

Desarrollo de Aplicaciones Empresariales con Spring Framework Core 5

ISC. Ivan Venor García Baños



Agenda

1. Presentación
2. Objetivos
3. Contenido
4. Despedida

3. Contenido

- i. Introducción a Spring Framework
- ii. Spring Core
- iii. Spring AOP
- iv. Spring JDBC – Transaction
- v. Spring ORM – Hibernate 5
- vi. Spring Data JPA
- vii. Fundamentos Spring MVC y Spring REST
- viii. Fundamentos Spring Security
- ix. Seguridad en Servicios REST
- x. Introducción Spring Boot

3. Contenido

- i. **Introducción a Spring Framework**
- ii. Spring Core
- iii. Spring AOP
- iv. Spring JDBC – Transaction
- v. Spring ORM – Hibernate 5
- vi. Spring Data JPA
- vii. Fundamentos Spring MVC y Spring REST
- viii. Fundamentos Spring Security
- ix. Seguridad en Servicios REST
- x. Introducción Spring Boot

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

a. POJOs

b. JavaBeans

c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

i.i ¿Qué es Spring Framework?

Objetivos de la lección

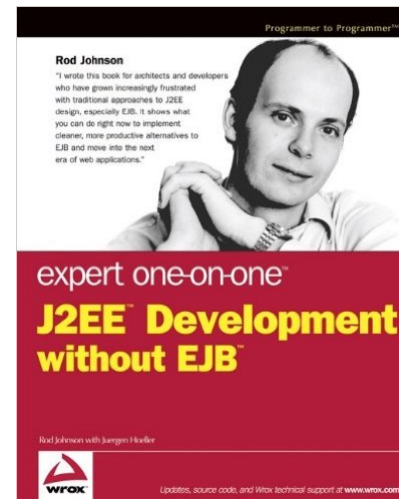
i.i ¿Qué es Spring Framework?

- Comprender el origen y objetivo de Spring Framework
- Conocer cual es la relación de Spring Framework con Java EE
- Conocer el modelo de programación que utiliza Spring Framework
- Conocer los conceptos básicos: POJO, Java Beans y Spring Beans

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework? (a)

- Framework ligero open source no intrusivo para el desarrollo de aplicaciones Java EE.
- Spring framework fue escrito originalmente por Rod Johnson y lanzado al mercado en junio del 2003.
- Expert one-on-one
“J2EE Development without EJB”



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework? (b)

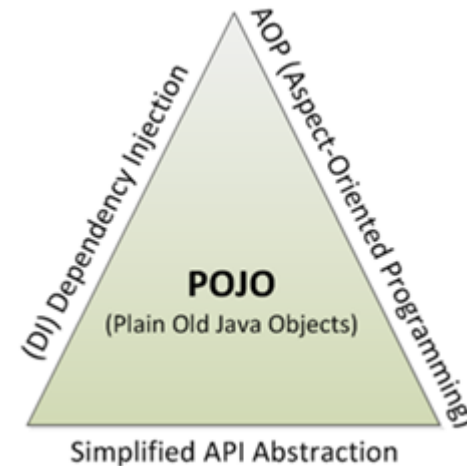
- Provee una infraestructura completa para el desarrollo de aplicaciones Java EE robustas.
- Provee un contenedor de Inversión de Control e Inyección de Dependencias que facilita la reutilización de componentes en aplicaciones Java EE.



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(c)

- Utiliza configuración a base de Java Beans y POJOs
- Implementa patrones de diseño, POO y buenas prácticas de programación.



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(d)

- POJO (Plain Old Java Object)
- No extiende ni implementa nada en especial
- En Java, sirve para enfatizar que una clase no depende de ningún Framework en especial.
- Surge como oposición a EJB anteriores a la versión 3.0

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(e)

- Java Bean
- Conocido también como simplemente Bean
- Contiene:
 - Constructor sin argumentos (por default)
 - Atributos privados y, setters y getters para cada uno de ellos
 - Implementa Serializable
 - Implementa el uso de lower-camel-case
- Ideales para generar modelo de objetos de dominio

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(f)

- Enterprise Java Bean
- Componente de negocio de Java EE
- Requiere contenedor de EJB
- Accede a recursos del servidor de aplicaciones
- Dependientes de la plataforma de despliegue
- Tipos:
 - EntityBean
 - SessionBean
 - MessageDrivenBean

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(g)

