Introdução à Programação

Conceitos Básicos de Orientação a Objetos

Java

- A linguagem de programação que adotaremos na disciplina é **Java**

"Java é uma linguagem de programação simples, orientada a objetos, distribuída, interpretada, robusta, segura, independente de arquitetura, portável, de alta performance, concorrente e dinâmica"

Propaganda da SUN Microsystems

Tópicos da Aula

- Antes de aprender a programar em Java, precisamos saber o que é Orientação a Objetos
  - Idéia geral
  - Conceitos Básicos
    - Objeto, Classe, Atributo, Método, Herança
- Depois veremos como podemos utilizar objetos para a definição de outros objetos
  - Definição de API
  - Elementos de uma API
  - Exemplo de API gráfica (miniJava)

O que é Orientação a Objetos?

- É considerar que tudo é um objeto:
  - Os sistemas são objetos
  - Os dados manipulados e comunicados entre os sistemas são objetos
  - Cada parte de um sistema (sub-sistema) é um objeto
  - Os dispositivos eletrônicos são objetos
  - A interface com o usuário é um objeto, composto de vários outros objetos...

Objeto

- Podemos definir **objeto** como uma entidade que possui:
  - **Estado**
    - Características ou propriedades do objeto
  - **Comportamento**
    - Operações que o objeto pode realizar
- O comportamento pode examinar ou alterar o estado de um objeto

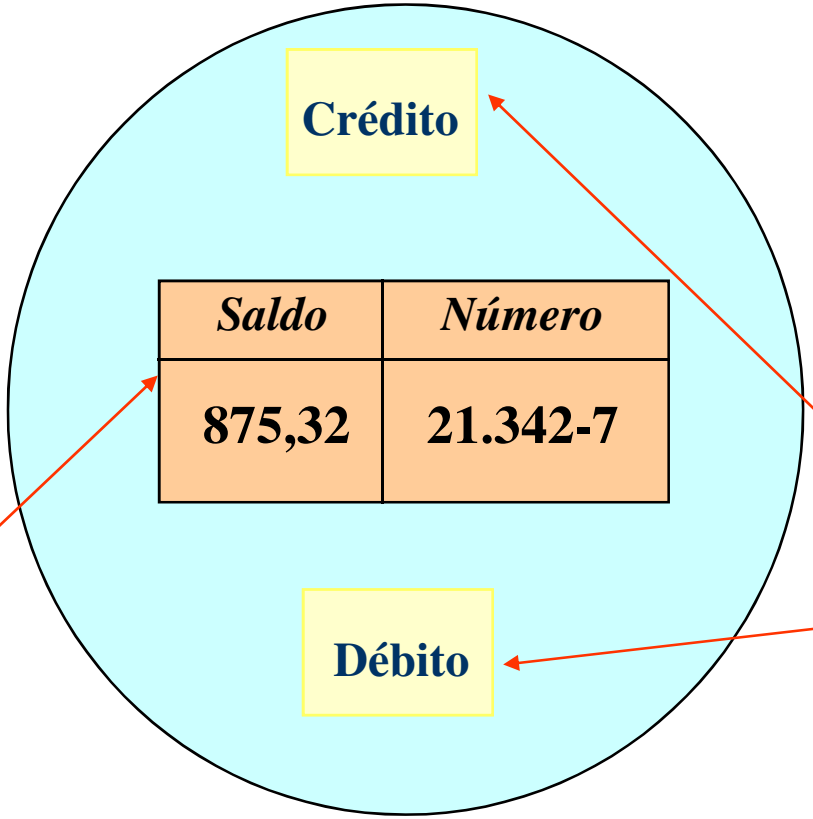
Objeto DVD

As operações que o DVD pode executar

Fonte: http://www.amazon.com

O estado atual do DVD; o que ele está fazendo...

