

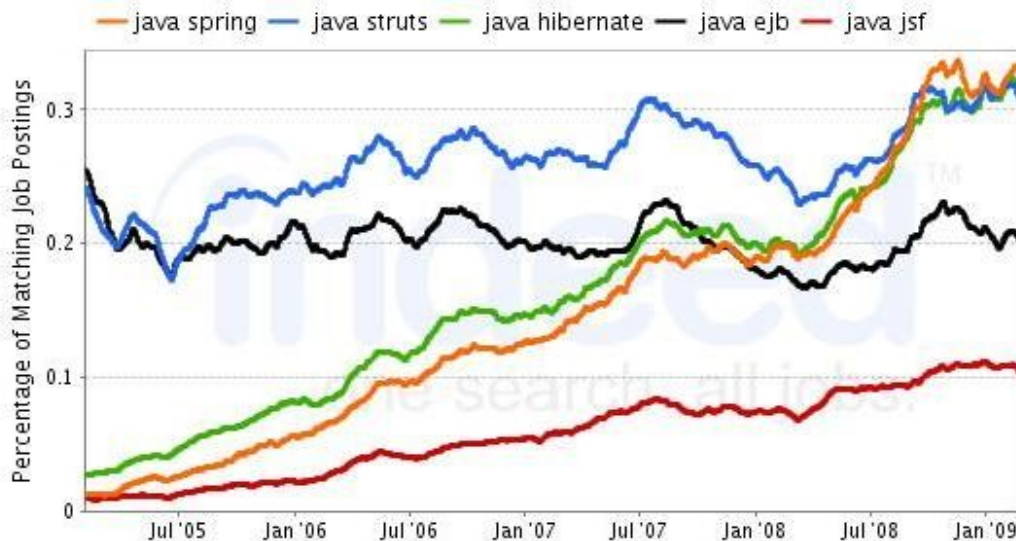
Spring

Introdução:

Spring é um framework que ajuda a estruturar a aplicação de uma maneira consistente, integrando diferentes frameworks em uma mesma arquitetura. Ele é dividido em três grandes partes: injeção de dependência, AOP e goodies (JDBC, MVC, transações...).

Em 1999/2000, quando as aplicações J2EE começavam a ser amplamente disseminadas, estas aplicações se mostravam complexas, exigindo grande esforço por parte do desenvolvedor e demonstravam baixa performance. A motivação para o Spring foi o ambiente J2EE, onde são oferecidos diversos serviços para uso nas aplicações.

Spring no mercado:



Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Utilizando injeção de dependência no Spring:

A injeção pode ocorrer de duas formas: setter ou construtor. Através da criação de um arquivo XML especificando geração de beans e quais propriedades serão carregadas.

Exemplo de Bean (Setter Injection):

```
<bean id="NovoBean" class="br.com.novo.NovoBean">
  <property name="cor">
    <value>vermelho</value>
  </property>
  <property name="numero" value="19"/>
</bean>
```

Exemplo de Bean (Constructor Injection):

```

<bean id="NovoBean " class="br.com.novo.NovoBean">
  <constructor-arg value="vermelho"/>
  <constructor-arg value="19"/>
</bean>

```

Para utilizar os beans especificados nos arquivos xml, basta carregar o contexto.

Carregando o contexto:

```

ClassPathXmlApplicationContext appContext =
new ClassPathXmlApplicationContext(new String[]{"applicationContext.xml"});

appContext.getBean("NovoBean");

```

Carregando o contexto via deployment descriptor:

```

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath:applicationContext.xml</param-value>
</context-param>

<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<listener>
  <listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>
</listener>

```

Utilizando AOP no Spring:

No fórum do Paca, anexamos a aplicação de exemplo número 3 que contém um exemplo de uso de AOP no Spring. Os passos abaixo estão exemplificados na aplicação.

1 - Identificar os conceitos ortogonais:

No nosso exemplo, são as demarcações das transações.

2 - Escolher os join points:

Vamos executar o advice “ao redor” da execução do método:

```
public void inserePessoa(Pessoa pessoa)
```

3 - Codificar o advice:

Nosso advice está implementado na classe aop.advice.MethodAdvice vide projeto de exemplo 3;
A classe implementa a interface MethodInterceptor.

4 - Criar o XML com as definições de AOP:

Está dentro do dispatcher-servlet.xml.

5 - Registrar o bean do advice:

```
<bean id="methodAdvice" class="aop.advice.MethodAdvice" />
```

6 - Criar o pointcut:

```
<aop:pointcut id="anyMethodPcd" expression="execution(* business.dao.*.insere*(..))" />
```

7 - Criar o aspecto:

```
<aop:advisor advice-ref="methodAdvice" pointcut-ref="anyMethodPcd" />
```

Referências:

- <http://www.springsource.org/>