- Orientado a Interfaces

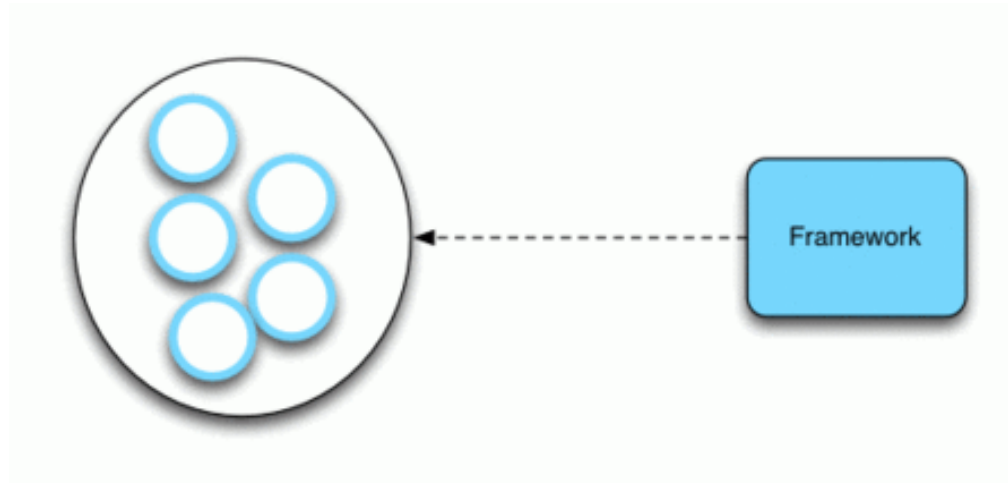


Multiple Inheritance in Java

i. Introducción a Spring Framework

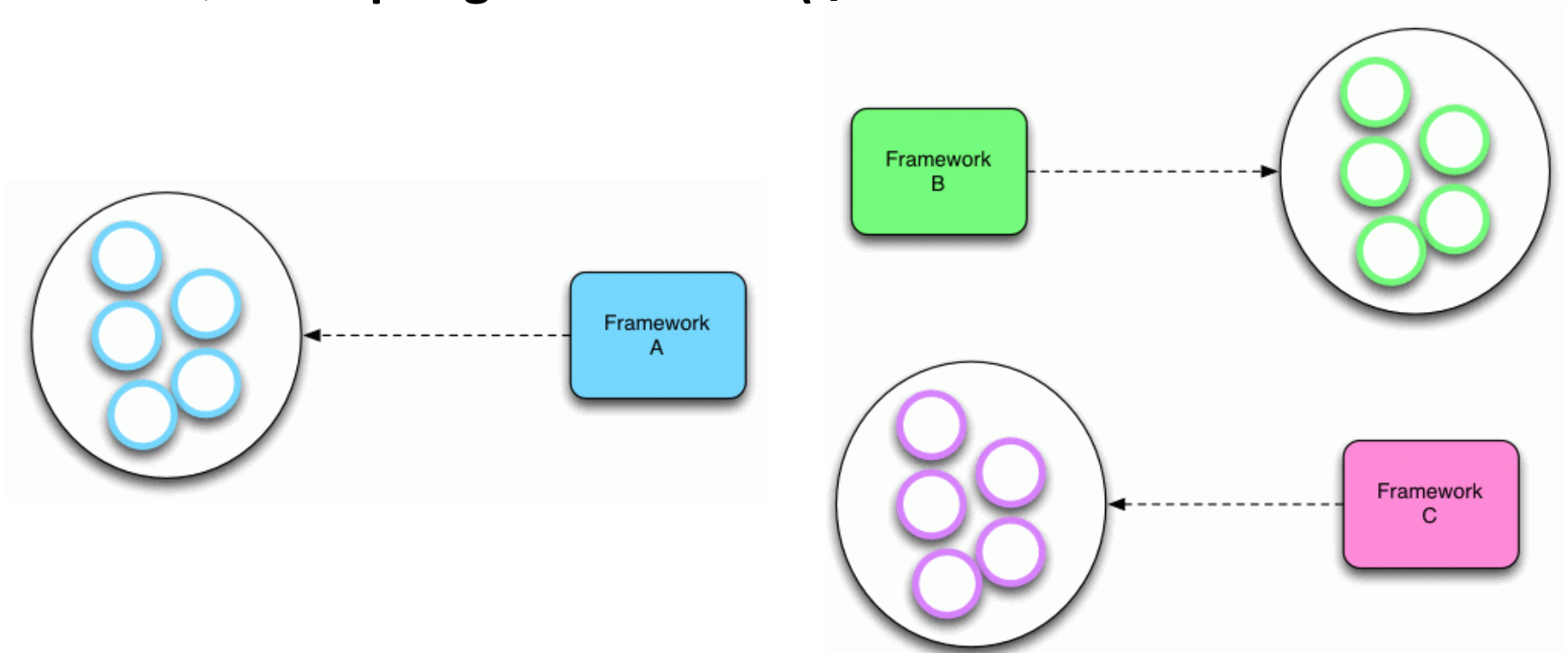
i.i ¿Qué es Spring Framework?(h)

- Es una plataforma de integración con otros frameworks como Struts, Hibernate, MyBatis, Quartz, Freemarker, así como todas las APIs Java EE.



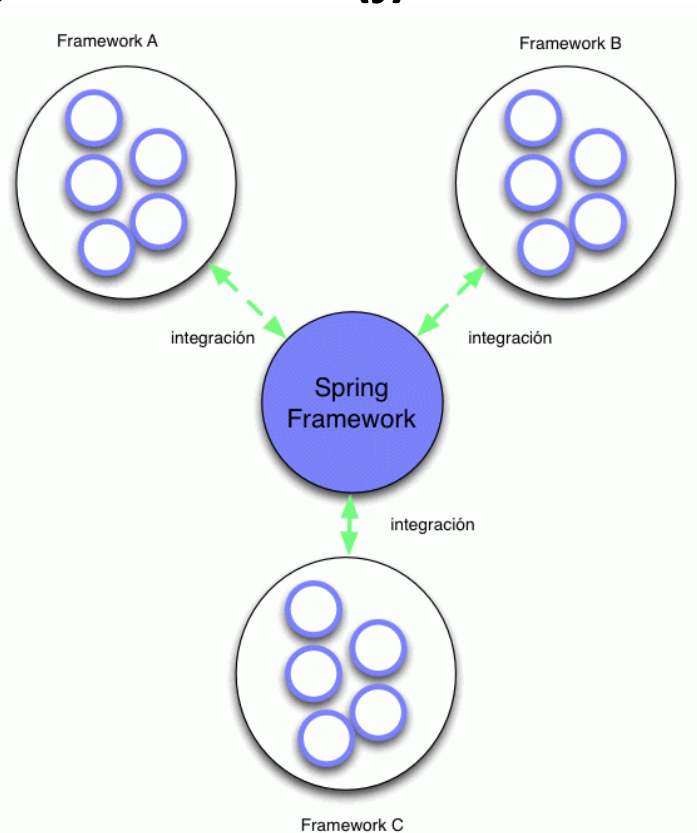
i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(i)



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(j)



a Spring Framework

i.i ¿Qué es Spring Framework?(k)

- Spring gestiona el ciclo de vida de los objetos de la aplicación.
- Spring gestiona los frameworks ya que cada framework gestiona su propio ciclo de vida en sus objetos.
- Promueve el desarrollo multi capa (N-Layer) orientado a componentes.



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(I)

- Spring extiende Java EE
- Total soporte Java EE
- Reutiliza, no reinventa la rueda



i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(m)

- Mucho de los conceptos que Spring Framework aporta no son simples y en su mayoría son complejos.
 - POO
 - AOP
 - Patrones de Diseño
 - SOLID
 - IoC
 - DI
 - etc...

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?(n)

- Spring Bean
- Es un POJO.
- Es un objeto configurado e instanciado en el contenedor de Spring.
- Necesita de el contenedor de Spring ya que éste provee de la mayoría de las características que Spring ofrece.
- Utiliza Inyección de Dependencias.

i. Introducción a Spring Framework

Resumen de la lección

i.i ¿Qué es Spring Framework?

- Comprendimos las características que provee Spring Framework
- Comprendimos los objetivos principales de Spring Framework
- Comprendimos por que Spring es buena opción para integrar Frameworks Java
- Comprendimos la diferencia entre POJO, Java Bean , EJB y Spring Bean

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

- a. POJOs
- b. JavaBeans
- c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

i.ii Motivación de Spring Framework

Objetivos de la lección

i.ii Motivación de Spring Framework

- Conocer y diferenciar el estilo de programación de EJB vs Spring Framework.
- Conocer la motivación de Spring Framework para el desarrollo de Aplicaciones Java EE.

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (a)

- El objetivo de Spring es ser un framework no intrusivo, reusable y mantenible para el desarrollo de aplicaciones Java EE a diferencia de EJB 2 y anteriores.

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (b)

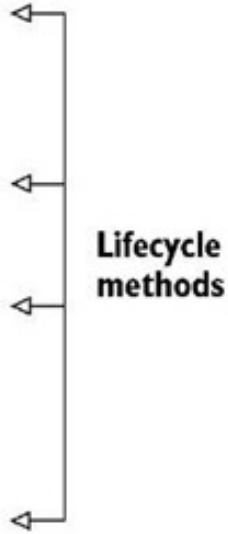
```
public abstract class UserBean implements EntityBean {  
    private EntityContext context;  
  
    public abstract String getUserId();  
    public abstract String getFirstName();  
    public abstract void setUserId(String userId);  
    public abstract void setFirstName(String firstName);  
    public abstract String getLastName();  
    public abstract void setLastName(String lastName);  
    public abstract String getUserType();  
    public abstract void setUserType(String userType);  
    public abstract Timestamp getBirthDate();  
    public abstract void setBirthDate(Timestamp birthDate);  
    public abstract Long getUserBillingId();  
    public abstract void setUserBillingId(Long userBillingId);  
}
```

Abstract
persistent
properties

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (c)

```
public String ejbCreate() {  
    return null;  
}  
  
public void ejbPostCreate() {  
}  
  
public void setEntityContext(EntityContext context)  
    throws EJBException {  
    context = context;  
}  
  
public void unsetEntityContext() throws EJBException {  
    context = null;  
}
```



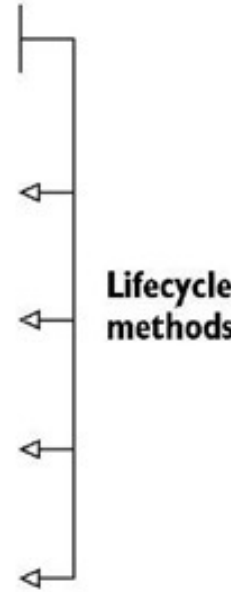
A vertical line with four left-pointing arrows groups the four methods listed on the left. To the right of this line is the text "Lifecycle methods".

