INSTITUTO FEDERAL
Rio Grande do Sul
Campus Erechim

AULA 04 - HERANÇA – Generalização e especialização

Linguagem de Programação I
Dário L. Beutler
dario.beutler@erechim.ifrs.edu.br

Objetivo da aula

- Entender conceitos ligados a HERANÇA na programação OO

Programa da aula

1) Hierarquia de Classes
2) Generalização
3) Especialização

Definição de Classes

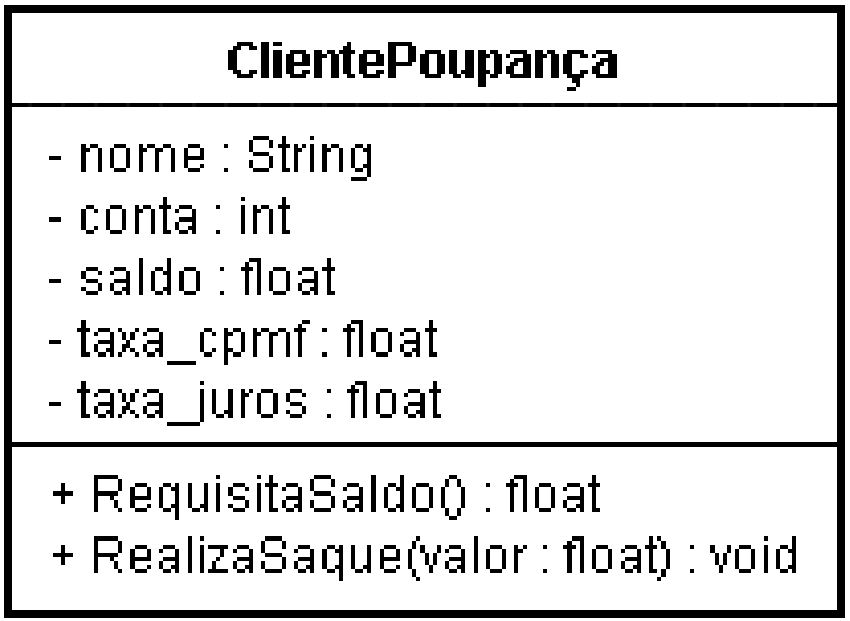
- Até então temos trabalhado com classes isoladamente;
- Por exemplo, 1 única classe foi suficiente para modelar o cliente de um Banco;
- Entretanto, podem existir situações em que 1 única classe não atenda às nossas necessidades;
- Exemplo: Diferença entre clientes conta corrente e conta poupança.

Diferença entre Tipos de Conta

- Um cliente conta corrente tem um saldo extra (chamado especial) além do seu saldo em conta;
- Um cliente poupança, diferentemente, não pode realizar uma retirada além do seu saldo;
- Uma vez por mês, o saldo de uma conta poupança é reajustado de acordo com a taxa da poupança no mês;
- Um cliente conta corrente pode utilizar cheques;

