

UNIP

UNIVERSIDADE PAULISTA

LPOO

- origem nos anos 60 na Noruega, com Kristen Nygaard e Ole-Johan Dahl, no Centro Norueguês de Computação
- da linguagem Simula 67, foram introduzidos os conceitos de classe e herança
- Xerox, seqüência de protótipos da linguagem Smalltalk. O líder desse projeto foi ***Alan Curtis Kay***, um dos criadores do termo “programação orientada a objetos”

- 1967: Simula - introduz os primeiros conceitos de OO
- 1972: Smalltalk
- 1980: C++ linguagem híbrida, derivada da linguagem C
- 1983: Ada criada para uso militar nos EUA
- 1984: Eiffel primeiras características formais de OO
- 1986: Object pascal
- 1995: JAVA - Linguagem puramente orientada a objetos
- 1995: Várias linguagens agregando conceitos de OO

- A primeira linguagem a incorporar facilidades para definir classes de objetos genéricos na forma de uma hierarquia de classes e sub-classes
- Foi idealizada em 1966, na Noruega, como uma extensão da linguagem ALGOL 60. Uma classe em Simula é um módulo englobando a definição da estrutura e do comportamento comuns a todas as suas instâncias (objetos).

- Smalltalk foi desenvolvida no Centro de Pesquisas da Xerox durante a década de 70 Incorporou idéias de Simula Criou o princípio de objetos ativos, prontos a "reagir" a "mensagens" que ativam "comportamentos" específicos do objeto C++

- Ser melhor do que C
- Suportar abstração de dados
- Suportar programação orientada a objetos
- C++ foi projetada para dar suporte a abstração de dados e programação orientada a objetos
- C++ não impõe um paradigma

- Ada é uma linguagem de programação criada através de um concurso realizado pelo U.S. Department of Defense (DoD)
- O principal projetista da equipe foi o francês Jean Ichbiah.
- Esse concurso foi feito para por ordem na situação, o DoD em 1974 usava cerca de 450 linguagens ou dialetos de programação.
- A linguagem foi primeiramente padronizada em 1983 pelo ANSI e em 1985 a Organização Internacional de Padronização (ISO).

- Eiffel é uma Linguagem de Programação avançada, puramente orientada a objeto que enfatiza o projeto e construção de software reusável e de alta qualidade.
- Eiffel foi criada por Bertrand Meyer que tinha uma extensa experiência com programação orientada a objeto, particularmente com SIMULA.

- O Object Pascal é uma linguagem orientada a objetos, isto é, todas as informações são tratadas como objetos
- Todos estes objetos pertencem a uma classe, que são categorias de objetos Delphi / Kylix / Lazarus são exemplos de ferramentas que utilizam esta linguagem.

- O Java é ao mesmo tempo um ambiente e uma linguagem de programação desenvolvida pela Sun Microsystems, Inc.
- Foi projetado para resolver os problemas da área de programação cliente/servidor.
- Os aplicativos em Java são compilados em um código de bytes independente de arquitetura. O código de bytes pode então ser executado em qualquer plataforma que suporte um interpretador Java.
- O Java requer somente uma fonte e um binário e, mesmo assim, é capaz de funcionar em diversas plataformas.

- Programação orientada a objetos é uma metodologia de desenvolvimento de software.
- Sua principal vantagem é a proximidade com a forma que os seres humanos visualizam e entendem o mundo ao seu redor.
- Tem como principal objetivo facilitar a modelagem e o desenvolvimento de sistemas, através da interação entre objetos

- Os sistemas, em geral, possuem uma divisão de código um pouco mais lógica e melhor encapsulada do que a empregada nos sistemas não orientados a objetos. Isto torna a manutenção e extensão do código mais fácil e com menos riscos de inserção de *bugs*. Também é mais fácil reaproveitar o código.
- É mais fácil gerenciar o desenvolvimento deste tipo de software quando temos uma equipe grande. Podemos fazer uma especificação UML antes de iniciar o desenvolvimento do software em si.