Lifecycle methods

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (d)

```
public void ejbRemove() throws EJBException,  
    RemoveException {  
}  
  
public void ejbActivate() throws EJBException {  
}  
  
public void ejbPassivate() throws EJBException {  
}  
  
public void ejbLoad() throws EJBException {  
}  
  
public void ejbStore() throws EJBException {  
}  
}
```



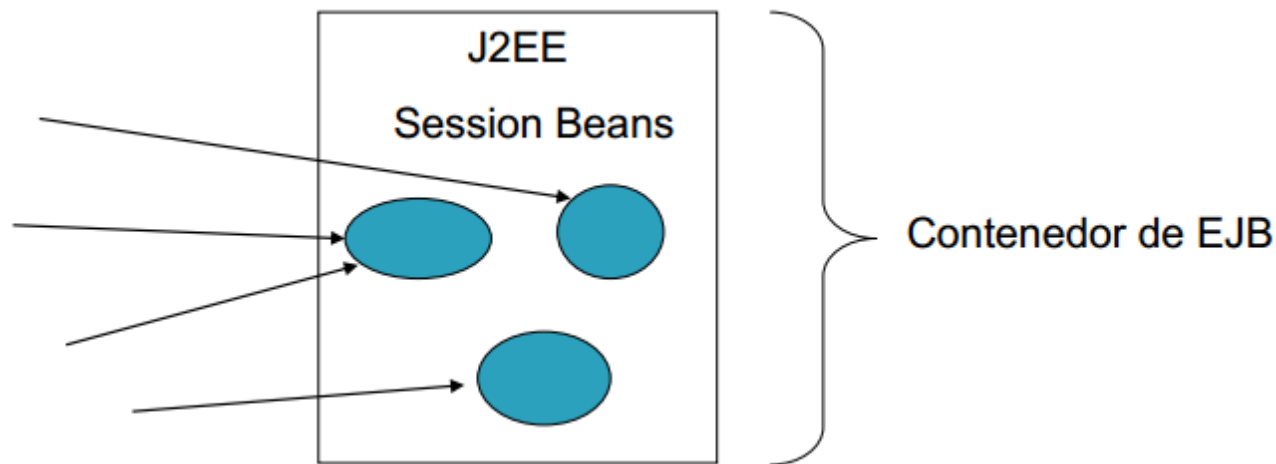
A vertical line with five horizontal arrows pointing left to the right-hand side of the code block. The arrows point to the lines containing `ejbActivate()`, `ejbPassivate()`, `ejbLoad()`, and `ejbStore()`. The text "Lifecycle methods" is positioned to the right of the line.

**Lifecycle
methods**

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (e)

- Java EE utilizando EJB se volvió difícil de mantener y probar.
- Java EE utilizando EJB se volvió dependiente del servidor de aplicaciones.



i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (f)

- Java EE estandariza la aplicación y uso de APIs como JNDI, JTA, JCA, JPA, etc.
- Un EJB es sólo una forma de implementar las APIs Java EE, de forma intrusiva.
- Spring Framework provee infraestructura no intrusiva para implementar APIs Java EE.
- EJBs requieren servidor de aplicaciones, Spring Framework no.

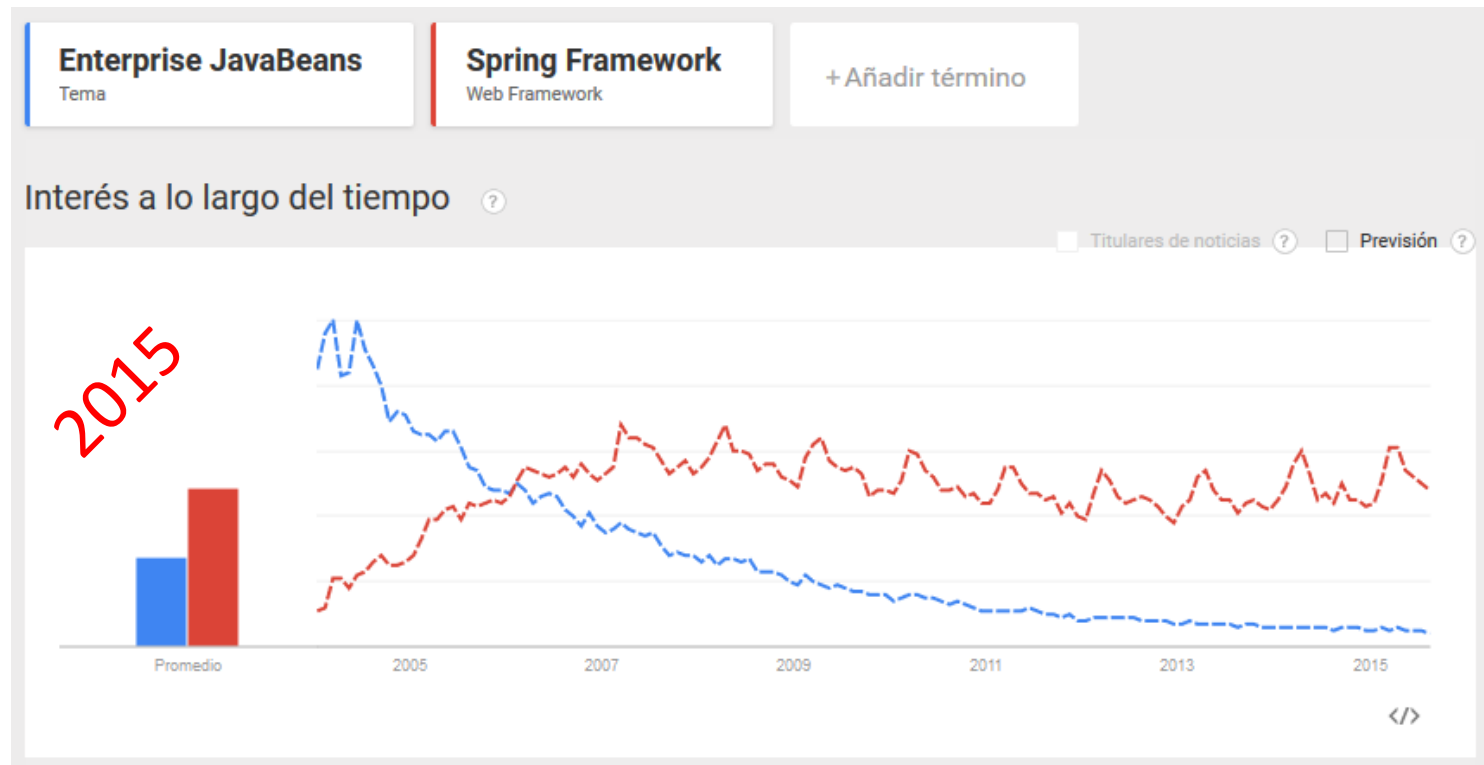
i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (g)

