Universidade Federal do Cariri

INTRODUÇÃO À POO

Paola Accioly

Material baseado nos slides de Thaís Alves Burity Rocha

Na aula passada vimos

- Processo de compilação + interpretação de programas Java
- Conceitos básicos de programação em Java
  - Variáveis
  - Tipos primitivos
  - Operadores matemáticos, relacionais e lógicos
  - Estruturas de controle condicionais e de repetição

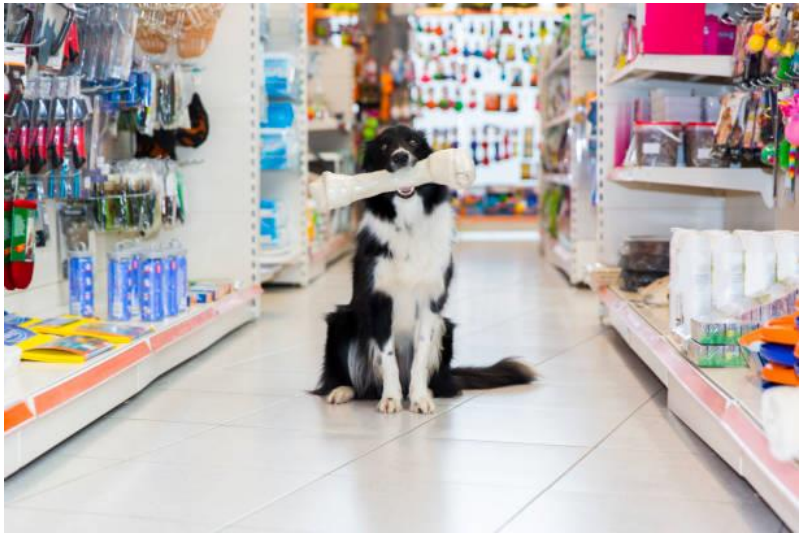
Hoje veremos

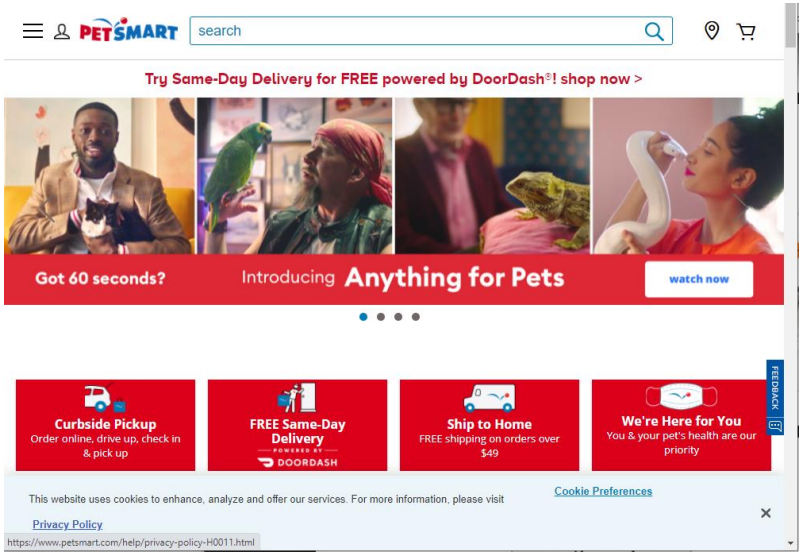
- Paradigma OO
- Classes
- Objetos
- Atributos
- Encapsulamento
- Métodos

Introdução

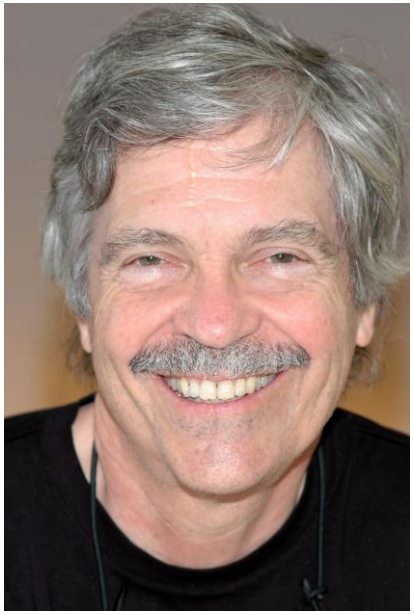
- Um paradigma de programação é um **estilo** de programação, caracterizado por uma seleção particular de **conceitos-chave**
- Conceitos-chave de POO
  - Abstração, classe, objeto
  - Encapsulamento
  - Ocultação de informação
  - Herança
  - Polimorfismo

