

## TAREFA 16 – PADRÕES DE PROJETOS – PARTE 2

A injeção de dependência é um padrão de design amplamente utilizado na programação orientada a objetos e frameworks como o Spring facilitam a implementação desse padrão. No contexto do Spring Framework em Java, a injeção de dependência é um mecanismo que permite que as dependências de uma classe sejam injetadas em tempo de execução, em vez de serem criadas e gerenciadas pela própria classe.

Em vez de uma classe criar suas próprias dependências, ela simplesmente declara as dependências que precisa em seus construtores, métodos ou diretamente em seus campos (usando anotações). O contêiner de injeção de dependência do Spring é responsável por criar e injetar essas dependências.

Aqui está um exemplo básico de como a injeção de dependência funciona no Spring:

1. Defina a classe dependente (por exemplo, um serviço):

```
public class MeuServico {  
    private MinhaDependencia minhaDependencia;  
  
    public MeuServico(MinhaDependencia minhaDependencia) {  
        this.minhaDependencia = minhaDependencia;  
    }  
  
    public void executar() {  
        // Usa a minhaDependencia  
        minhaDependencia.metodo();  
    }  
}
```

2. Defina a classe de dependência (por exemplo, um repositório):

```
public class MinhaDependencia {  
    public void metodo() {  
        // Lógica do método  
    }  
}
```

3. Configure a injeção de dependência no arquivo de configuração do Spring (por exemplo, usando anotações):

```
@Configuration  
public class ConfiguracaoDoSpring {  
    @Bean  
    public MeuServico meuServico() {  
        return new MeuServico(minhaDependencia());  
    }  
  
    @Bean  
    public MinhaDependencia minhaDependencia() {  
        return new MinhaDependencia();  
    }  
}
```

Neste exemplo, o Spring irá criar uma instância de `MinhaDependencia` e injetá-la automaticamente no construtor de `MeuServico` quando uma instância de `MeuServico` for solicitada em algum lugar do código.

A injeção de dependência traz benefícios, como reduzir o acoplamento entre as classes, facilitar a testabilidade (porque as dependências podem ser facilmente substituídas por mocks durante os testes) e permitir uma maior flexibilidade no desenvolvimento de software.

Existem várias formas de configurar a injeção de dependência no Spring, incluindo o uso de anotações, XML de configuração, Java Config e outras. O exemplo acima utiliza a abordagem de configuração baseada em anotações.