**Coesão e Acoplamento**

Quando falamos de ***acoplar*** uma coisa em outra coisa, estamos falando de conectar estas duas coisas. Uma Roda está acoplada a um Carro. Uma Bateria está acoplada a um Smartphone. Uma Cabeça está acoplada a um Corpo.

.



Na estrutura de um sistema temos acoplamento em todo lugar, em qualquer fase do projeto, em qualquer fragmento de escopo do software. Quando falamos, por exemplo, de relacionamento entre Classes, Tabelas, Domínios, Subsistemas, Casos de Uso etc. estamos falando de acoplamento.

Podemos afirmar que no contexto de um software qualquer relacionamento gera acoplamento.

Entendemos então que o acoplamento é algo determinístico ao processo de produção de software, ou seja, fatalmente estará presente neste tipo de atividade e em suas entregas. E tem que estar mesmo, pois as partes do todo precisam estar conectadas para se formar o todo (que no nosso caso, é um sistema).

Para viabilizar essa mensuração utilizamos duas formas de classificar o nível de acoplamento: ***Forte Acoplamento e Fraco Acoplamento.***

Quando um sistema possui entre seus componentes uma relação de interdependência fraca, significa que a dependência entre seus componentes é baixa, ou seja, estão acoplados, mas fracamente acoplados. Isso chamamos de Fraco Acoplamento.

Quando um sistema possui entre seus componentes uma relação de interdependência forte, significa que a dependência entre seus componentes é alta, ou seja, estão acoplados, mas fortemente acoplados. Isso chamamos de Forte Acoplamento.

**Quanto mais uma classe conhece ou depende de outras classes, maior é o grau de acoplamento entre elas**

**Em sistemas de forte acoplamento**

**- mudanças em uma classe forçam mudanças em outra classes relacionadas;**

**- fica mais difícil entender as classes isoladamente é mais difícil reusar as classes, já que elas dependem da presença uma das outras.**

No contexto de um software podemos entender que ***Acoplamento*** é uma medida “inter” componentes, e ***Coesão*** é uma medida “intra” componentes.

A primeira refere-se ao mundo externo do componente, como ele se inter-relaciona com os outros componentes do sistema. A segunda refere-se ao mundo interno do componente, como ele se inter-relaciona consigo mesmo.

Sempre ouvimos falar das duas juntas: ***Acoplamento e Coesão.***

Cada parte de um todo possui seu contexto único, seu escopo, suas responsabilidades. Na estrutura de um software, cada um de seus componentes possuirá coesão, é algo determinístico também. São partes de um todo, e cada parte possui um conteúdo.

Uma “parte vazia” é um contrassenso, pois não teria razão de existir. É necessário haver um “conteúdo” para uma parte, e havendo conteúdo, há coesão.

Um componente de software sempre possui alguma Coesão. Mas esta Coesão pode ser Alta ou Baixa (ou Fraca ou Forte, dependendo da literatura consultada***). E qual a diferença entre ambas?***

Um componente com Alta Coesão é um componente que possui apenas uma única responsabilidade, que possui em seu conteúdo/suas funções, apenas aquilo que realmente deve fazer.

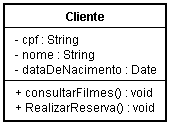
Ele não assume responsabilidades de outros componentes e não as mistura com as suas, nem delega suas responsabilidades a outros componentes, e não depende dos outros componentes para realizar suas funções conforme suas responsabilidades.

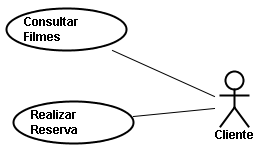
Já um componente com Baixa Coesão é o inverso disso. Possui responsabilidades além das suas, depende de outros componentes para realizar suas funções etc.

E o efeito da causa “Baixa Coesão” nos componentes de um software é semelhante ao efeito gerado pela causa “Forte Acoplamento”, onde podemos destacar:

* Dificuldades para dar manutenção no software – o grau desta dificuldade será proporcional à medida de Acoplamento e Coesão.
* Dificuldade de reuso de componentes – reuso não combina com responsabilidades misturadas, com falta de unidade.
* Dificuldades de interpretação clara dos modelos utilizados no projeto – gerando baixa performance na equipe devido ao custo adicional para entender o sistema.