Objetivo: Aproximar o mundo digital do mundo real





Idéia – Alan Kay



"O computador ideal deveria funcionar como um organismo vivo, isto é, cada "célula" comportar-se-ia relacionando-se com outras a fim de alcançar um objetivo, contudo, funcionando de forma autônoma. As células poderiam também reagrupar-se para resolver um outro problema ou desempenhar outras funções"

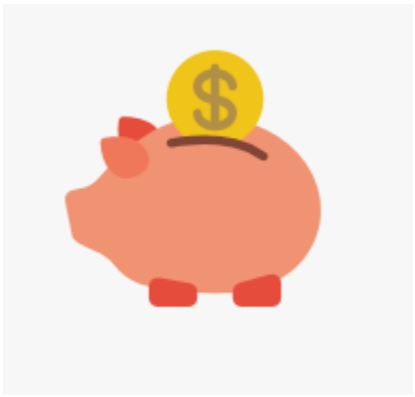
Paradigma OO

- Ênfase nos **dados**
- Dados são modelados primeiro
- Dados mudam através das operações
- Entidades do mundo real são vistas como **objetos**
- Objetos encapsulam dados e executam operações
- Um programa consiste em vários objetos que se comunicam

Vantagens

- Facilidade de manutenção de código
  - Entender a parte é melhor que entender o todo
- Maior capacidade de reuso e de código
  - É possível preservar código anterior já testado e apenas adicionar ou personalizar comportamentos
- Evolução sobre o paradigma procedimental
  - Foco maior nas funcionalidades do que nos detalhes de implementação

O que é um objeto?



3

instanciação

```
modelo Cachorro

início do modelo
    dado tamanho
    dado raca
    dado nome

    operação latir()
      início
        reproduz latido
      fim
fim do modelo
```

1

abstração

Tamanho	Raça	Nome
40 cm	Pug	Jesuíno

2

formalização

Objeto

3

instanciação

```
modelo Cachorro

início do modelo
    dado tamanho
    dado raca
    dado nome

    operação latir()
      início
        reproduz latido
      fim
fim do modelo
```

Classe

1

abstração

Tamanho	Raça	Nome
40 cm		Jesuíno

Dados

2

formalização

Classes e objetos

- Classes são **projetos** para a criação de objetos
- Classes definem a estrutura e comportamento de objetos semelhantes
- Classes definem **categorias** e **tipos de dados**
- Um objeto é uma **instância** de uma classe

Atributos e métodos

- Ao projetar uma classe, pense nos objetos que serão criados
  - Quais suas características?
  - O que eles fazem?
- Atributos (variáveis de instância)
  - Dados encapsulados em um objeto
  - Cada objeto possui seus próprios valores
- Métodos
  - São as operações que o objeto pode executar
  - Manipulam os atributos