Objeto Conta Bancária



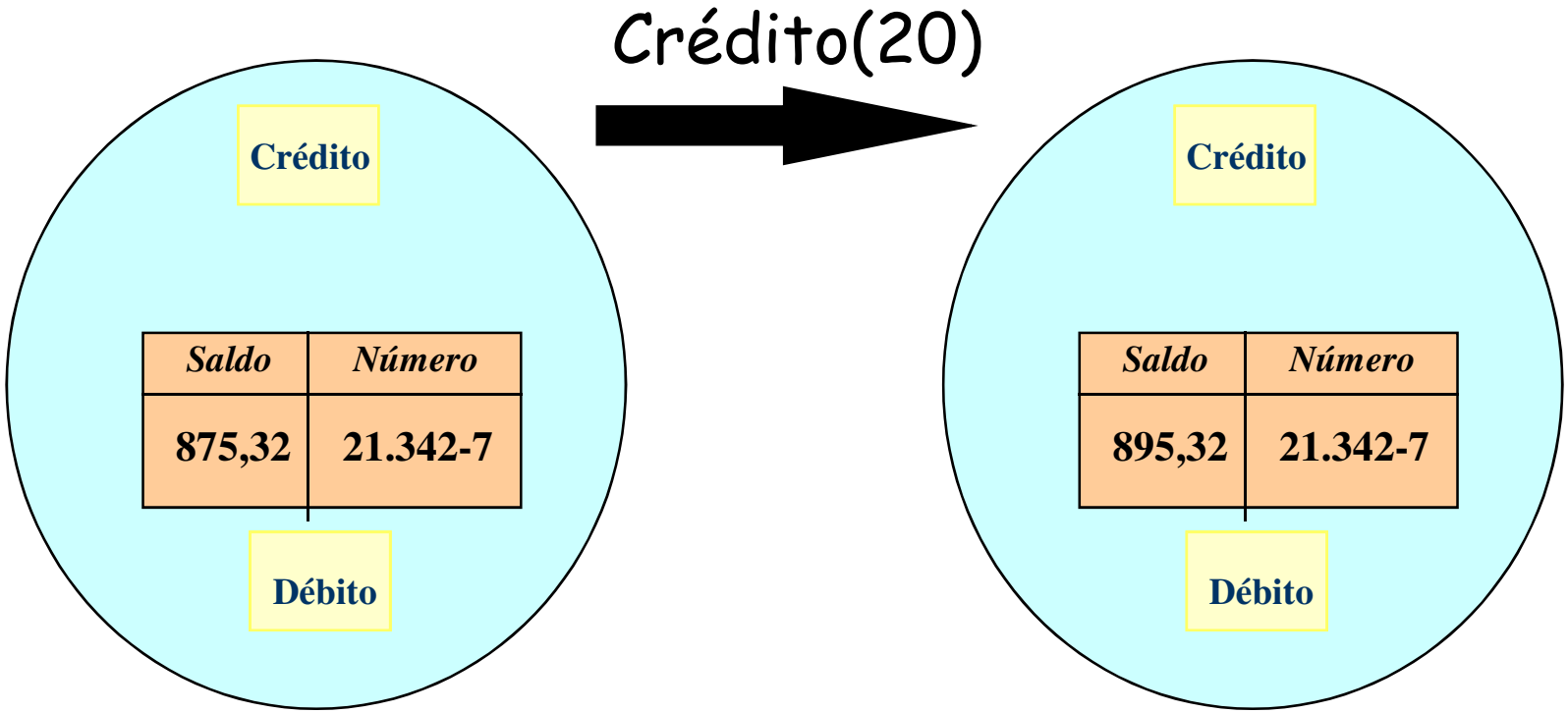
O estado atual da conta

Comportamento

Operações que uma conta pode executar

Estados do Objeto Conta

- Comportamento neste caso mudou o estado do objeto conta

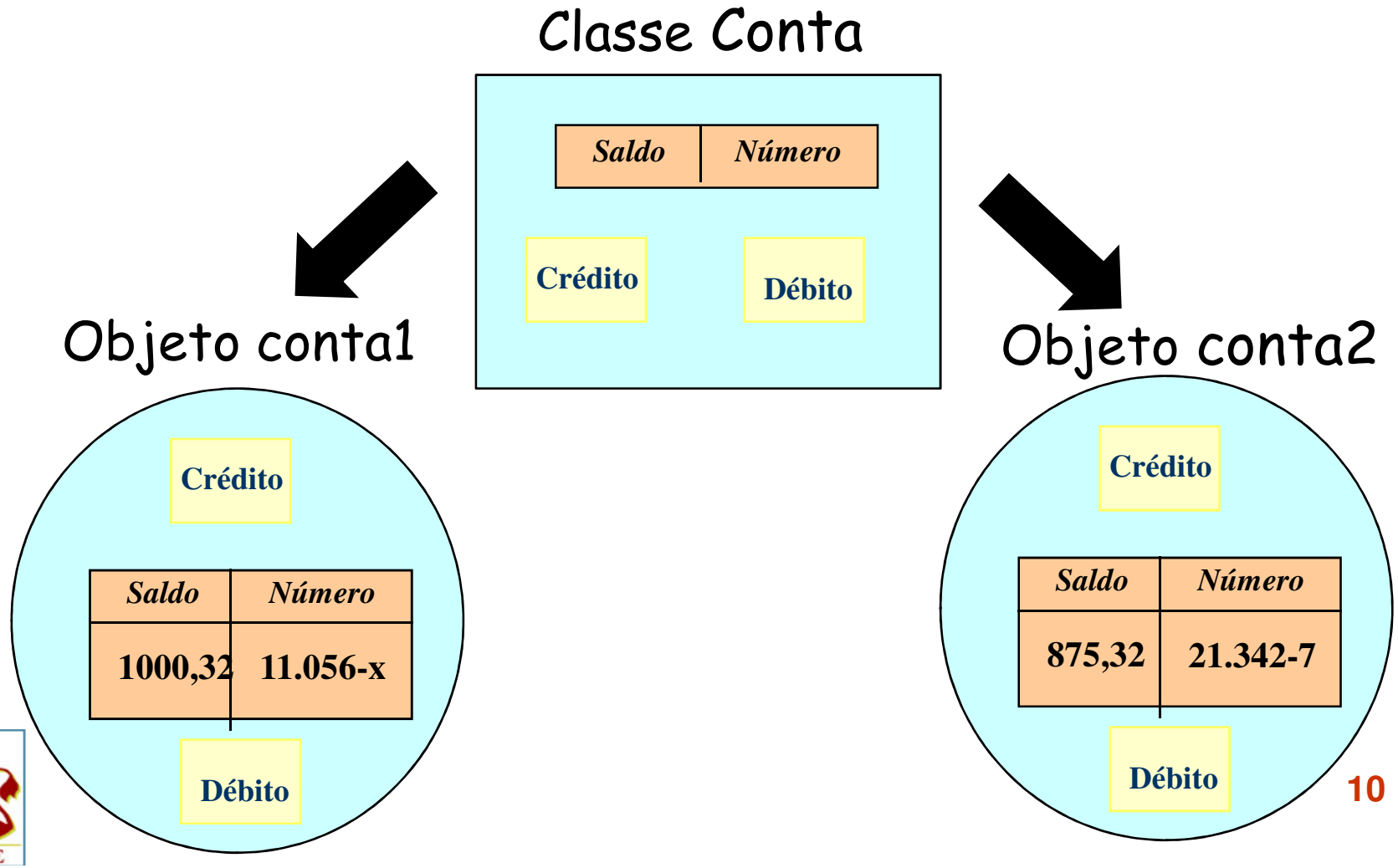


Classe

- **Classe** é um agrupamento de objetos que têm propriedades comuns e podem realizar as mesmas operações
- É uma definição que descreve como objetos pertencentes à classe são estruturados internamente (quais propriedades e operações o objeto possui)
- Classe é um conceito e o objeto é uma instância deste conceito
- Portanto, podemos ter vários objetos pertencentes a mesma classe