- Proveer una infraestructura simple, reusable, mantenible, extensible y robusta para la implementación de aplicaciones Java empresariales.
- Promover un diseño Orientado a Objetos enfocado al negocio y no a la infraestructura.
- Proveer un modelo de desarrollo ágil mediante el uso de buenas prácticas mediante patrones de diseño.

i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (h)



i. Introducción a Spring Framework

i.ii Motivación de Spring Framework (i)

● Enterprise JavaB ... Tema	● Spring Framework Aplicación	● EJB Término de búsqueda	● Java Persistence ... API	● Spring Boot Término de búsqueda
--------------------------------	----------------------------------	------------------------------	-------------------------------	--------------------------------------



i. Introducción a Spring Framework

Resumen de la lección

i.ii Motivación de Spring Framework

- Comprendimos el estilo de programación de EJB mediante implementación de interface vs POJOs de Spring Framework.
- Comprendimos los diferentes puntos de visión tanto de EJB como de Spring Framework.
- Comprendimos el concepto de programación no invasiva.

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

a. POJOs

b. JavaBeans

c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

i.iii Arquitectura y Módulos principales

Objetivos de la lección

i.iii Arquitectura y Módulos principales

- Conocer brevemente los diferentes módulos de Spring Framework.

i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (a)

- La arquitectura de Spring Framework es modular, no es requerido incluir Spring Framework completo.

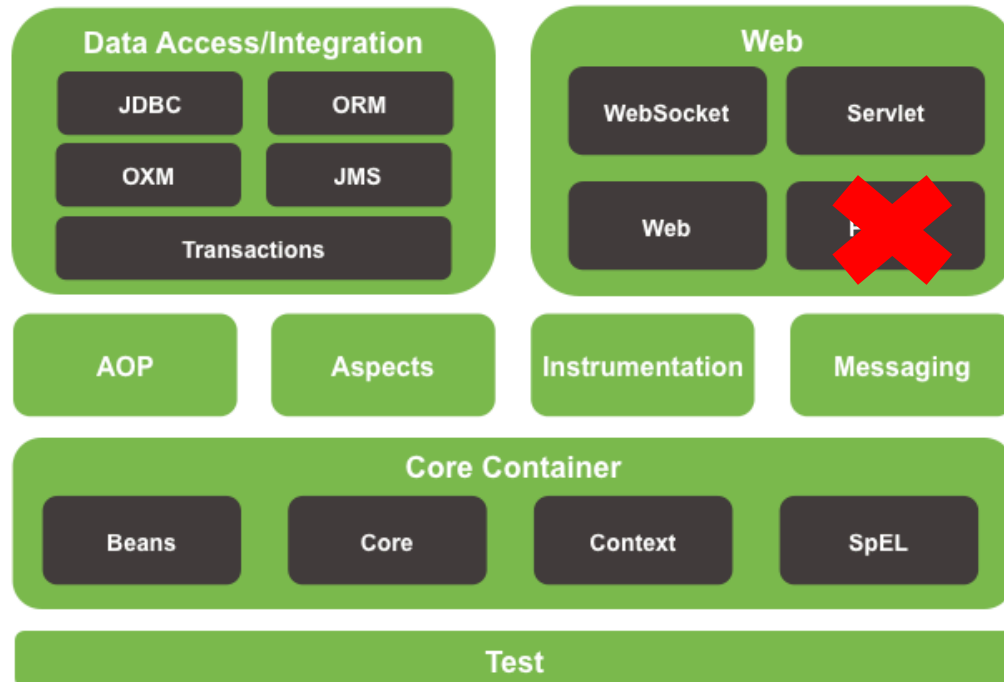


i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (b)



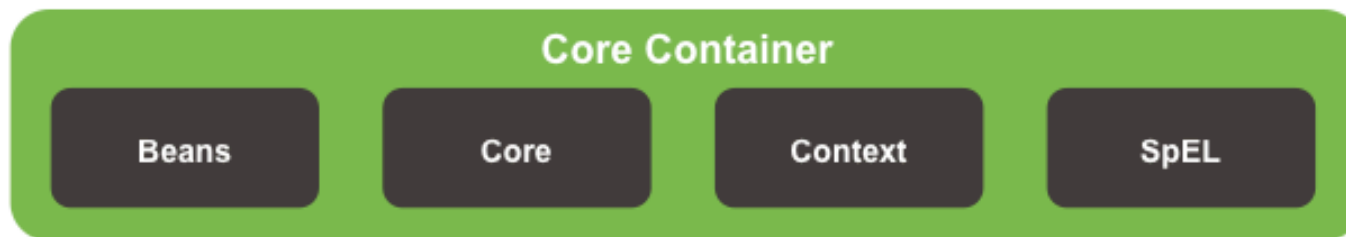
Spring Framework Runtime



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (c)

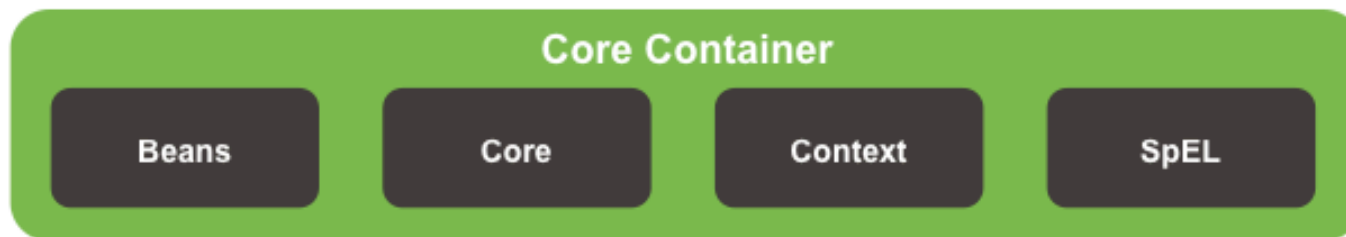
- Spring Core
 - Inversion of Control (IoC)
 - Inyección de Dependencias (DI)
- Beans
 - BeanFactory (factory pattern)



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (d)

- Context
 - Applciation Context (factory pattern)
 - JNDI, Internacionalización, integración EhCache entre otros
- SpEL
 - Spring Expression Language ej. `#{bean.value + bean2.value2}`



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (e)

- AOP
 - Implementación AOP alliance
 - Interceptores y pointcuts
- Aspects
 - Integración AspectJ

AOP

Aspects

i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (f)

- Instrumentation
 - Clases de soporte instrumentación e implementaciones classloader utilizadas en algunos servidores de aplicaciones.
- Messaging
 - Abstracciones de Spring Integration
 - Message, MessageChannel, MessageHandler
 - Soporte para mensajería