Objeto Cachorro

Atributos

Métodos

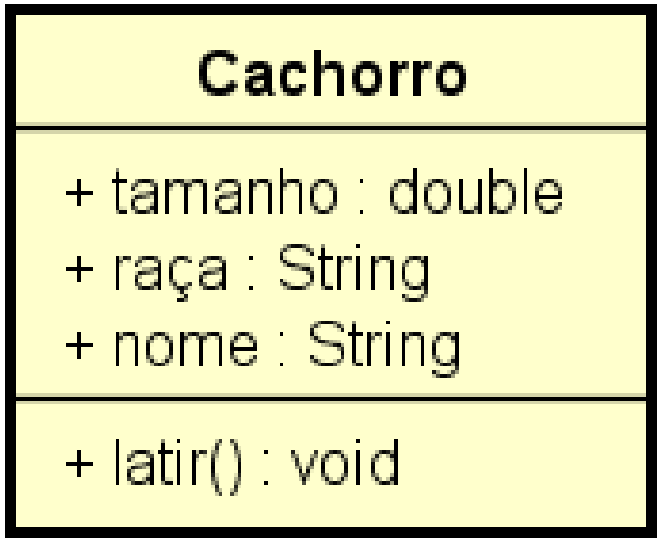


Diagrama de classes UML

Poderia haver um método CORRER e o atributo PESO, que influenciaria na velocidade

Cachorro de Maria		
Tamanho	Raça	Nome
40 cm	Pug	Jesuíno
Latido de volume moderado		

Objeto botão

- Quando projetar uma classe, pense nos objetos que serão criados...

Botão
+ rótulo : String + cor : String
+ configurarCor() : void + configurarRotulo() : void + pressionar() : void

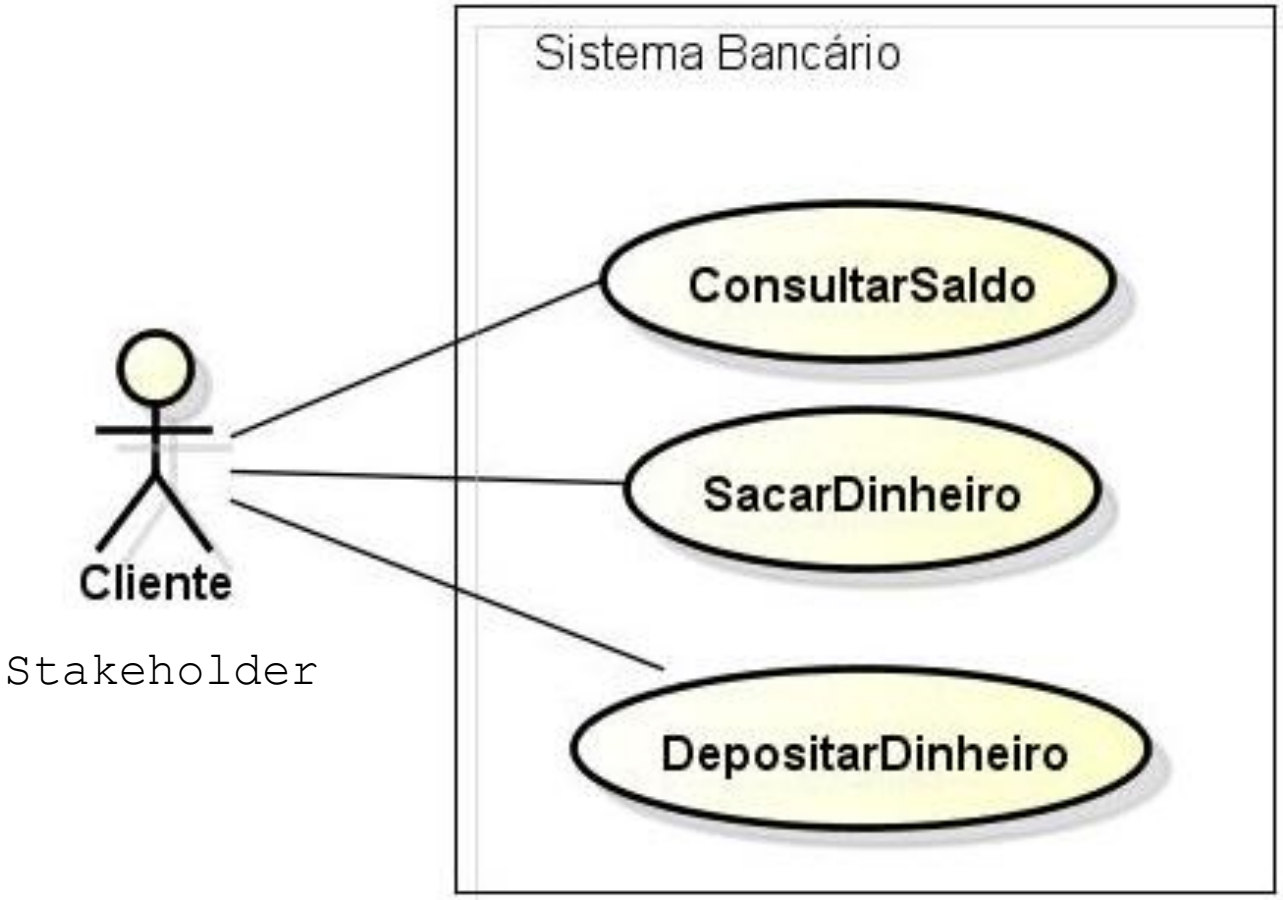
Poderia ter ícone (imagem) e método para configurá-lo