Classe x Objeto

- Múltiplos objetos podem ser criados à partir da mesma classe



Classe e Tipo

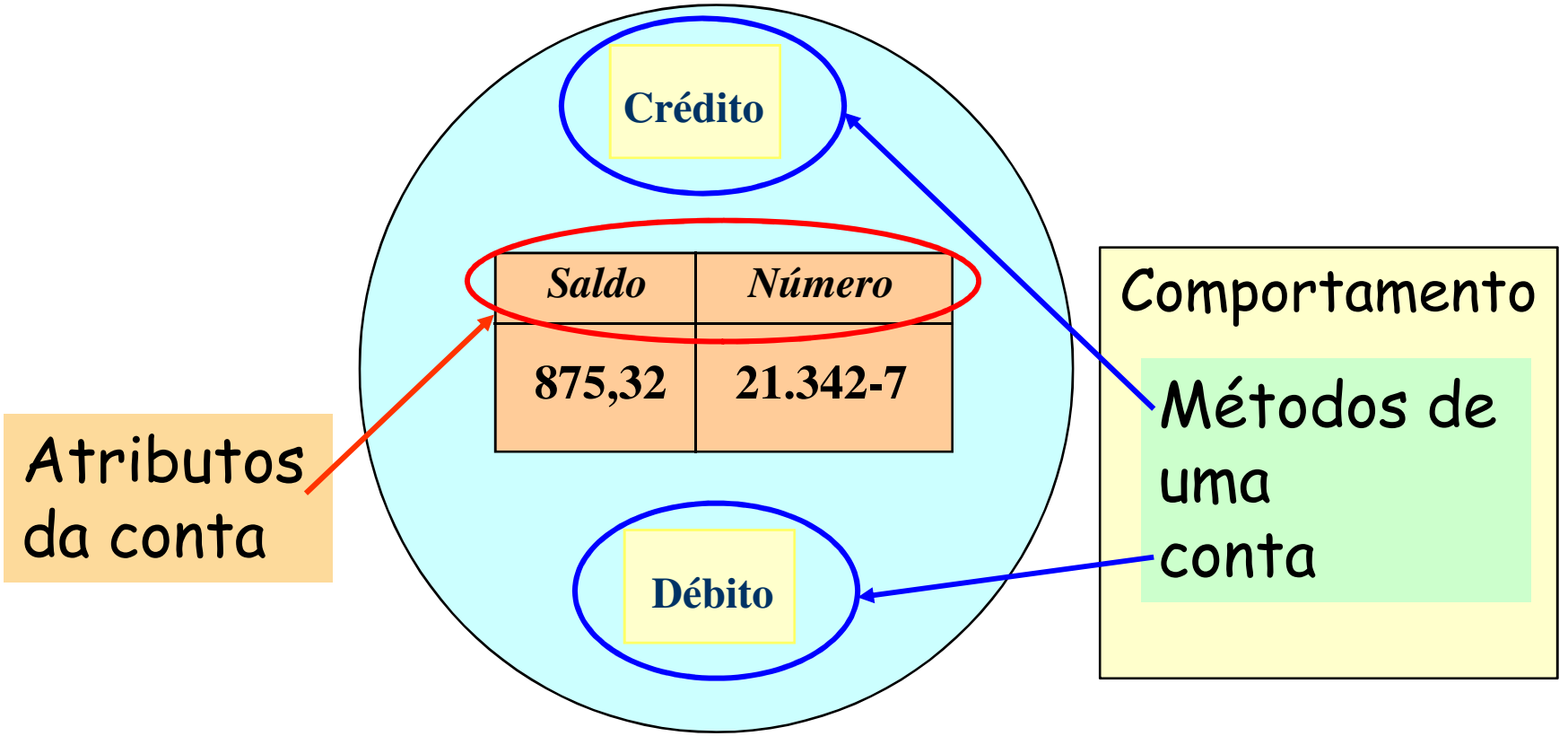
- Tipo é um conjunto de valores relacionados que são capazes de realizar as mesmas operações
  - Ex: inteiro, caractere, Conta, Banco, etc

- Uma classe define um tipo: o tipo cujos elementos são objetos com as mesmas propriedades e comportamento