- O aprendizado do paradigma de programação orientada a objetos é bem mais complicado no início do que os sistemas procedurais.
- Para começar a programar é necessário ter estabelecido uma série de conceitos bastante complexos. Já na programação procedural tradicional, basta decorar meia dúzia de comandos e você já consegue fazer um programa simples.
- Dificilmente uma linguagem orientada a objetos conseguirá ter um desempenho em tempo de execução superior a linguagens não orientadas a objetos.

Objetos

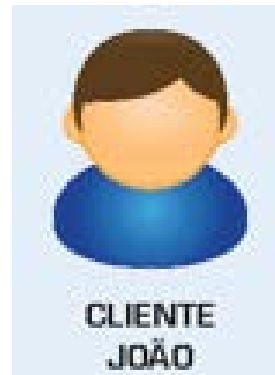
- Objetos podem ser considerados uma imitação do comportamento intrínseco de entidades reais.
- Tal como em sistemas reais, em um programa OO não é viável abrir um objeto e olhar em seu interior, tampouco alterar seu estado.
- Neste paradigma, a única forma de fazer evoluir um programa é permitir que objetos compartilhem dados entre si através de trocas explícitas de mensagens

A esposa de um jovem casal se encontra no segundo mês de gravidez. No meio da noite, a grávida acorda com uma azia terrível. Como é natural, a mulher pede que seu marido providencie um antiácido para ela.

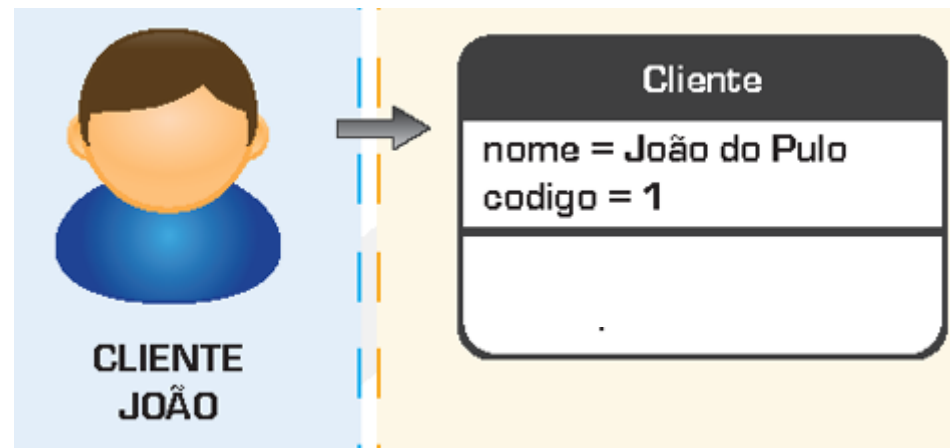
- - O objeto marido recebe uma mensagem do objeto esposa;
- - O objeto marido responde à mensagem da esposa mediante uma ação: *buscar antiácido*;
- - A esposa não tem que dizer ao marido onde ele deve procurar, é responsabilidade dele procurar pelo antiácido;
- - Ao objeto esposa basta ter emitido uma *mensagem* ao objeto marido.

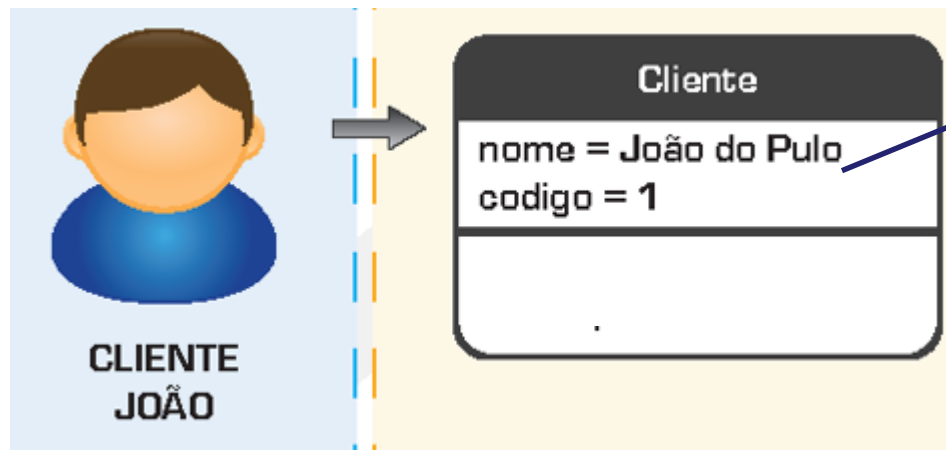
- Nas aplicações OO, as entidades são representadas por objetos.
 - Uma aplicação orientada a objetos é composta por objetos.
 - Em geral, um objeto representa uma entidade do domínio.