Objeto carrinho de compras

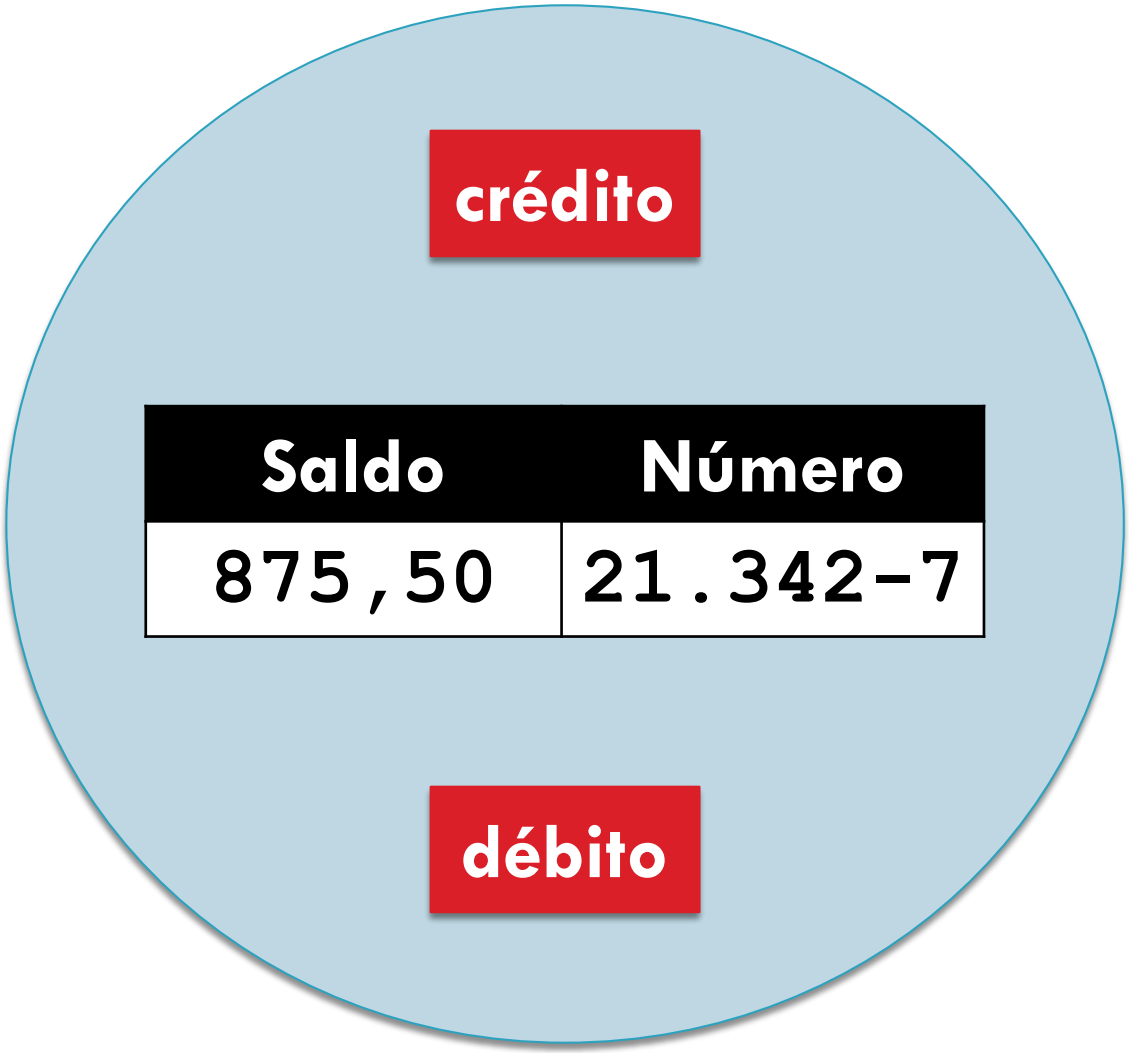
- Pense num carrinho de compras de loja virtual
- Atributos?
- Métodos?

