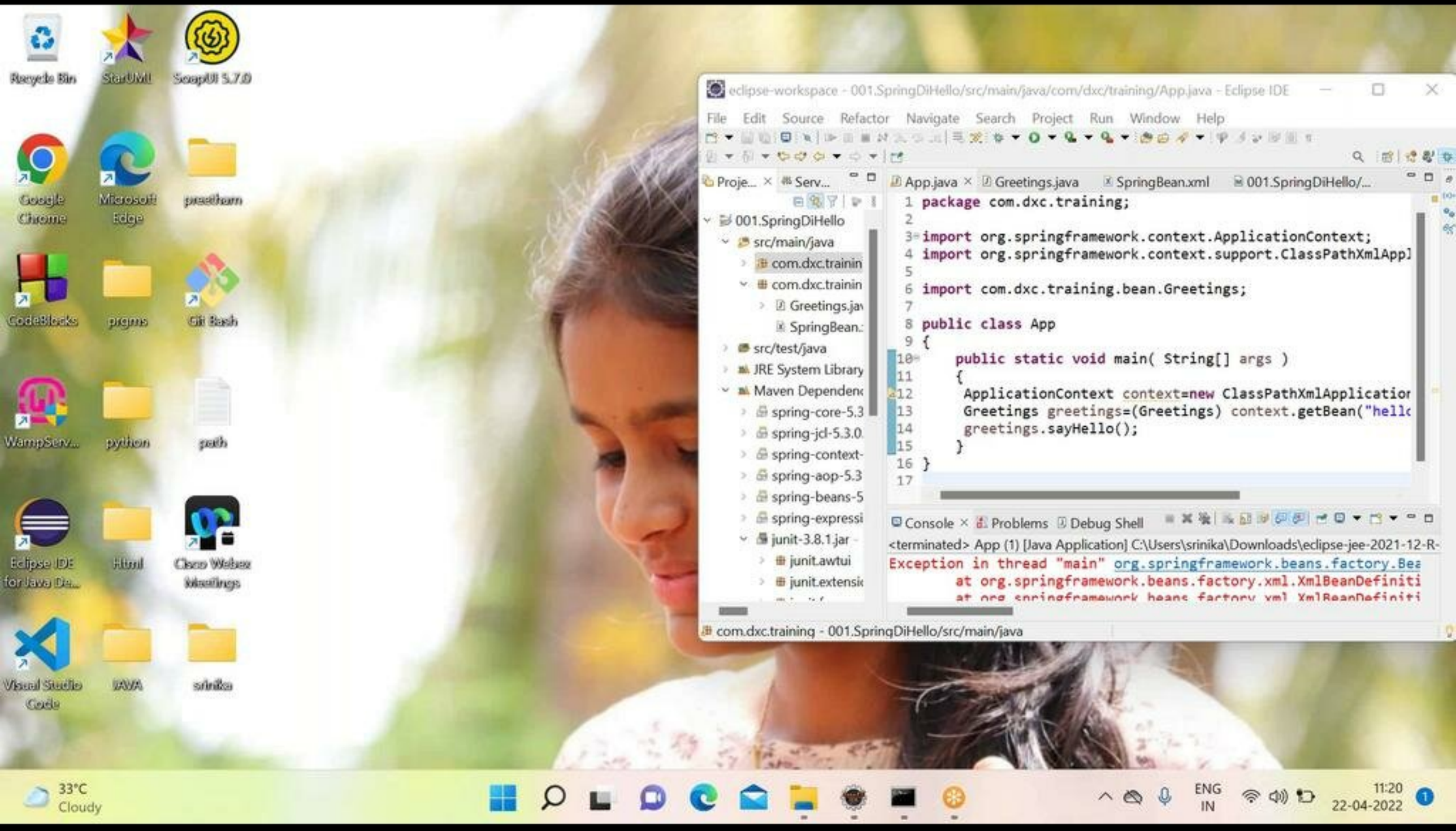
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_311\bin\javaw.exe (22-Apr-2022, 9:00:55 am)

```
Apr 22, 2022 9:01:00 AM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this log
Apr 22, 2022 9:01:03 AM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this log
Apr 22, 2022 9:01:05 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8082"]
Apr 22, 2022 9:01:05 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in [7090] milliseconds
```