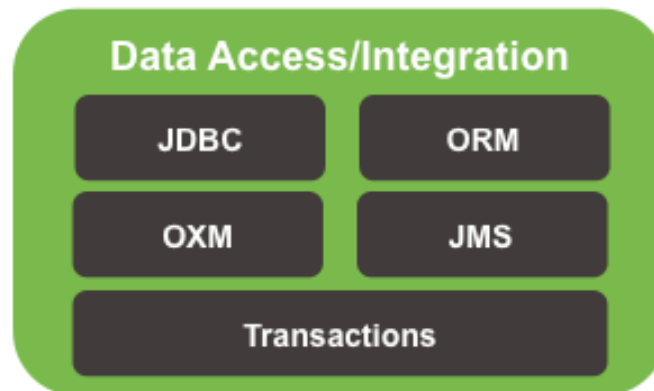
Instrumentation

Messaging

i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (g)

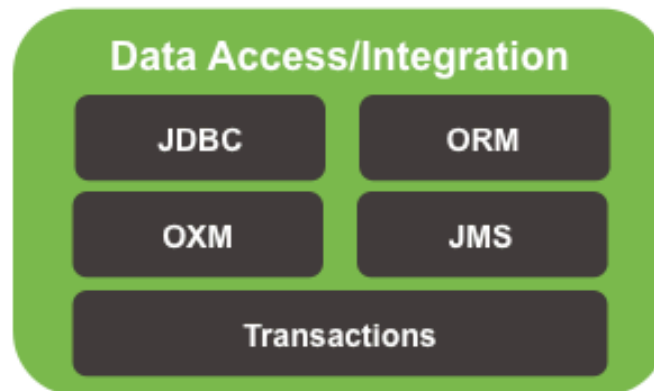
- Spring Data
- JDBC
 - Abstracción JDBC, JDBC Template (template pattern)



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (h)

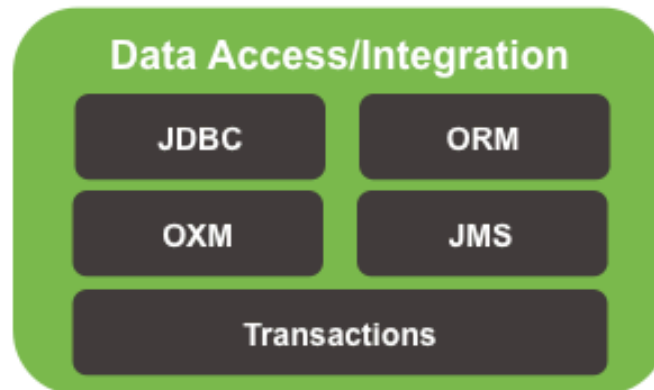
- Spring Data
- ORM
 - Integración Hibernate, JPA, JDO, Toplink, MyBatis



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (i)

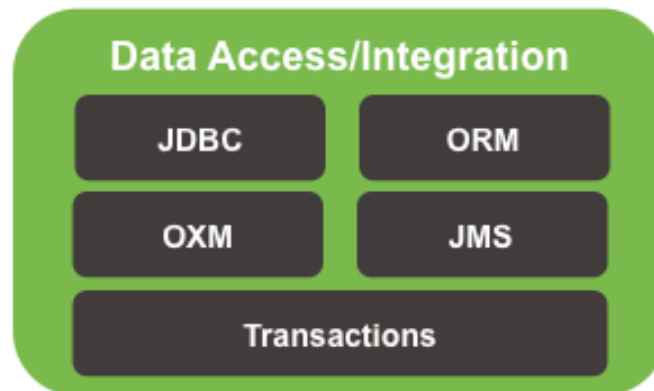
- Spring Data
- OXM
 - Abstracción Object XML Mappers, abstracción Castor, JAXB, Xstream, entre otros



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (j)

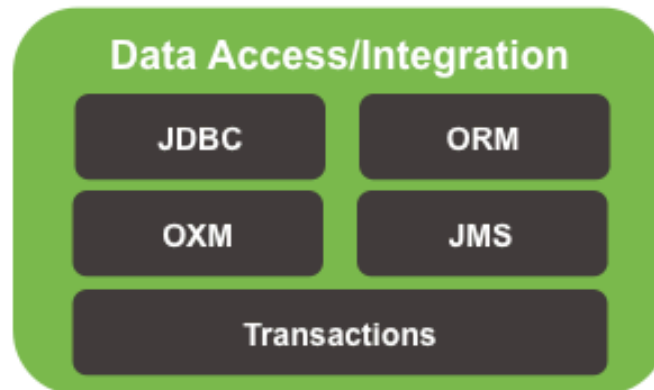
- Spring Data
- JMS
 - Integración Java Messaging Service, JMSTemplate (Template pattern)



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (k)

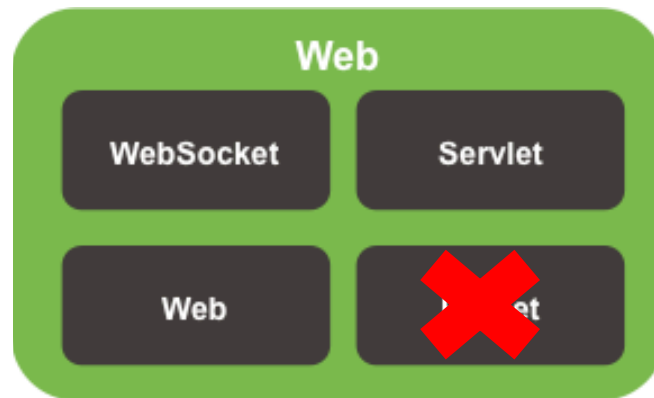
- Spring Data
- Transactions
 - ACID, Transacciones Locales y Globales y, Transacciones Programáticas y Declarativas



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (I)

- Spring Web
 - Spring Web (MVC)
 - Spring Web – Servlet
 - ~~— Spring Web – Portlet~~
- Spring WebFlux
- Spring WebSocket



i. Introducción a Spring Framework

i.iii Arquitectura y Módulos principales (m)

- Test
 - Soporte para pruebas unitarias y de integración para componentes Spring.
 - Soporte integración junit o TestNG.
 - Provee servicios de carga y cacheo de application context.
 - Soporte para mock objects con mockito.

Test

i. Introducción a Spring Framework

Resumen de la lección

i.iii Arquitectura y Módulos principales

- Conocimos el landscape de los módulos principales de Spring Framework.

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

a. POJOs

b. JavaBeans

c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

i.iv Programación orientada a interfaces

Objetivos de la lección

i.iv Programación orientada a interfaces

- Comprender el beneficio de la programación orientada a interfaces.
- Diferenciar entre interface y clase concreta.
- Analizar el diseño de interfaces.
- Comprender los términos alta cohesión y bajo acoplamiento.

i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (a)

- La programación orientada a interfaces se basa en utilizar interfaces, en lugar de clases concretas, para definir un conjunto de declaraciones que se deben cumplir.
- Las interfaces especifican el “¿que debe hacerse?” más no el “¿como debe hacerse?”.
- Para cada interface puede existir más de una implementación concreta diferente.

i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (b)

- Las programación orientada a interfaces promueve bajo acoplamiento.
- Las programación orientada a interfaces promueve disminución de código intrusivo.
- Las programación orientada a interfaces posibilita definir distintas clases concretas, implementaciones, que implementen la misma interface.
- Las programación orientada a interfaces facilita el intercambio de implementaciones en tiempo de ejecución.