Sistema bancário

- Diagrama de caso de uso da UML



Objeto conta bancária

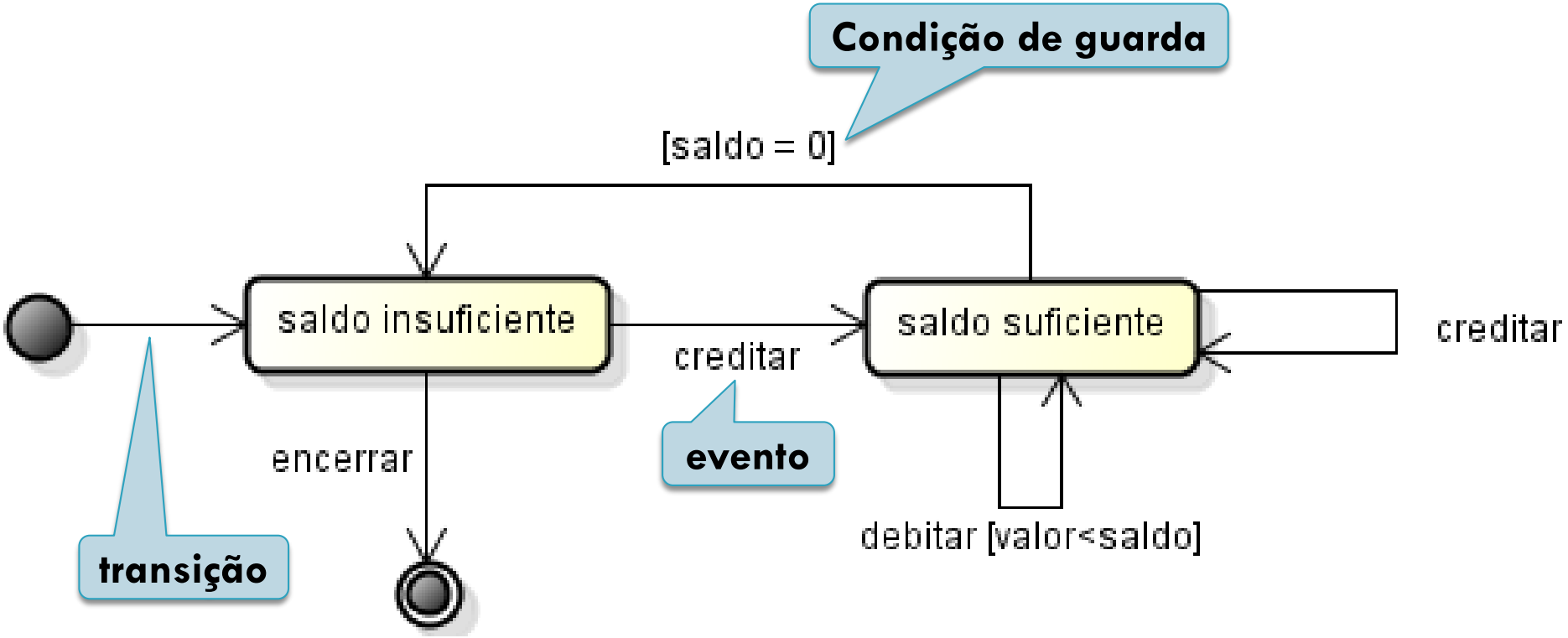


Características de um objeto

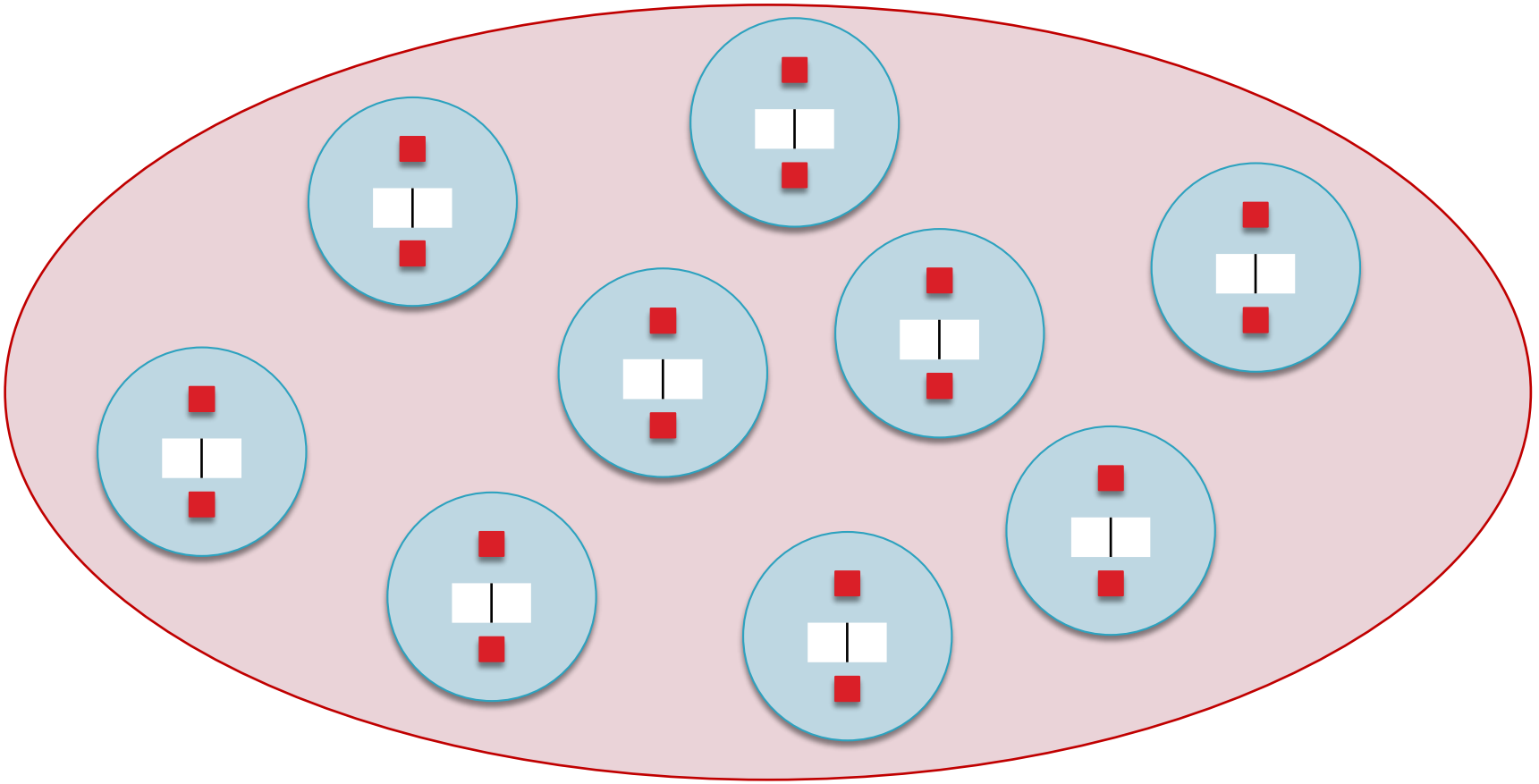
- **Identidade:** O objeto é identificado de forma única
  - Diferentes espaços na memória
- **Estado:** Valor dos atributos do objeto
  - Varia ao longo do tempo
- **Comportamento:** É definido pelos métodos que o objeto pode executar
  - Exemplo: Quando o botão de crédito é pressionado

Estados do objeto conta bancária

- Diagrama de estados da UML



Qual é a diferença entre classe e objeto?