- no domínio de um determinado banco exista um cliente chamado João



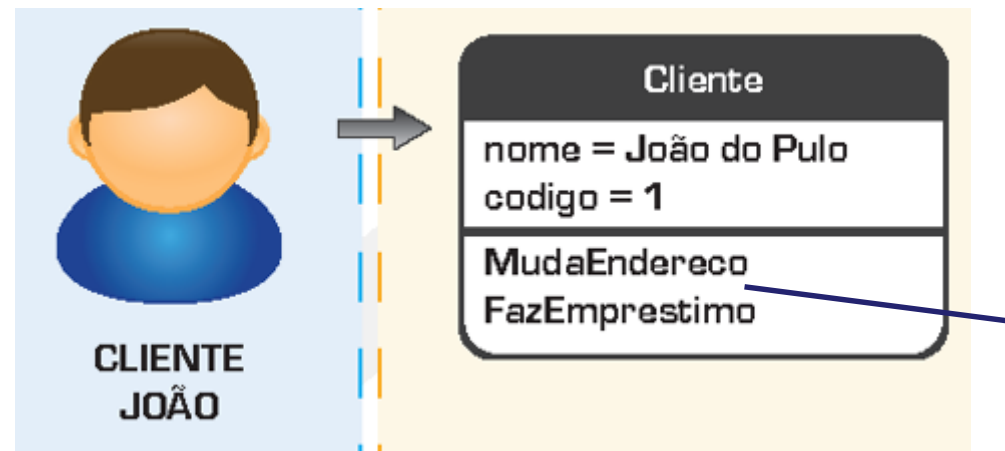
- Dentro de uma aplicação orientada a objetos correspondente a esse domínio, deve existir um objeto para representar esse cliente.
- Algumas informações do cliente João como nome, data de nascimento e sexo são importantes para o banco. O objeto que representa esse cliente deve possuir essas informações. Esses dados são armazenados nos atributos do objeto que representa o João.





Atributos

- O próprio objeto deve realizar operações de consulta ou alteração dos valores de seus atributos. Essas operações são definidas nos métodos do objeto