Atributo e Método (1)

- **Atributo** representa alguma propriedade do objeto
- O conjunto de atributos de um objeto determina o estado do objeto
- Um atributo pode conter um valor simples tal como um número real ou até outro objeto
- **Método** representa uma operação que um objeto pode fazer
- O conjunto de métodos de um objeto determina o comportamento do objeto

Atributo e Método (2)

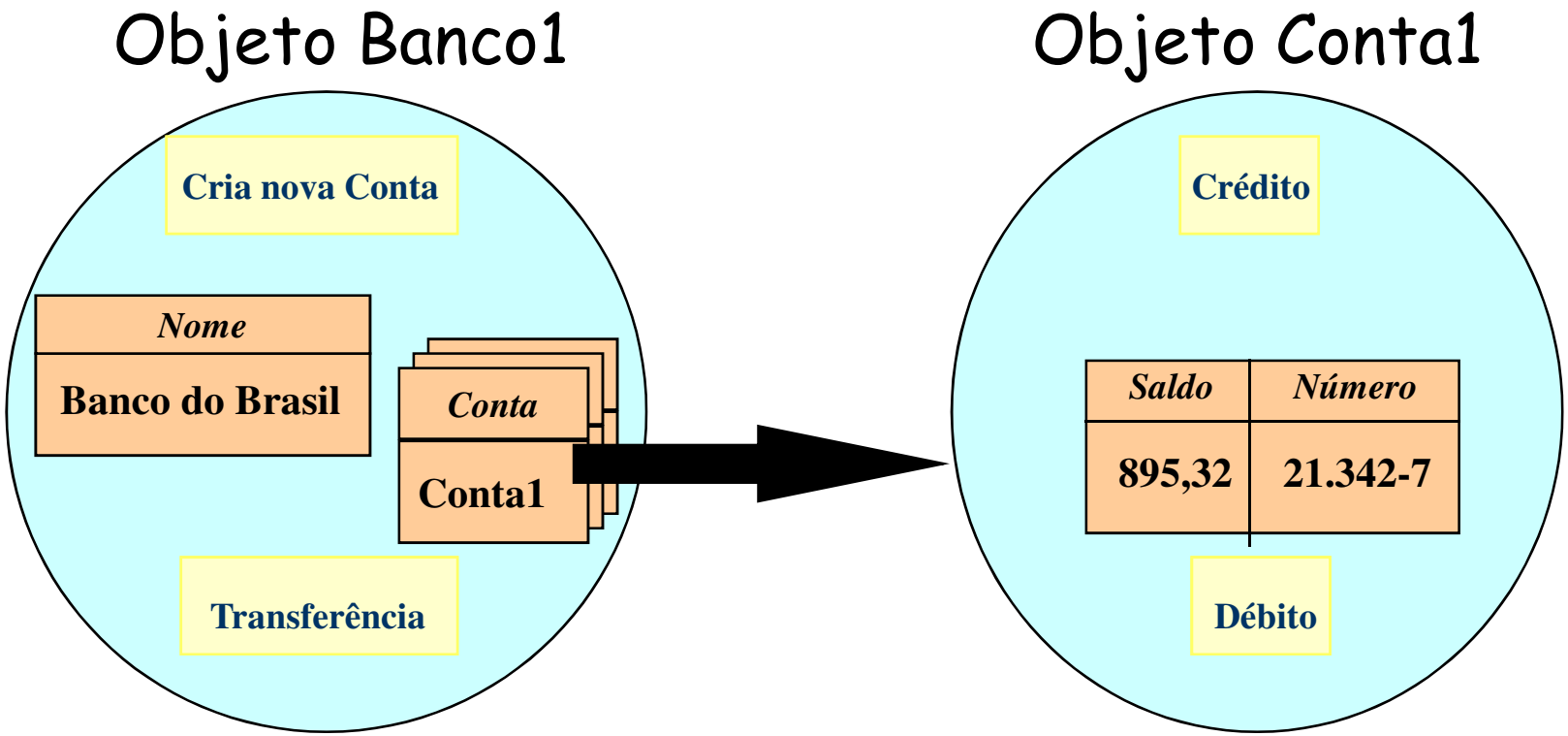


Relações entre Objetos

- Geralmente, um software é construído a partir de vários objetos (classes) que se relacionam
- As relações de **Cliente** e **Herança** são essenciais para a construção de programas
- Classe *B* é **Cliente** de classe *A*, se todo objeto de tipo *B* contiver informação sobre um ou mais objetos do tipo *A*
- Classe *B* é **Herdeira** de classe *A*, se *B* representa uma versão especializada de *A*
  - Métodos e os atributos da classe A se tornam, também, parte da classe B
  - Características e comportamento de A são passadas para B

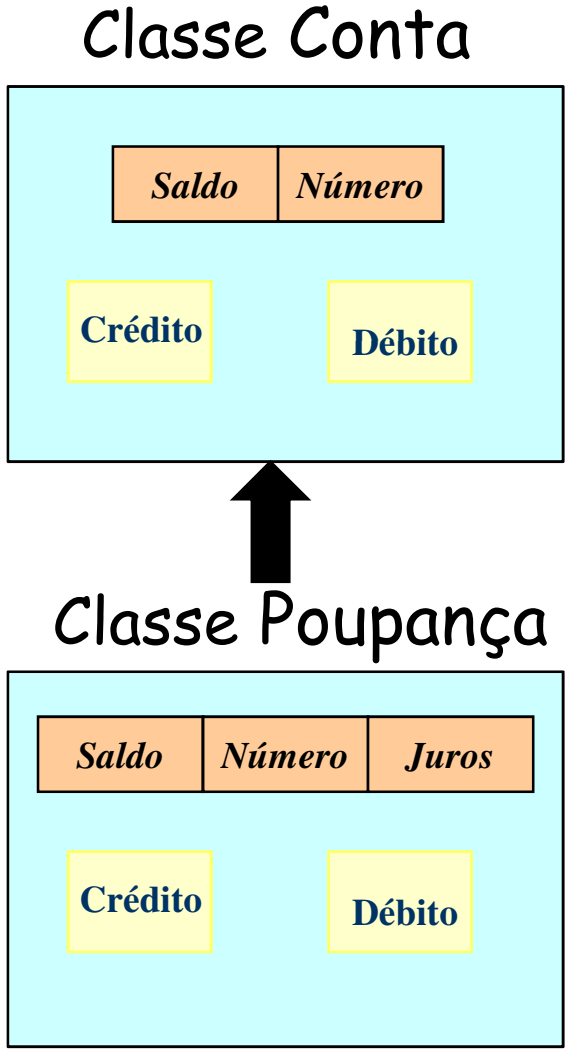
Relação Cliente

- Objeto do tipo Banco têm vários objetos do tipo Conta ➡ Banco é cliente de Conta



Relação Herança

- Classe Poupança é uma versão especializada de Conta
  - ➡ Métodos e atributos são herdados



Utilizando OO para Desenvolver Programas

- Desenvolvimento de uma solução se torna mais fácil quando quebramos esta solução em módulos gerenciáveis
- Quando estamos programando, podemos desenvolver módulos separados, onde cada um é responsável por uma certa parte da solução
  - Programação Modular
- OO facilita a programação modular
  - Módulos são as classes e objetos

Algumas Considerações sobre OO

- Orientação a Objetos ➡ Modularidade
  - Reusabilidade
  - Extensibilidade
- Classe é uma entidade que combina 2 conceitos: **Módulo e Tipo**
  - **Módulo** é um conceito **sintático**, pois a decomposição do sistema em módulos afeta **somente** a forma como o programa é escrito, não a funcionalidade dele
  - **Tipo** é um conceito **semântico**, pois o tipo influencia diretamente a execução do sistema, definindo a forma dos objetos que são criados e manipulados pelo sistema em tempo de execução