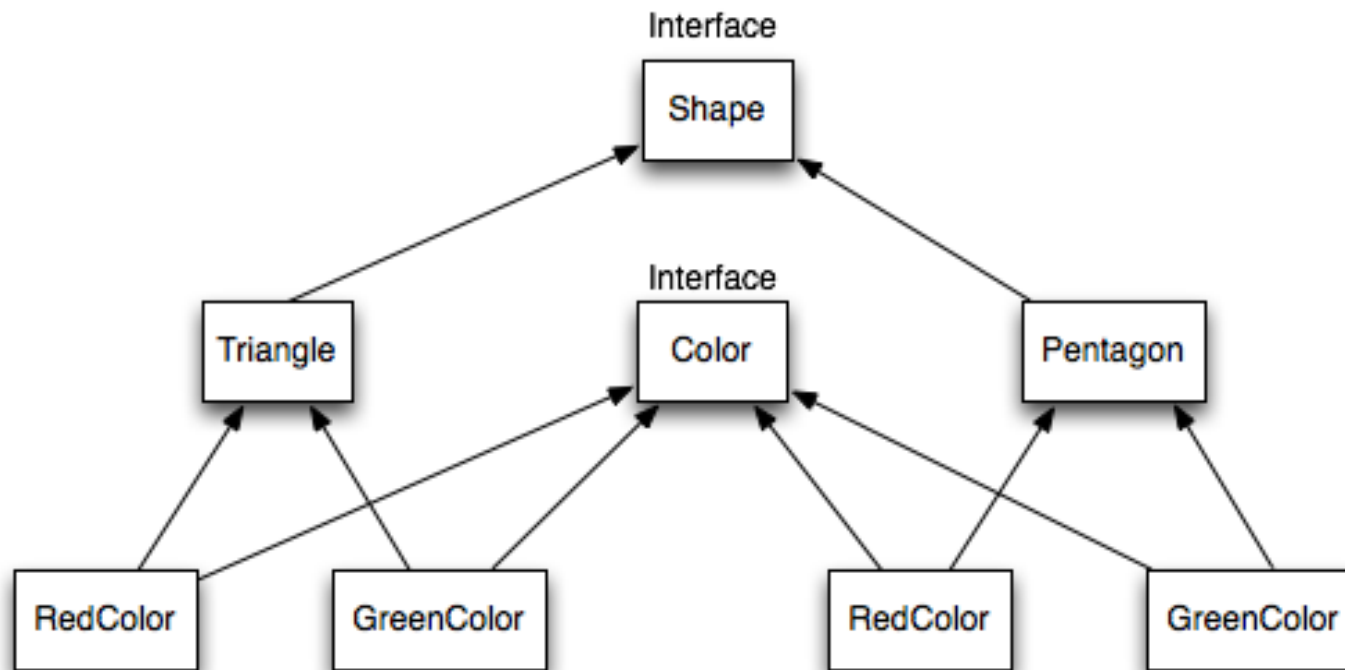
i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (c)

- Una interface es un conjunto de constantes y métodos abstractos cuya funcionalidad es:
 - Determinar el qué debe implementar una clase, determinar un API, determinar su funcionalidad.
 - Definir un “contrato” que estipula que métodos que deben implementar las clases que deseen cumplir con cierta API.
- Más información de interfaces consultar Anexo 1.

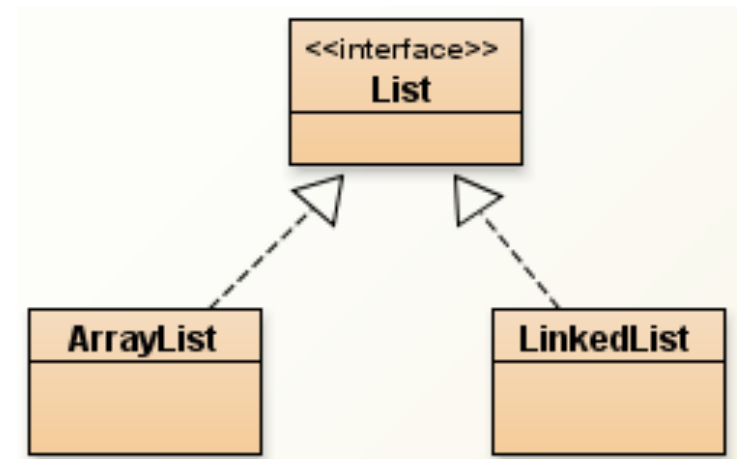
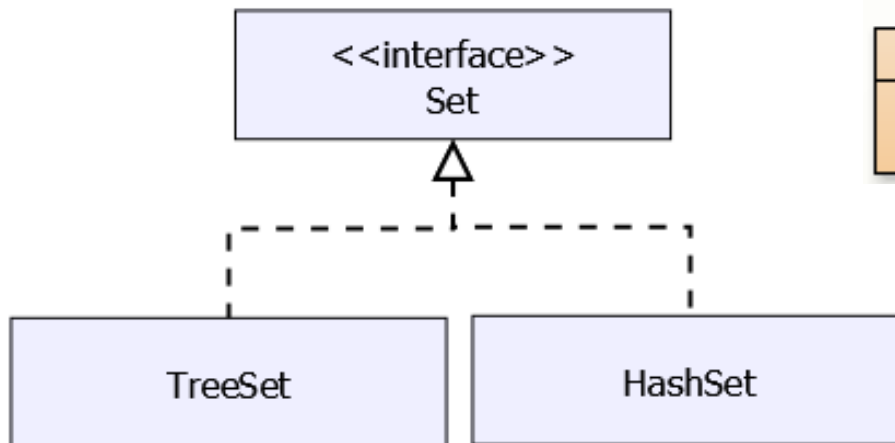
i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (d)



i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (e)



i. Introducción a Spring Framework

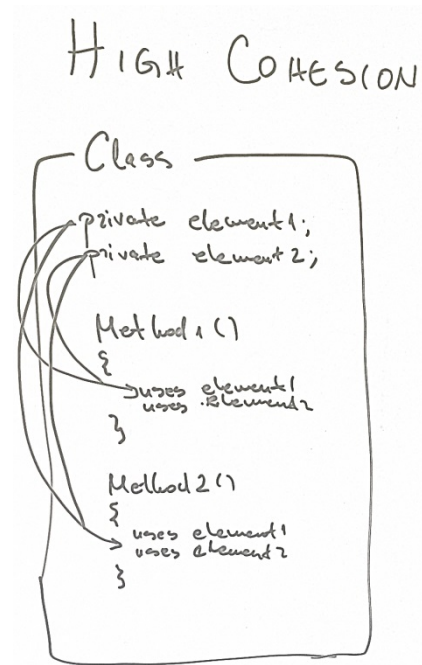
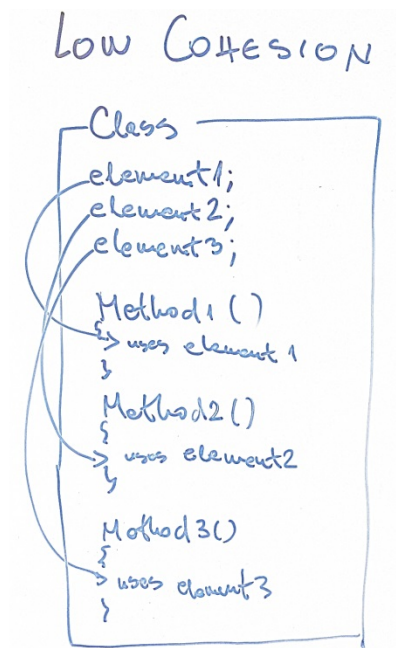
i.iv Programación orientada a interfaces (f)