Métodos

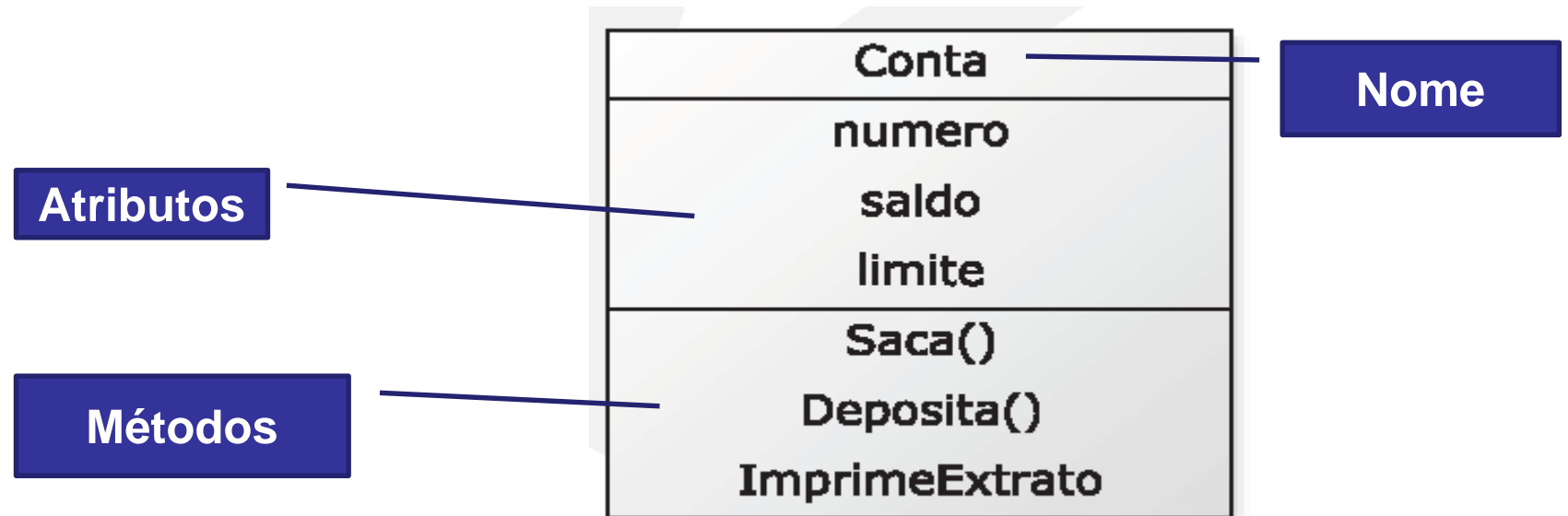
- Um objeto é composto por atributos e métodos.
 - Atributos são as características do objeto.
 - Métodos são as ações que um objeto pode executar.

Classes

- A partir de uma classe, podemos construir objetos na memória do computador que executa a nossa aplicação
- Antes de um objeto ser criado, devemos definir quais serão os seus atributos e métodos
- Uma classe funciona como uma “receita” para criar objetos. Inclusive, vários objetos podem ser criados a partir de uma única classe

- Planta
 - várias casas ou prédios podem ser construídos a partir de uma única planta
- Receita
 - vários bolos podem ser preparados a partir de uma única receita
- Projeto
 - vários carros podem ser construídos a partir de um único projeto

- Podemos representar uma classe através de diagramas UML
- de uma classe é composto pelo nome da mesma e pelos atributos e métodos
- Todos os objetos criados a partir da classe Conta terão os atributos e métodos mostrados no diagrama UML



- Uma classe em Java

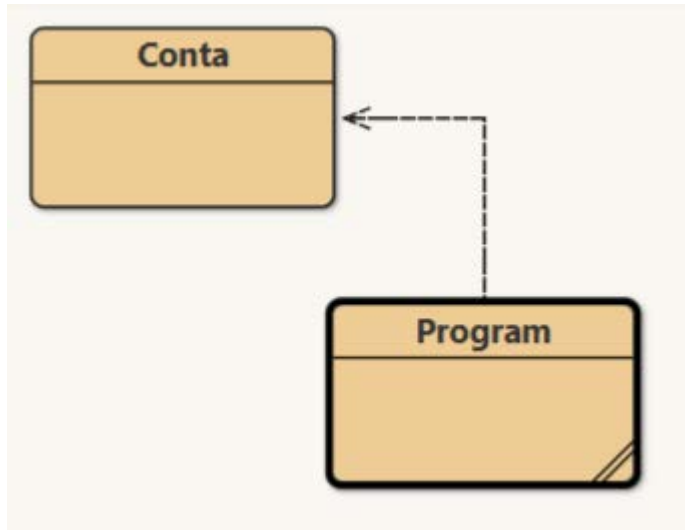
```
public class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}
```

- A classe Conta é declarada utilizando a palavra reservada class.
- No corpo dessa classe, são declaradas três variáveis que são os atributos que os objetos possuirão.
- Os tipos dos atributos são definidos no código.
- Os atributos saldo e limite são do tipo double, e o atributo numero é do tipo int.
- O modificador public é adicionado em cada atributo para que eles possam ser acessados a partir de qualquer ponto do programa

```
public class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}
```