OO e Linguagens de Programação

- Linguagens OO têm em objetos e classes seus elementos fundamentais para construção de programas
  - Estruturas da linguagem permitem mapeamento direto dos conceitos de OO
- Porém, é possível fazer um programa OO em uma linguagem não OO
  - É possível, também, fazer um programa que não seja OO em uma linguagem OO

Conceito de Orientação a Objetos é independente da linguagem de programação

API e Programação (1)

- Geralmente na construção de um programa não precisamos implementar todas as partes que o compõem
- Muitas vezes, reutilizamos funcionalidades que já foram implementadas por outros módulos ou programas
  - Pode ser visto, também, como um programa requisitando um serviço a outro programa ou módulo
  - Não precisamos saber como a funcionalidade (serviço) foi implementada e sim como ela funciona
- *Application Programming Interface* (API) define o conjunto de serviços que um determinado módulo ou programa oferece e de que forma estes serviços podem ser utilizados

API e Programação (2)

- Portanto um programa (módulo) acessa os serviços fornecidos por um outro programa (módulo) através da API deste último
  - O programador inclui chamadas ao serviços da API dentro do código do programa
- O que importa é a **interface** da API, ou seja quais são os serviços oferecidos e como podemos utilizá-los
- O conceito de API está presente em qualquer ambiente de programação

Conceito de API é independente de Orientação a Objetos

API em Programação OO

- APIs implementadas em linguagens OO, geralmente, apresentam:
  - Classes
  - Métodos das classes
  - Construtores
    - Tipo especial de método que cria um objeto da classe desejada
    - Uma classe pode ter mais de um construtor
- Apenas algumas informações (referentes ao modo de uso e semântica) dos métodos e construtores são conhecidas pelo programador que deseja usá-las
- Essas informações são denominadas de **assinatura** do método ou construtor

Assinatura de um Método

- A assinatura de um método apresenta :
  - Nome e semântica do método
  - Tipo de valor retornado
    - primitivo – inteiro, caractere, real, etc
    - Objeto – Conta, Banco, DVD, etc
  - Tipo e semântica de cada parâmetro
    - Um **parâmetro** é um valor que deve ser fornecido a um método para que este possa realizar a funcionalidade desejada
  - Os erros que podem ocorrer durante a execução do método (exceções)

Assinatura de um Construtor

- A assinatura de um construtor apresenta:
  - Nome do construtor
    - Linguagens como Java exigem que o construtor tenha o mesmo nome da classe
  - Tipo e semântica de cada parâmetro
    - O objeto criado pode ter alguns atributos inicializados através dos parâmetros
  - Os erros que podem ocorrer durante a criação do objeto (exceções)

Exemplo de uma API Gráfica

- Neste curso, utilizaremos uma API gráfica para fazer programas
  - miniJava
- Esta API foi implementada em Java
- Uma API gráfica contém elementos que permitem criar interfaces gráficas, de modo a facilitar a interação usuário-máquina
  - Janelas, Botões, Menus, Campos de texto, etc
- Uma visão geral da API pode ser acessada aqui

Pacotes de miniJava

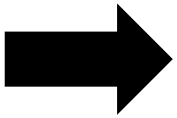
- A API miniJava é organizada em diversos pacotes
- **Pacote** é uma entidade que agrupa classes logicamente relacionadas
  - Funciona como uma espécie de biblioteca
  - Evita conflito de nomes de classes
- Pacotes de miniJava:
  - br.ufpe.cin.miniJava.gui
  - br.ufpe.cin.miniJava.gui.listener
  - br.ufpe.cin.miniJava.io
  - br.ufpe.cin.miniJava.util
  - br.ufpe.cin.miniJava.exception

Utilizando miniJava

Janela

Window(“Minha Janela”)

void include(Botao,50, 90)



Botao

void setText(“Fechar”)

Button( )

Resumindo...

- Orientação a Objetos
  - Objeto, Classe, Atributo, Método
  - Relação entre os conceitos acima
  - Relação entre objetos para construir um programa
- API
  - Conceito de API
  - Elementos de uma API
  - Interface dos elementos de uma API
  - API gráfica - miniJava
  - Uso informal de miniJava