

## Projeto no Github

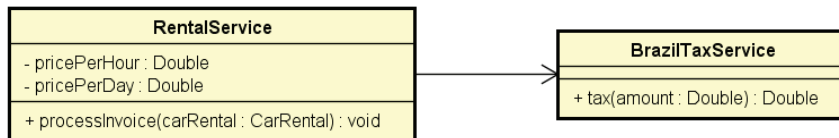
<https://github.com/acenelio/interfaces1-java>

## Inversão de controle, Injeção de dependência

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

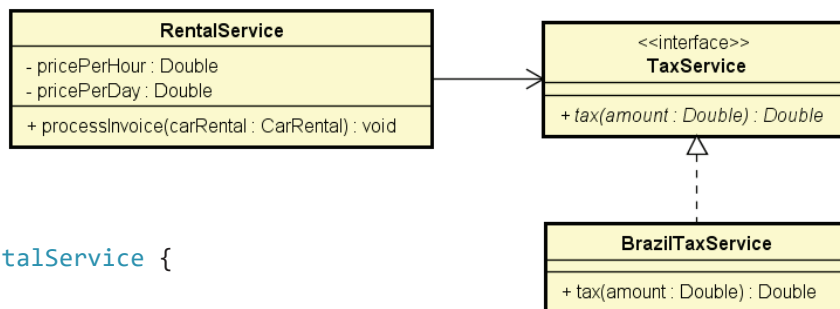
- Acoplamento forte
- A classe RentalService conhece a dependência concreta
- Se a classe concreta mudar, é preciso mudar a classe RentalService



```

class RentalService {
    (...)
    private BrazilTaxService taxService;
}
  
```

- Acoplamento fraco
- A classe RentalService não conhece a dependência concreta
- Se a classe concreta mudar, a classe RentalService não muda nada



```

class RentalService {
    (...)
    private TaxService taxService;
}
  
```

## Injeção de dependência por meio de construtor

```
class Program {  
    static void Main(string[] args) {  
  
        (...)  
  
        RentalService rentalService = new RentalService(pricePerHour, pricePerDay, new BrazilTaxService());  
    }  
}
```

```
class RentalService {  
  
    private TaxService taxService;  
  
    public RentalService(double pricePerHour, double pricePerDay, TaxService taxService) {  
        this.pricePerHour = pricePerHour;  
        this.pricePerDay = pricePerDay;  
        this.taxService = taxService;  
    }  
}
```

upcasting



## Inversão de controle

- **Inversão de controle**

Padrão de desenvolvimento que consiste em retirar da classe a responsabilidade de instanciar suas dependências.

- **Injeção de dependência**

É uma forma de realizar a inversão de controle: um componente externo instancia a dependência, que é então injetada no objeto "pai". Pode ser implementada de várias formas:

- Construtor
- Classe de instanciação (builder / factory)
- Container / framework