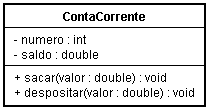
**Exemplo1: Baixo grau de Coesão**





* **A classe Cliente apresente um baixo grau de coesão (por tratar de assuntos diferentes: clientes e aluguel de filmes).**

**Exemplo2: Alto grau de Coesão**

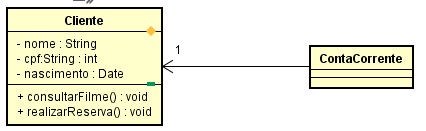


* **Nenhum outro “assunto” não relacionado a contas correntes é tratado por esta classe. Assim sendo, pode-se classificar esta classe como altamente coesa.**

**Os método não fazem sentido em um cenário bancário**

**Baixa coesão**

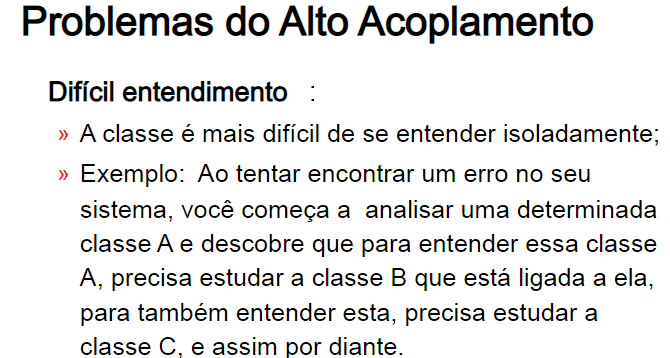
**Reuso da classe Cliente em um Sistema Bancário**

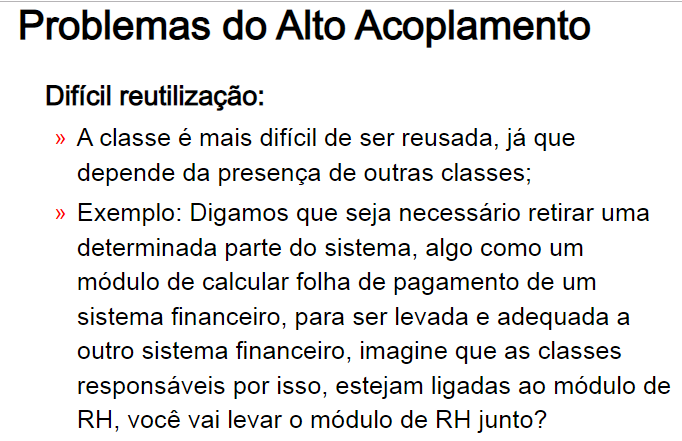


**Os método não fazem sentido em um cenário bancário**

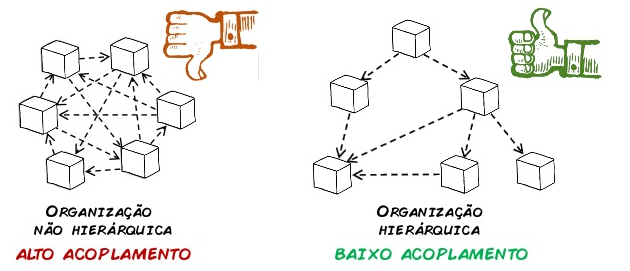
* Se esta classe for reutilizada em outro contexto, todos os seus atributos e sua interface pública (métodos) seriam úteis?
  + **Em caso negativo:**
    - **dividir a responsabilidade desta classe entre outras classes**
    - **definir subclasses que assumam parte desta responsabilidade;**

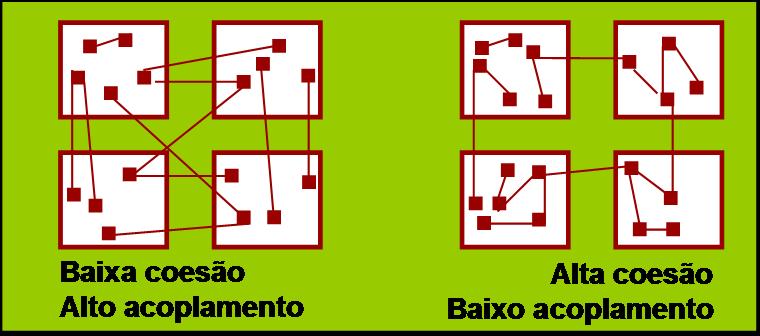
**Exemplos**





Quanto mais baixa for a coesão nos componentes de um software, mais problemático será mantê-lo.





Exercícios

1) Explique o que é acoplamento e coesão.

2) Quais são as vantagens de ter acoplamento fraco e coesão alta