AspectJ: Pontos de Corte

Eduardo Figueiredo

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

AspectJ

- Linguagem orientada a aspectos mais madura e difundida
- Extensão simples da linguagem Java
  - Gera arquivos .class compatíveis com a máquina virtual Java (JVM)
- Possui bom suporte de ferramenta
  - AspectJ Development Tools (AJDT)
  - Integrado a plataforma Eclipse

Extensões de AspectJ

- Pontos de Junção
- Pontos de Corte
- Adendos
- Declarações Intertipo

Pontos de Junção

Modelo de Pontos de Junção

- Define como e onde será feita a junção entre classes e aspectos

- Exemplos
  - Chamada a métodos e construtores
  - Execução de métodos e construtores
  - Instanciação de objetos
  - Acesso e atualização de dados, etc.

Exemplos de Pontos de Junção

- Chamadas ao método **creditar** da classe **Conta**
  - call(void Conta.creditar(double))
- Acessos para leitura do atributo **numero** na classe **Conta**
  - get(String Conta.numero)

Quantificação

- Podemos usar coringas para indicar mais pontos de junção
  - Maior expressividade
- Tipos de coringas
  - ***** denota qualquer tipo e qualquer quantidade de caractere
  - **+** denota qualquer subclasse da classe
  - **..** denota qualquer quantidade de parâmetros

Exemplos de Quantificação

- Todas as chamadas a qualquer método da classe Cliente que tenham apenas um parâmetro e retorne void
  - **call(void Cliente.*(*))**

- Todas as chamadas a métodos começando com set e qualquer quantidade de parâmetros na classe Conta
  - **call(* Conta.set*(..))**

Outros Exemplos

- Chamadas a qualquer método set* da classe Conta e suas subclasses
  - **call(* Conta+.set*(..))**
- Acessos de leitura de qualquer atributo da classe Cliente com tipo String
  - **get(String Cliente.*)**
- Acessos de escrita a qualquer atributo de qualquer classe (de todo o sistema)
  - **set(* *.*)**

Ponto de Corte

Ponto de Corte

- Um ponto de corte é um conjunto de pontos de junção
  - Meio de identificar pontos de junção
- Pontos de junção podem ser compostos usando operadores lógicos
  - **&&** (*and*) intercepta um ponto quando ambas as condições são satisfeitas
  - **||** (*or*) intercepta um ponto quando uma das condições é satisfeita
  - **!** (*not*) intercepta todos os pontos que não estão negados na condição

Exemplos de Composição

- Chamadas à métodos set das classes Cliente ou Conta (anônimos)
  - **call(* Cliente.set*(*)) || call(* Conta.set*(*))**

- Ou equivalente (nomeados)
  - pointcut setCliente() : call(* Cliente.set*(*))
  - pointcut setConta() : call(* Conta.set*(*))
  - pointcut sets() : setCliente() || setConta()

Exemplo de *Logging*

```
public class Conta {
 private String numero;
 private double saldo;
 ...
 public void debitar (double valor) {
  if (this.getSaldo() >= valor){
      this.setSaldo(this.getSaldo()-valor);
      System.out.println("ocorreu um debito!");
 }
 public void creditar (double valor) {
      this.setSaldo(this.getSaldo()+valor);
      System.out.println("ocorreu um credito!");
 }...
}
```

Queremos separar o código em vermelho que implementa *logging*

Definindo Pontos de Corte

Palavra reservada que identifica um ponto de corte

```
pointcut logCredito():
        call (* Conta*.creditar(double));

pointcut logDebito():
        call (* Conta*.debitar(double));
```

