

Dependency Injection

Análise e Desenvolvimento de Sistemas - 6º Período

Dependency Injection

- É um padrão de projeto que foca em **desacoplamento**
- Seu objetivo final é remover a dependência entre diferentes classes
- É um padrão de projeto mas não está catalogado dentro dos padrões GOF

Dependency Injection

- Queremos classes que buscam implementar **apenas suas responsabilidades e que não sejam responsáveis pela instanciação de outros objetos**
- Se conseguirmos garantir isso, iremos **desacoplar as classes**
- Garantir que os objetos tenham o **mínimo de dependências possível**

Dependency Injection

- Imagine uma class **A** que necessite usar um método da classe **B**
- Essa classe poderia internamente simplesmente **instanciar B e chamar seu método desejado**
- No entanto isso cria uma dependência muito forte da classe **A** com a **B**
- Além disso, a classe **A** passa a ser responsável pela instanciação de **B**

Dependency Injection

- A injeção de dependencia sugere uma nova maneira de utilizar instancias de outras classes dentro de uma classe específica
- Portanto, no lugar de instanciar, passamos como parâmetro do construtor a classe do objeto que queremos utilizar. Dessa maneira podemos usá-la como atributo. Dizemos então que estamos **injetando** essa dependencia na classe

Dependency Injection

- Uma outra maneira que podemos fazer é criar uma interface para a dependência e setar o tipo dela como sendo dessa interface. Isso funciona especialmente para linguagens tipadas
- Existe ainda uma terceira maneira, criando métodos setters e passando a dependência para esses métodos. **Apesar do python não possuir modificadores de acesso, podemos usar essa estratégia**

Exemplo Java - Classe com acoplamento

```
public class Pedido
{
    public void getPedidos(pedido){
        ///Retorna pedido
    }
}

public class Cliente
{
    Pedido meuPedido = new Pedido();
    public void ObterPedidos()
    {
        meuPedido.getPedidos(pedido);
    }
}
```

Exemplo Java - Injetando Dependência

```
public class Cliente
{
    public Cliente(Pedido meuPedido)
    {
        this.meuPedido = meuPedido;
    }

    public void ObterPedidos()
    {
        meuPedido.getPedidos(pedido);
    }
}
```


Exemplo Java - Injetando Dependência

```
public static void main(String[] args){  
    // Invertemos o controle da criação do objeto  
    // Injetamos a dependência no construtor  
        Cliente cliente = new Cliente(new Pedido());  
}
```

Dependency Injection

- Perceba que no exemplo anterior estamos fazendo uma injeção de dependência através de uma **inversão de controle**
- Isso significa que a classe **Cliente** não é mais responsável por controlar a criação do objeto **Pedido**, mas sim o código client
- Se for necessário mudar o tipo do Pedido (objeto), não precisamos modificar nossa classe **Cliente**, basta passar no código client o objeto desejado como parâmetro do construtor