Classes Diferentes para Tipos Diferentes

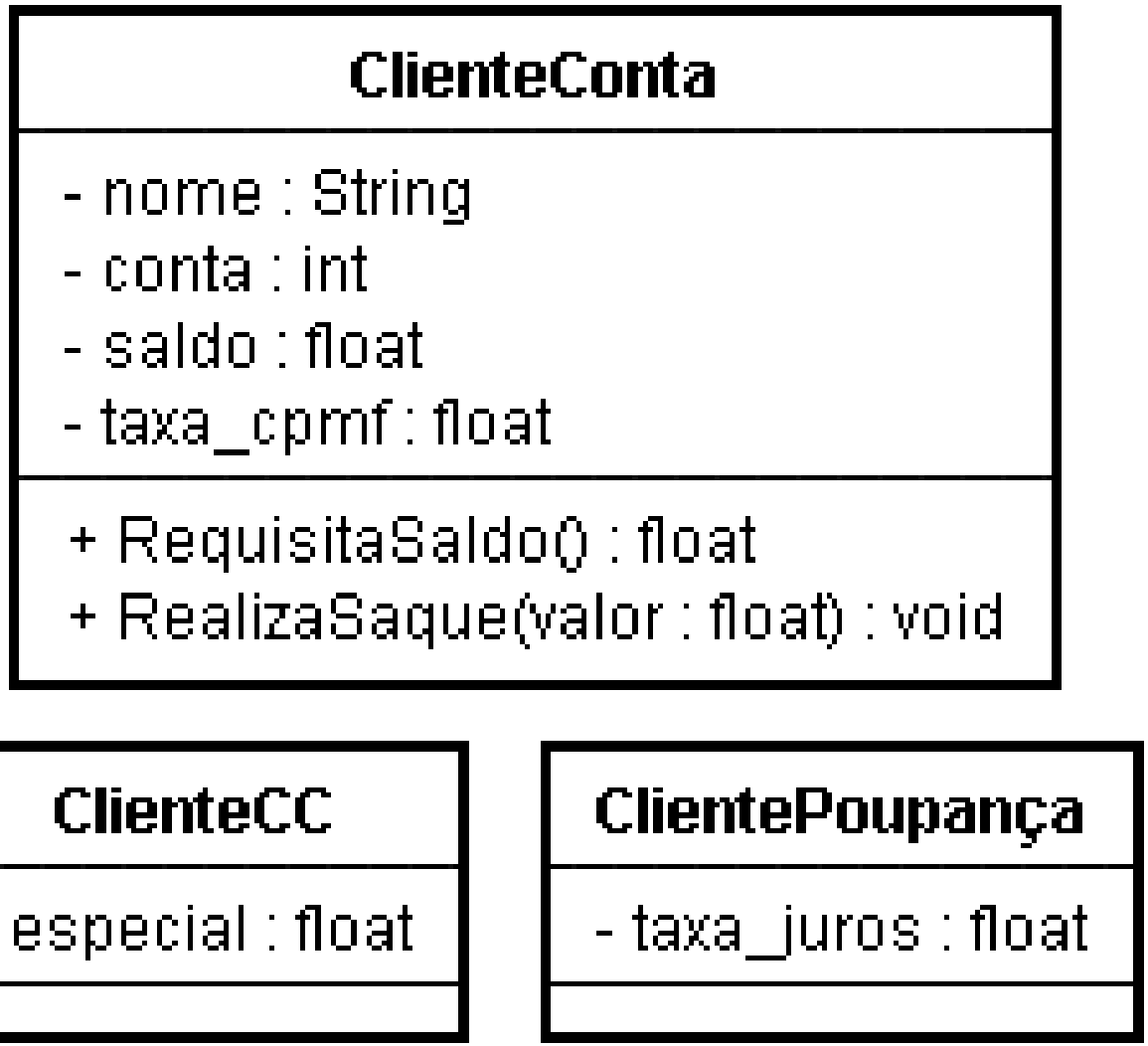
- Todas essas diferenças fazem com que não consigamos definir 1 única classe para clientes de um banco;
- Ou seja, podemos definir uma classe



Características em Comum

- Com as 2 classes não temos o inconveniente, por exemplo, de ter um cliente conta poupança com saldo especial;
- Entretanto, temos um valor de CPMF, único, declarado em 2 locais;
- Com isso, uma possível alteração na taxa implicará na atualização em 2 locais diferentes;
- Uma solução para este problema é colocarmos as características comuns num local comum entre as classes;