Kms
fac work
Classification of Foot The...
GoToMeeting 003
GoToMeeting 007
This PC
WhatsApp
BHAGAVAT...
GoToMeeting 004
GoToMeeting 008
Recycle Bin
Paadaigal-S...
BHAGAVAT... AADHAR
GoToMeeting 005
GoToMeeting 009
Adobe Reader XI
Spotify
GOKUL AADHAR
dxc
GoToMeeting 010
CCleaner
Irandam Ulagam(20...
MITHRAN
DXCInsurance
Firefox
An IOT based Alert Syste...
eclipse - Shortcut
eclipse
Google Chrome
Iraivi (2016) Tamil 1080...
GoToMeeting 000
DXC1
KMPlayer
Batch 29 Methodology
GoToMeeting 001
SoapUI 5.7.0
Microsoft Edge
COMMENT An IOT base...
GoToMeeting 002
GoToMeeting 006


DXC2 - 001.SpringDiHello/src/main/java/com/dxc/training/App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
001.SpringDiHello
src/main/java
com.dxc.training
App.java
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
Calculator1
Calculator1Client
PlayWithNumberService
RESTFulWebServiceDemo1
Servers
App.java
1 package com.dxc.training;
2
3 /**
4 * Hello world!
5 *
6 */
7 public class App
8 {
9 public static void main(String[] args)
10 {
11 System.out.println("Hello World!");
12 }
13 }
14
Markers Prope... Servers Data ... Snipp... Conso...
<terminated> AuthenticatedUserRealm [Java Application] C:\Users\Kms\OneDrive\D
Exception in thread "main" java.lang.NoClassDefFoundError:
at org.apache.catalina.util.LifecycleBase.<clinit>()
Caused by: java.lang.ClassNotFoundException: org.apache.jul
at java.base/jdk.internal.loader.BuiltinClassLoader
at java.base/jdk.internal.loader.ClassLoaders$AppCl
at java.base/java.lang.ClassLoader.loadClass(ClassL
... 1 more
Activate Windows
Go to Settings to activate Windows.
src/main/java - 001.SpringDiHello

33°C
Partly sunny
ENG
IN
10:31
22-04-2022


Recycle Bin
StarUML
SoapUI 5.7.0
Google Chrome
Microsoft Edge
preethem
CodeBlocks
pgms
Git Bash
WampServ...
python
path
Eclipse IDE for Java De...
Html
Cisco Webex Meetings
Visual Studio Code
JAVA
srinika
eclipse-workspace - 001.SpringDiHello/src/main/java/com/dxc/training/App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Proje... Serv...
001.SpringDiHello
src/main/java
com.dxc.trainin
com.dxc.trainin
Greetings.jav
SpringBean.
src/test/java
JRE System Library
Maven Dependenc
spring-core-5.3
spring-jcl-5.3.0
spring-context-
spring-aop-5.3
spring-beans-5
spring-expressi
junit-3.8.1.jar
junit.awtui
junit.extensi
App.java
Greetings.java
SpringBean.xml
001.SpringDiHello/...
1 package com.dxc.training;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlAppl
5
6 import com.dxc.training.bean.Greetings;
7
8 public class App
9 {
10 public static void main(String[] args)
11 {
12 ApplicationContext context=new ClassPathXmlApplicatior
13 Greetings greetings=(Greetings) context.getBean("hello
14 greetings.sayHello();
15 }
16 }
17
Console Problems Debug Shell
<terminated> App (1) [Java Application] C:\Users\srinika\Downloads\eclipse-jee-2021-12-R-
Exception in thread "main" org.springframework.beans.factory.Bea
at org.springframework.beans.factory.xml.XmlBeanDefiniti
com.dxc.training - 001.SpringDiHello/src/main/java
33°C
Cloudy
ENG
IN
11:20
22-04-2022

Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- 001.SpringDiHello
  - src/main/java
    - com.dxc.training
      - App.java
      - Greetings.java
    - com.dxc.training.bean
    - StringBean.xml
  - src/test/java
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
    - main
    - test
  - target
  - pom.xml

Move

Update references to 'Greetings.java'

Update fully qualified names in non-Java text files (forces preview)

File name patterns: *

The patterns are separated by commas (* = any string, ? = any character)

Preview > OK Cancel

Grammars

bundle

Bina

Cac

project

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> App [Java Application] C:\New folder\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Apr 22, 2022, 11:30:08 AM – 1

```
Apr 22, 2022 11:30:10 AM org.springframework.context.support.AbstractApplicationContext refresh
WARNING: Exception encountered during context initialization - cancelling refresh attempt: org.springframework.beans.factory.CannotLoadBeanClas
Exception in thread "main" org.springframework.beans.factory.CannotLoadBeanClassException: Cannot find class [com.dxc.training.bean.Greetings]
        at org.springframework.beans.factory.support.AbstractBeanFactory.resolveBeanClass(AbstractBeanFactory.java:1539)
        at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.determineTargetType(AbstractAutowireCapableBeanFactory.
        at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.predictBeanType(AbstractAutowireCapableBeanFactory.java
        at org.springframework.beans.factory.support.AbstractBeanFactory.isFactoryBean(AbstractBeanFactory.java:1667)
        at org.springframework.beans.factory.support.AbstractBeanFactory.isFactoryBean(AbstractBeanFactory.java:1142)
        at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:924)
```

com.dxc.training.Greetings.java - 001.SpringDiHello/src/main/java

Type here to search

33°C Light rain ENG 11:35 AM 4/22/2022

DXC_SPRING - 005.SpringDiAsObject/src/main/java/com/dxc/training/bean/Triangle.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- 001.SpringDiHello
- 002.SpringDiCalculator
- 003.SpringDiByConstructorInje
- 004.SpringDiBySetterInjection
- 005.SpringDiAsObject
  - src/main/java
    - com.dxc.training
    - com.dxc.training.bean
      - Triangle.java
      - TrianglePoints.java
      - ApplicationContext.xml
  - src/test/java
  - JRE System Library [J2SE-1.
  - Maven Dependencies
  - src
  - target
  - pom.xml
- 006.SpringDiByCollectionObje
- 007.SpringDiAutowiredByNam
  - src/main/java
    - com.dxc.training
      - App.java
    - com.dxc.training.bean
      - Triangle.java
      - TrianglePoints.java
      - ApplicationContext.xml
  - src/test/java
  - JRE System Library [J2SE-1.
  - Maven Dependencies
  - src
  - target

ApplicationContext.xml | *Triangle.java | Triangle.java

```
1  package com.dxc.training.bean;
2
3  public class Triangle {
4      private TrianglePoints pointA;
5      private TrianglePoints pointB;
6      private TrianglePoints pointC;
7      public TrianglePoints getPointA() {
8          return pointA;
9      }
10     public void setPointA(TrianglePoints pointA) {
11         this.pointA = pointA;
12     }
13     public TrianglePoints getPointB() {
14         return pointB;
15     }
16     public void setPointB(TrianglePoints pointB) {
17         this.pointB = pointB;
18     }
```

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> App (6) [Java Application] D:\eclipse-jee-2021-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (22-Apr-2022, 1:10:30 pm

```
TrianglePoints ==> xPoint=55, yPoint=89
TrianglePoints ==> xPoint=34, yPoint=98
TrianglePoints ==> xPoint=75, yPoint=58
```

Writable | Smart Insert | 41 : 2 [955]