- Alta cohesión
- La cohesión determina el grado de coherencia que una clase tiene con respecto a los atributos que almacena y a las funcionalidades que realiza, con respecto a las responsabilidades que la clase tiene.
- Alta cohesión es la medida en la que una pieza de software realiza sólo la tarea para la cual fue creada, delegando las tareas complementarias a otros componentes.

i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (g)

- Alta cohesión



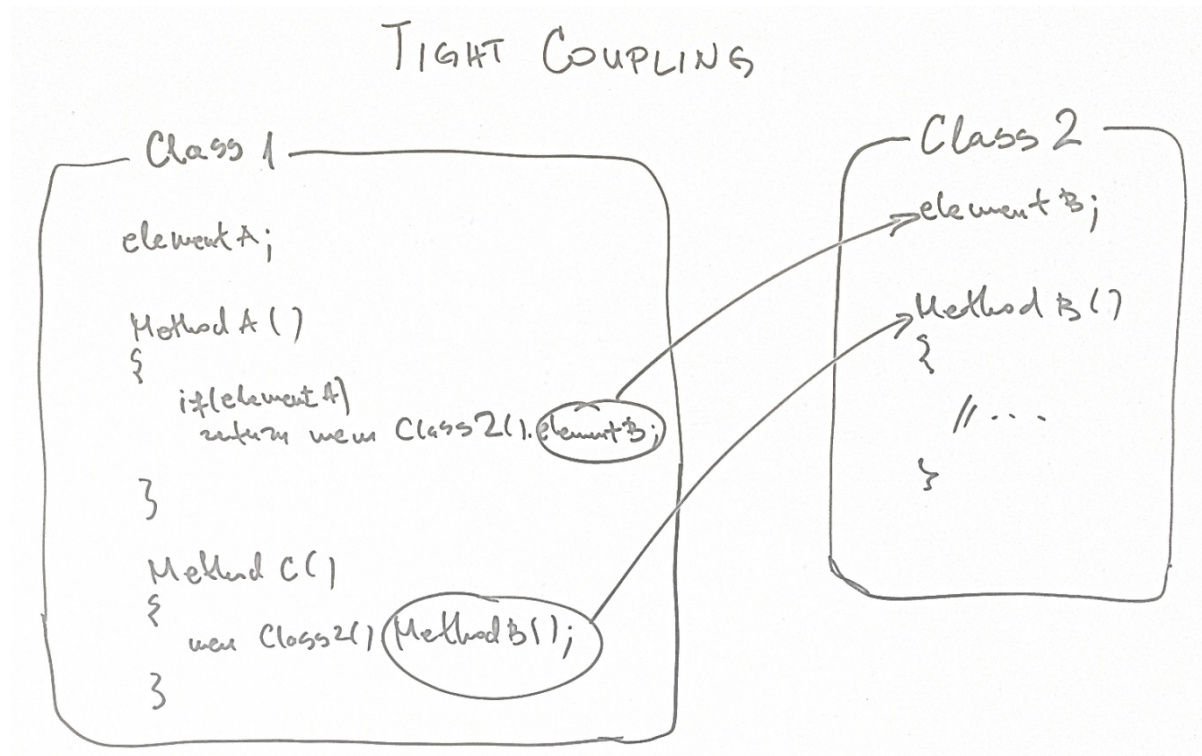
i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (h)

- Bajo acoplamiento
- Un bajo acoplamiento significa la poca o nula cantidad de detalle interno que una clase necesita conocer con respecto a otra(s) con las que colabore.
- El acoplamiento es el grado de dependencia o interconexión que tiene una clase con otra; es la idea de mantener dos o más clases lo menos ligadas entre sí de tal forma que en caso de producirse modificaciones en una de ellas, exista la mínima repercusión posible en las otras con las que colabore.

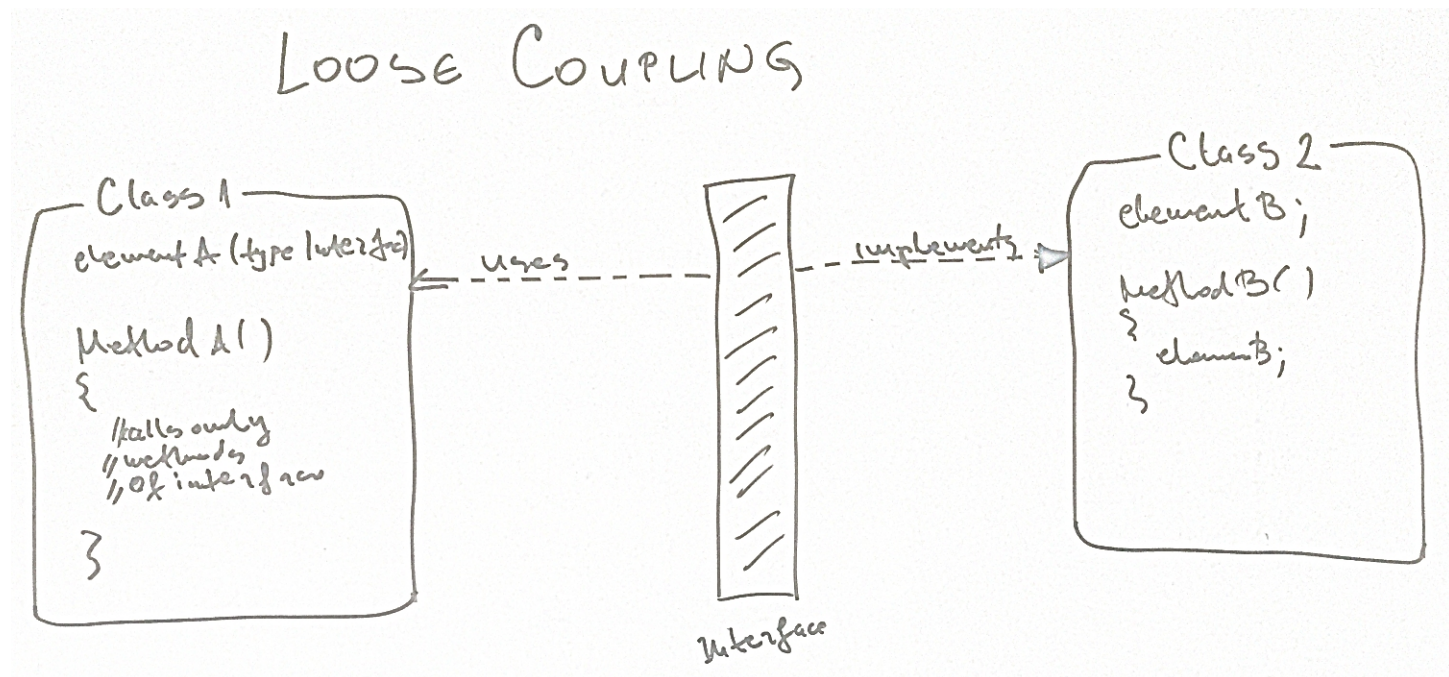
i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (i)



i. Introducción a Spring Framework

i.iv Programación orientada a interfaces (j)



i. Introducción a Spring Framework

Resumen de la lección

i.iv Programación orientada a interfaces

- Comprendimos el beneficio de implementar programación orientada a interfaces.
- Analizamos el concepto de interfaces y clases concretas.
- Comprendimos los términos alta cohesión y bajo acoplamiento.