Definindo Pontos de Corte

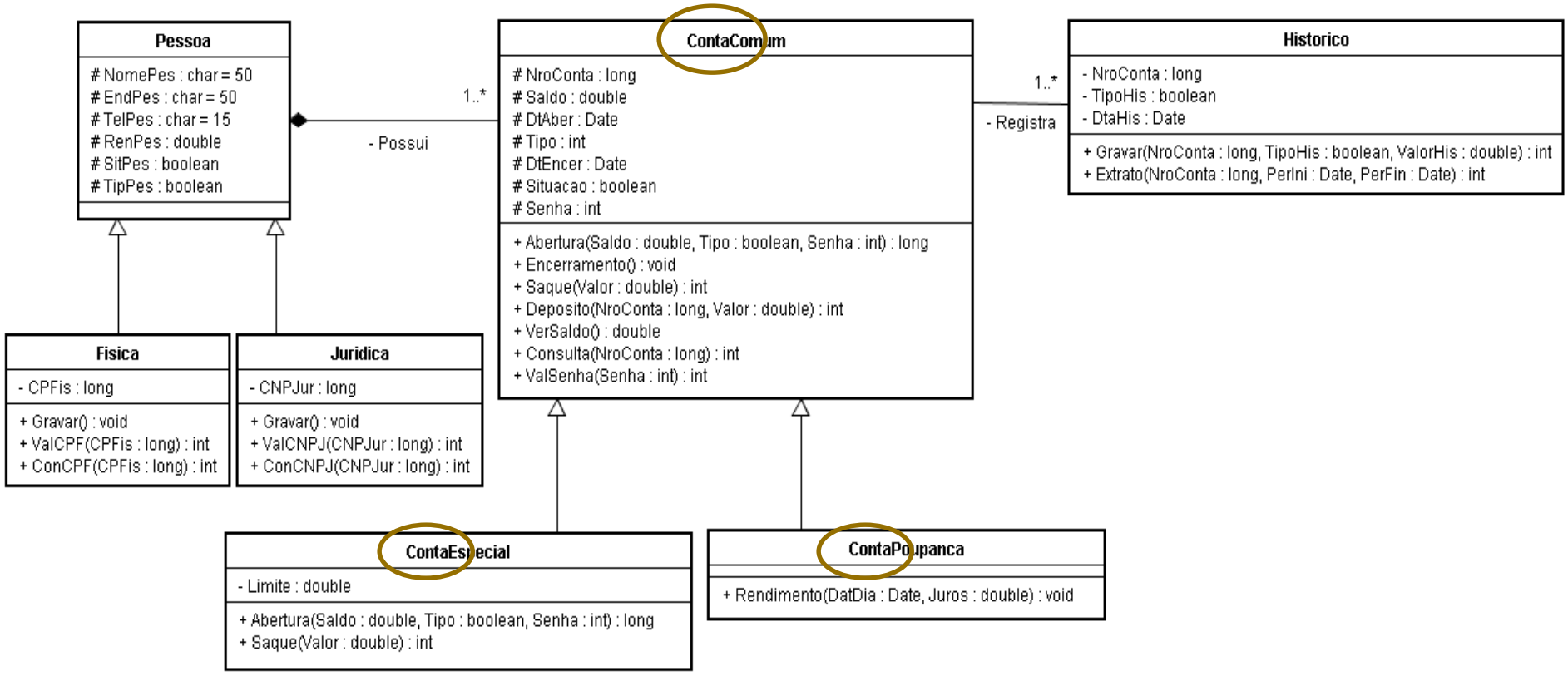
Todas as chamadas ao método creditar de todas as classes de nome começando com Conta

```
pointcut logCredito():
      call (* Conta*.creditar(double));

pointcut logDebito():
      call (* Conta*.debitar(double));
```

Todas as chamadas ao método debitar de todas as classes de nome começando com Conta

Exemplos de Contas



Bibliografia Principal

- R. LADDAD. **AspectJ in Action**, 2ª Ed. 2010.
  - Part 1 Understanding AOP and AspectJ

- Sergio Soares. **Programação Orientada a Aspectos com AspectJ**. Minicurso CBSoft 2010.