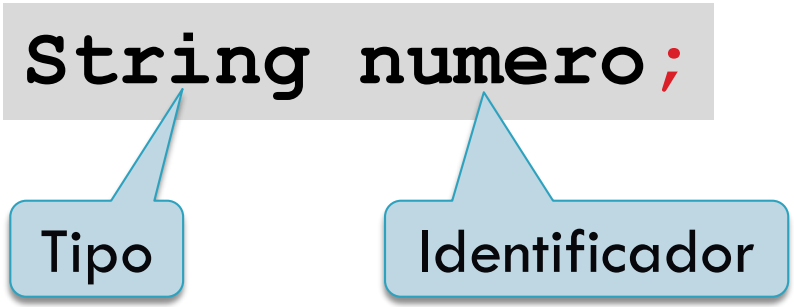


Declaração de classe

```
class Conta {
   //atributos
   //métodos
}
```

- Convenção de nome
  - Substantivo
  - Inicial maiúscula: `Cliente` `ClientePessoaFisica`
  - Evitar abreviações

Declaração de atributos



- Convenção de nome (mesmo de variável)
  - Substantivo
  - Inicial minúscula
  - Junção de palavras captalizadas
  - Nomes curtos e significativos

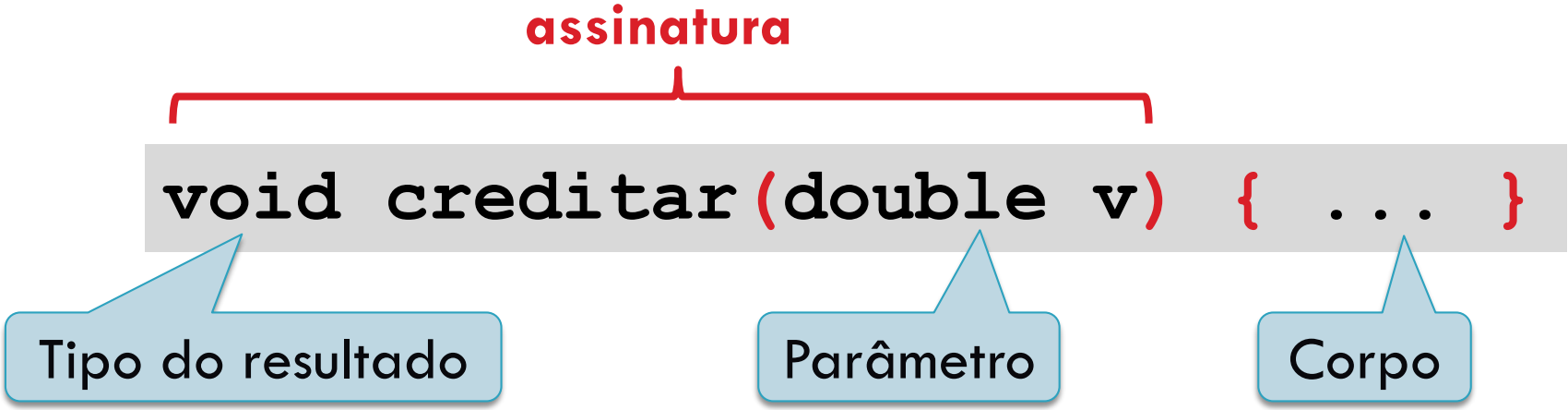
Classe Conta

```
class Conta {

   String numero;
   double saldo;
   String nome, sobrenome;

}
```

Declaração de métodos



- Convenção de nome
  - Inicial minúscula
  - Junção de palavras captalizadas: `consultarSaldo`
  - Verbo

