

## Métricas para Programas Orientados a Objetos (CK)

Eduardo Figueiredo

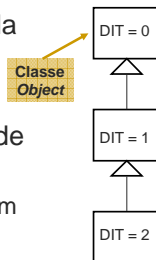
<http://www.dcc.ufmg.br/~figueiredo>

## Métricas de Programas OO

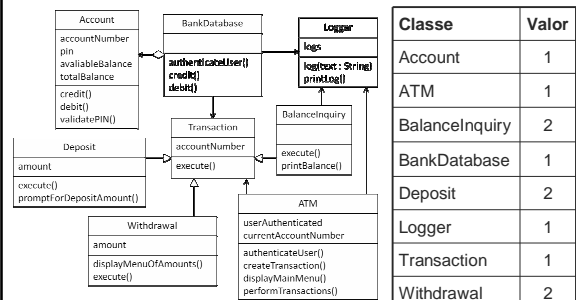
- Métricas de Chidamber-Kemerer (CK)
  - Profundidade da Herança (DIT)
  - Número de Filhos (NOC)
  - Acoplamento entre Objetos (CBO)
  - Resposta para uma Classe (RFC)
  - Falta de Coesão em Métodos (LCOM)
  - Métodos Ponderados por Classes (WMC)

## Profundidade de Herança (DIT)

- Representam o número de níveis que uma classe herda métodos e atributos
- Quanto maior a profundidade
  - Mais complexo o projeto
  - Mais difícil de se entender um módulo

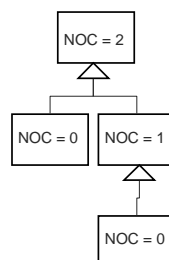


## Exemplo de DIT

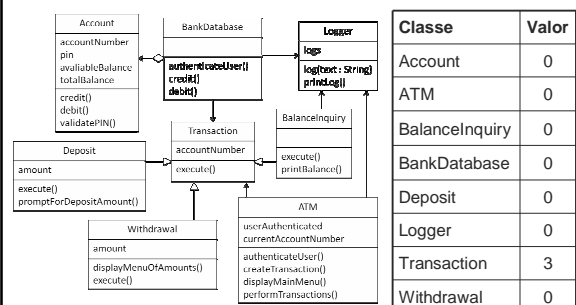


## Número de Filhos (NOC)

- Conta o número de subclasses diretas
  - Mede a largura da hierarquia de uma classe
- Valor alto, pode indicar maior reuso

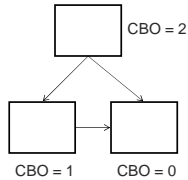


## Exemplo de NOC

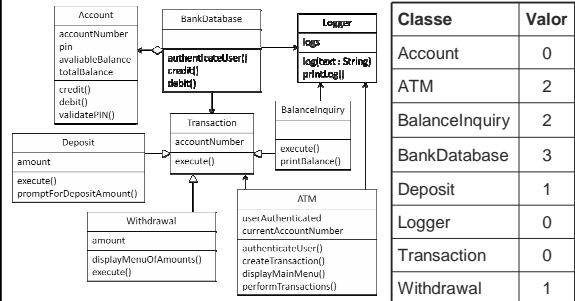


## Acoplamento entre Objetos (CBO)

- Conta as classes chamadas por uma classe
- Quanto mais acoplado uma classe
  - Mais difícil de entender e de manter



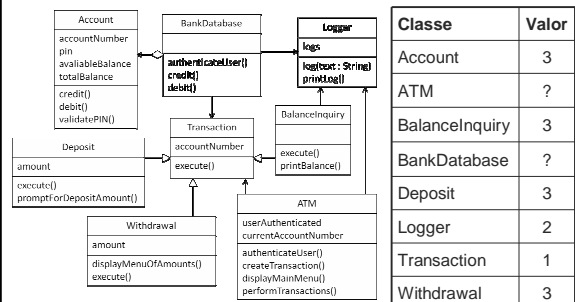
## Exemplo de CBO



## Resposta para uma Classe (RFC)

- Conta o número de métodos que podem ser executados em resposta a uma mensagem recebida por um objeto
  - Conta o número de métodos da classe mais o número de métodos chamados pelos métodos da classe
- Quanto maior o RFC, mais complexa é a classe

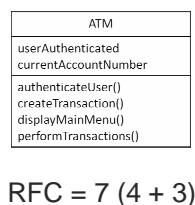
## Exemplo de RFC



## Exemplo de RFC

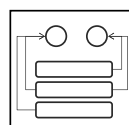
```

public class ATM {
    ...
    private void authenticateUser() { ... }
    private void performTransactions() {
        ...
        switch ( mainMenuSelection ) {
            case DEPOSIT:
                currentTransaction.execute();
                break;
            ...
            case LOG:
                Logger.printLog();
                break;
        }
    }
    private int displayMainMenu() {
        ...
        Logger.log("User option: " + option);
        return option;
    }
    private Transaction createTransaction() { ... }
}
    
```

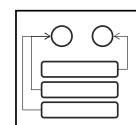


## Falta de Coesão (LCOM)

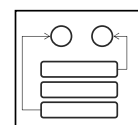
- Mede o quanto os métodos de uma classe acessam atributos em comum
  - Mais atributos em comum, maior coesão, menor perda de coesão (LCOM)



LCOM = 0 (1 - 2)

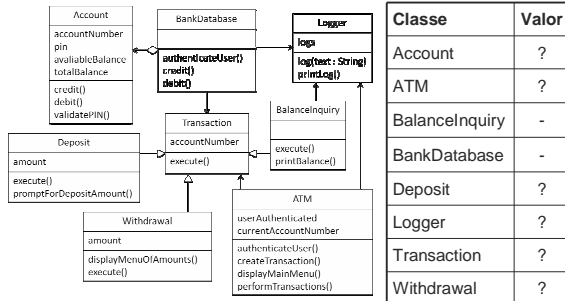


LCOM = 1 (2 - 1)



LCOM = 3 (3 - 0)

## Exemplo de LCOM



## Exemplo de LCOM

```
public class ATM {
    private boolean userAuthenticated;
    private int currentAccountNumber;
    ...

    private void authenticateUser() {
        if ( userAuthenticated ) {
            currentAccountNumber = accountNumber;
        }
    }

    private void performTransactions() {
        ... // Do not access variable in common
    }

    private int displayMainMenu() {
        ... // Do not access variable in common
    }

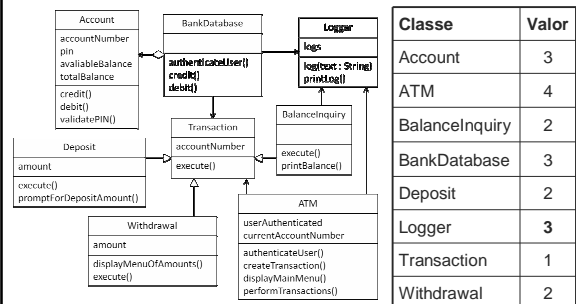
    private Transaction createTransaction() {
        Transaction temp = new BalanceInquiry( currentAccountNumber );
        ...
    }
}
```

LOCM = 4 (5 - 1)

## Métodos Ponderados por Classes

- WMC atribui pesos aos métodos de uma classe
- Pode-se pesar os métodos por
  - Linhas de código
  - Complexidade ciclomática
  - Número de parâmetros, etc.
- Valores altos indicam complexidade

## Exemplo de WMC



## Bibliografia da Aula

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
  - Cap. 24 Gerenciamento de Qualidade (Seção 24.4)