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

a. POJOs

b. JavaBeans

c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

i.v Instalación ambiente de desarrollo

Objetivos de la lección

i.v Instalación ambiente de desarrollo

- Instalar el ambiente de desarrollo Spring Tool Suite STS (eclipse).
- Configurar variables de entorno JAVA_HOME y PATH

i. Introducción a Spring Framework

i.v Instalación ambiente de desarrollo (a)

Instalar Java JDK 8 o 9, de preferencia JDK 8.

link: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Instalar Spring Tool Suite STS. Se sugiere STS 3.9.x (versión reciente de preferencia)

link: <https://spring.io/tools/sts/all>

i. Introducción a Spring Framework

i.v Instalación ambiente de desarrollo (b)

Instalar Maven 3.2.5+, para ello descargar los binarios (zip) y descomprimir en cualquier lugar del equipo y agregar el comando mvn al PATH.

link: <http://www-us.apache.org/dist//maven/maven-3/3.2.5/binaries/>

Verificar que JAVA_HOME este bien asignada
C:\Program Files (x86)\Java\jdk1.8.0_201

Verificar que JAVA_HOME se encuentre en PATH
%JAVA_HOME%\bin;

Verificar que mvn se encuentre en PATH
C:\apache-maven-3.2.5\bin;

i. Introducción a Spring Framework

i.v Instalación ambiente de desarrollo (c)

Verificar que mvn se encuentre en PATH
C:\apache-maven-3.2.5\bin;

Ejecutar comandos **java** y **mvn** y, verificar que se encuentren disponibles.

Descargar e instalar Tomcat 8.5.x (Servlet API 3.0 o 3.1)

- Descargar el binario (.zip) y descomprimir.
- <https://archive.apache.org/dist/tomcat/tomcat-8/>

i. Introducción a Spring Framework

Resumen de la lección

i.v Instalación ambiente de desarrollo

- Instalamos y comprobamos el ambiente de desarrollo pertinente al curso.

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework

i. Introducción a Spring Framework

i.i ¿Qué es Spring Framework?

a. POJOs

b. JavaBeans

c. Spring Beans

i.ii Motivación de Spring Framework

i.iii Arquitectura y Módulos principales

i.iv Programación orientada a interfaces

i.v Instalación ambiente de desarrollo

Práctica 1. Definición y uso de Interfaces

Práctica 1. Definición y uso de Interfaces

Objetivos de la lección

Práctica 1. Definición y uso de Interfaces

- Realizar práctica sobre interfaces, introductoria al desarrollo de aplicaciones con Spring Framework.
- Comprender el principio Interface Segregation.

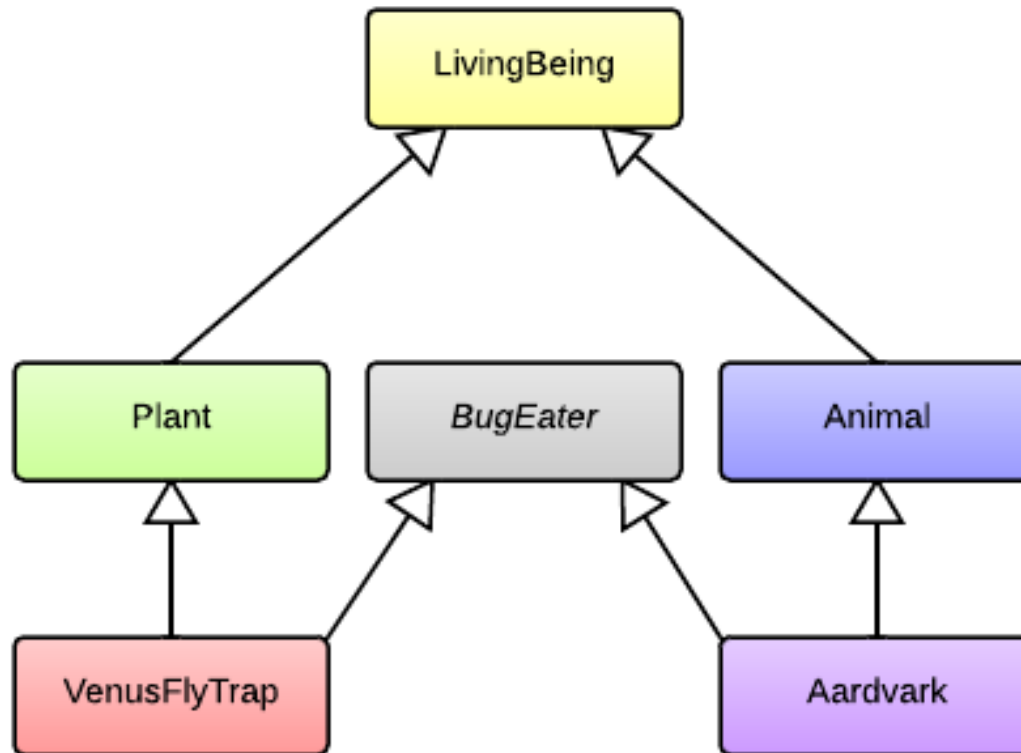
i. Introducción a Spring Framework

Práctica 1. Definición y uso de Interfaces (a)



i. Introducción a Spring Framework

Práctica 1. Definición y uso de Interfaces (b)



i. Introducción a Spring Framework

Práctica 1. Definición y uso de Interfaces (c)

- Crear ejercicio LivingBeing.
- Crear y analizar el ejercicio siguiente:

```
public interface Operaciones {  
  
    public static final double NUMERO1=3.6;  
    public static double NUMERO2=5.9;  
  
    double suma();  
    double resta();  
    double multiplicacion();  
    double division();  
  
}
```

i. Introducción a Spring Framework

Práctica 1. Definición y uso de Interfaces (d)

- ¿Es correcta la definición de la interface Operaciones?
- ¿Son válidos las definiciones de las constantes?
- ¿Qué sucederá si necesitamos implementar ésta interface en una clase Calculadora que sólo sume o reste?
- Analizar el principio Interface Segregation.

i. Introducción a Spring Framework

Resumen de la lección

Práctica 1. Definición y uso de Interfaces

- Realizamos práctica referente a interfaces.
- Se analizó el beneficio de utilizar programación orientada a interfaces.
- Analizamos el diseño de interfaces.
- Comprendimos el principio de Interface Segregation.
- Se implementó el principio de Interface Segregation.

i. Introducción a Spring Framework

Esta página fue intencionalmente dejada en blanco.

i. Introducción a Spring Framework