Exemplo de declaração de método

```
class Conta{
    String numero;
    double saldo;

    double consultarSaldo(){
        return saldo;
    }

    void creditar(double v){
        saldo = saldo + v;
    }

}
```

Sem parâmetros

Retorna valor

1 parâmetro

Não retorna valor

Como seria o método debitar?

```
void debitar(double valor){
    saldo = saldo - valor;
}
```

```
void debitar(double valor){
    if(valor <= saldo){
        saldo = saldo - valor;
    }
}
```

Acesso à atributos

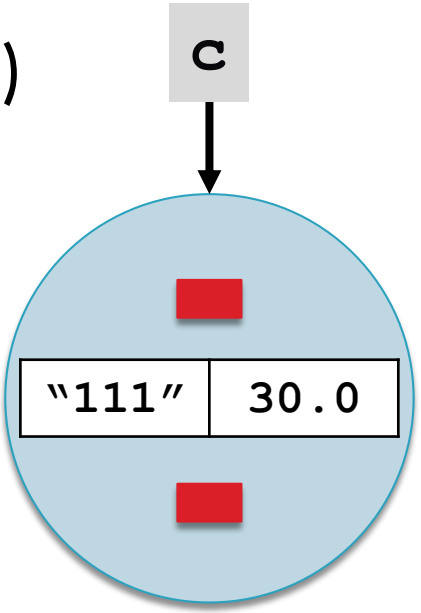
- Um atributo é acessado através de um objeto

- Exemplo (acesso de fora da classe)

```
Conta c = new Conta();

c.saldo = 30;
c.numero = "111";
```

Os atributos estão encapsulados em **c**

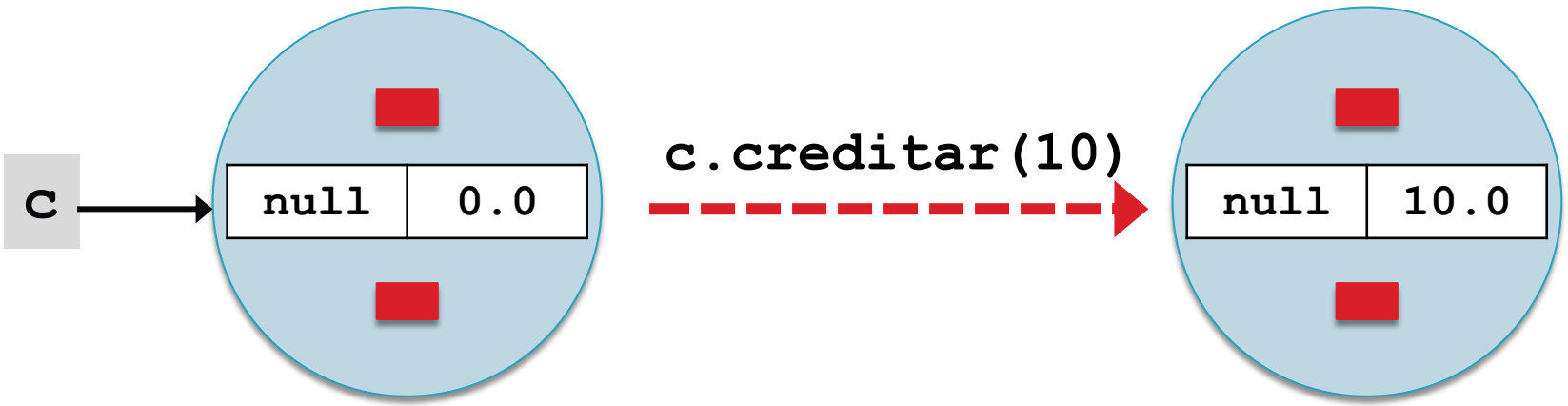


Chamada de métodos

- Um método sempre é executado por um objeto

- Exemplo (acesso de fora da classe)

```
Conta c = new Conta();
c.creditar(10);
```



Conclusão

- Conceitos-chave do paradigma OO
- O que são objetos
- Classes vs. Objetos
- Encapsulamento