Criação de um objeto a partir de uma classe

- O comando para criar objetos é o new.



```
public class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}
```

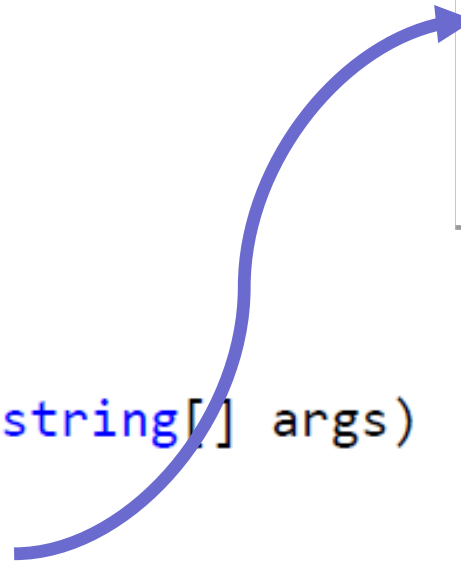
```
public class Program
{
    public static void main(String[] args)
    {
        new Conta();
    }
}
```

Criação de um objeto a partir de uma classe

- O comando para criar objetos é o new.

```
class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}
```

```
class Program
{
    static void Main(string[] args)
    {
        new Conta();
    }
}
```



```
objeto
{
    double saldo;
    double limite;
    int conta;
}
```

Criação de um objeto a partir de uma classe

- Para criarmos três objetos:

```
public class Program
{
    public static void main(String[] args)
    {
        new Conta();
        new Conta();
        new Conta();
    }
}
```

```
class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}
```

```
class Program
{
    static void Main(string[] args)
    {
        new Conta();
        new Conta();
        new Conta();
    }
}
```