Classe com Características Comuns



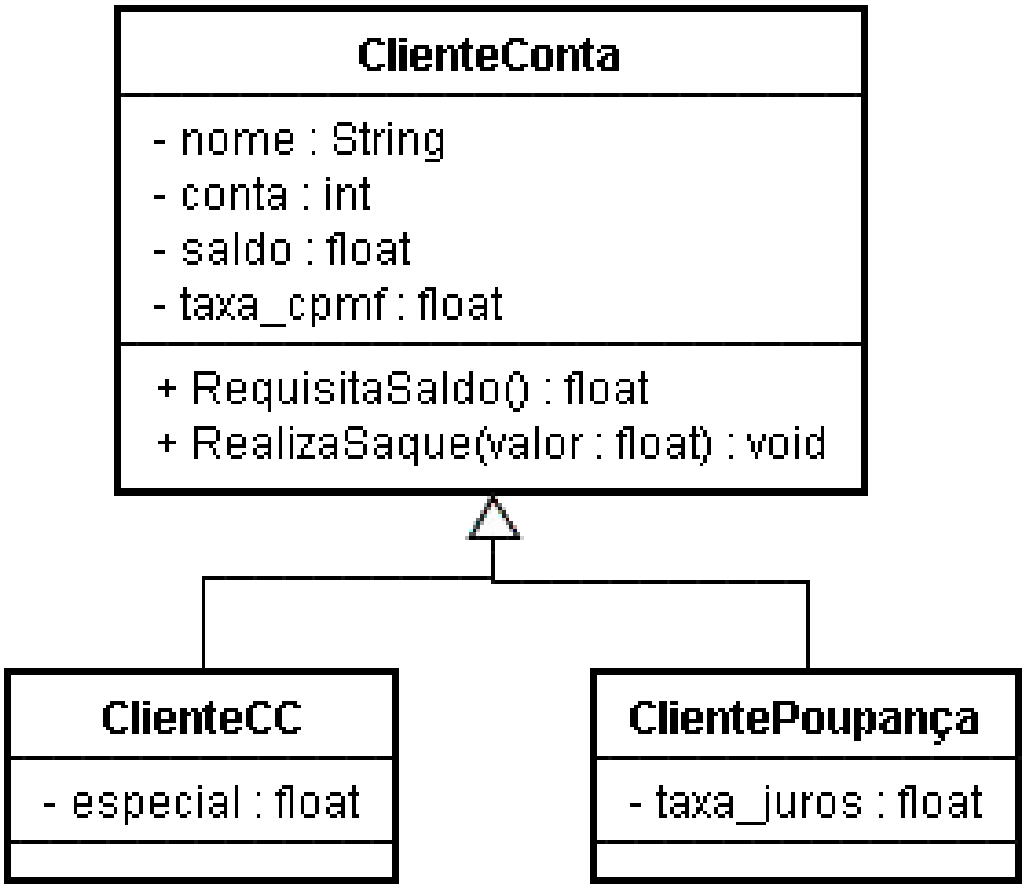
Analogia com a Matemática

- Na Matemática, usamos o termo "por em evidência" quando selecionamos partes comuns de uma expressão;

$$3xy + 6xz + 12x^2z = 0$$
$$3x(y + 2z + 4xz) = 0$$

- A expressão é a mesma; Foi apenas reescrita de forma diferente, mais limpa;
- Os parênteses são o artifício para se identificar a reescrita;

Analogia aos Parênteses



- Na classe ClienteConta colocamos “em evidência” os atributos comuns das classes ClienteCC e ClientePoupança;

Herança

- Um indivíduo da classe ClienteCC (ou ClientePoupança) possui os seus atributos + os atributos da classe comum;
- Dizemos que as classes ClienteCC e ClientePoupança herdam atributos e operações da classe ClienteConta;
- Ou seja, a classe ClienteConta é o ponto comum entre as classes ClienteCC e ClientePoupança;

Definições de Termos OO

- A classe ClienteConta é chamada de classe mãe, superclasse das classe ClienteCC e ClientePoupança;
- As classes ClienteCC e ClientePoupança são chamadas de subclasses ou classes filhas da classe ClienteConta;
- As classes ClienteCC e ClientePoupança são especializações da classe ClienteConta;
- A classe ClienteConta é uma generalização das classes ClienteCC e ClientePoupança;

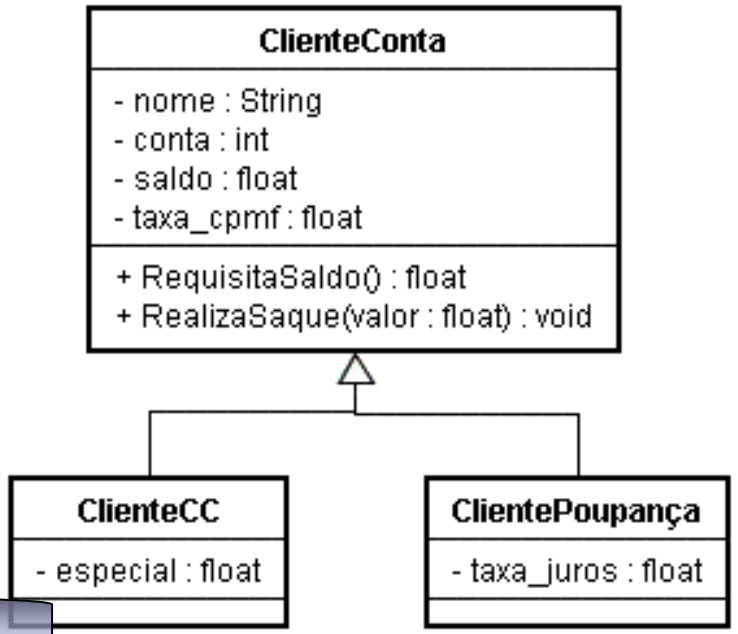
Definições de Termos OO

- Com isso, uma instância de uma subclasse contém os atributos e operações declarados nesta subclasse + os declarados em sua superclasse;

Herança em Java

```
public class ClienteConta {
      String nome; int conta;   float saldo;
      static float taxa_cpmf;

      public void RealizaSaque (float s) {
            if (s <= saldo)
            saldo = saldo – s;
      }
      public float RequisitaSaldo() {
         return saldo;
      }
}
public class ClienteCC extends ClienteConta {
      float especial;
      public void RealizaSaque (float s) {
      if (s <= (saldo + especial))
            saldo = saldo – s;
      }
}
public class ClientePoupanca extends ClienteConta {
      static float taxa_juros;
}
```



Herança em Java – Construtores

```
public class ClienteConta {
      String nome; int conta;
      float saldo; static float taxa_cpmf;
      ClienteConta (String pNome,
                          int pConta,
                          float pSaldo) {
            nome = pNome;
            conta = pConta;
            saldo = pSaldo;
      }
}
public class ClienteCC extends ClienteConta {
      float especial;
     public ClienteCC (String pNome, int pConta, float pSaldo,
            float pEspecial) {
            super(pNome, pConta, pSaldo);
            this.especial = pEspecial;
      }
}
```