objeto

```
{
    double saldo;
    double limite;
    int conta;
}
```

objeto

```
{
    double saldo;
    double limite;
    int conta;
}
```

objeto

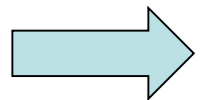
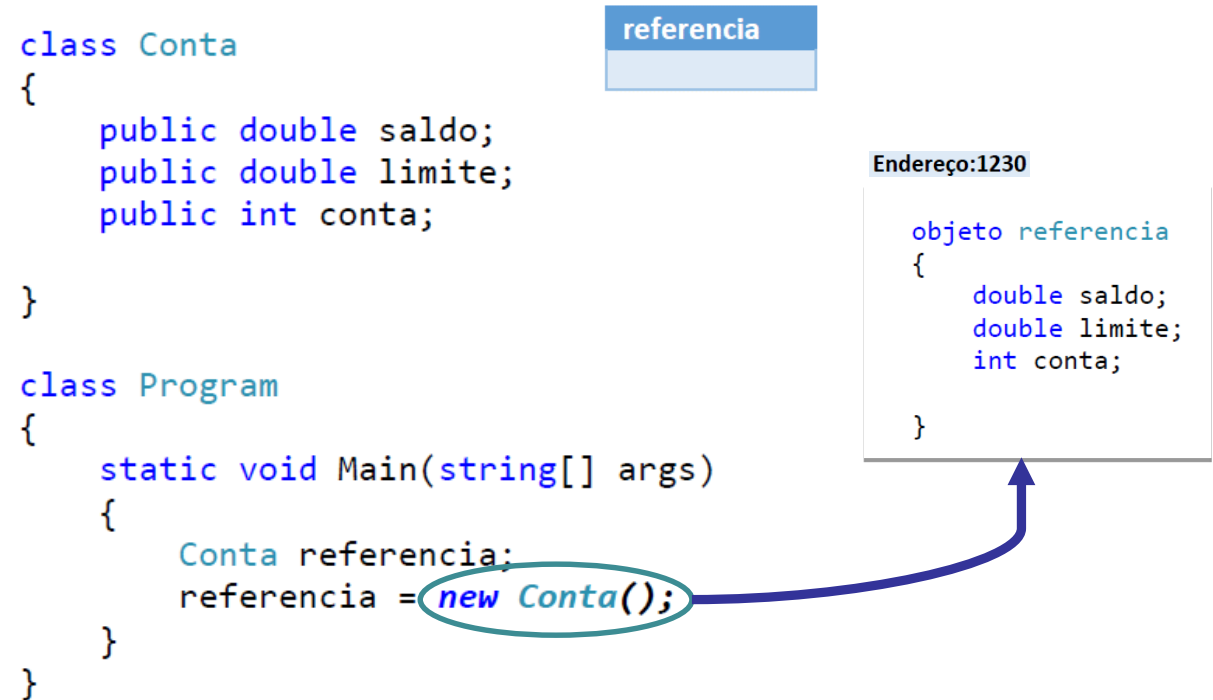
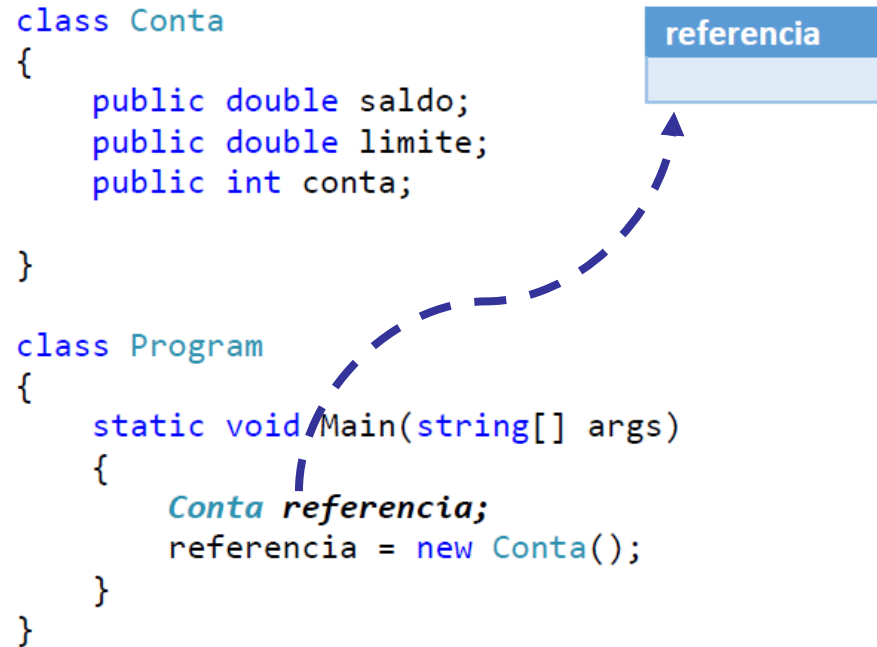
```
{
    double saldo;
    double limite;
    int conta;
}
```

- Todo objeto possui uma referência.
- A referência de um objeto é a única maneira de acessar os seus atributos e métodos.
- Para guardar as referências devolvidas pelo comando new, devemos utilizar variáveis

```
Conta referencia = new Conta();
```

ou

```
Conta referencia;  
referencia=new Conta();
```

```
class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}

class Program
{
    static void Main(string[] args)
    {
        Conta referencia;
        referencia = new Conta();
    }
}
```

referencia

Endereço:1230

Endereço:1230

objeto referencia

{

double saldo;

double limite;

int conta;

}

- Podemos alterar ou acessar os valores guardados nos atributos de um objeto se tivermos a referência a esse objeto.
- Os atributos são acessados pelo nome, utilizando o operador ".".

```
class Conta
{
    public double saldo;
    public double limite;
    public int conta;
}

objeto Referencia
{
    public double saldo; 1000
    public double limite; 300
    public int conta; 123
}

class Program
{
    static void Main(string[] args)
    {
        Conta referencia = new Conta();
        referencia.saldo = 1000.0;
        referencia.limite = 300;
        referencia.conta = 123;
    }
}
```

ATÉ A PRÓXIMA!