



Estrutura geral de uma classe

DC – UFRPE

Programação II

Prof. Gustavo Callou

gustavo.callou@ufrpe.br

Até agora



- ▶ Definimos classes sem atributos
 - ▶ Apenas método main
- ▶ A maior utilidade das classes é definir seus atributos e o comportamento de objetos

Roteiro

- ▶ Classe
- ▶ Objeto
- ▶ Atributos
- ▶ Classe Conta



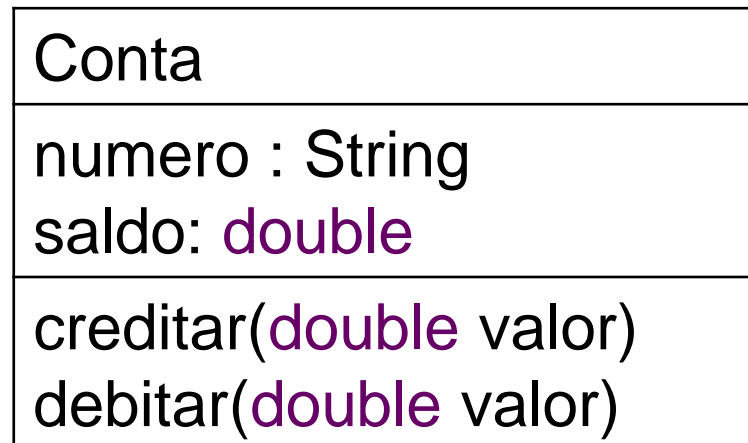
Forma geral para uma classe pública

- ▶ Nome da classe igual ao nome do arquivo .java

```
// lista de imports
public class NomeDaClasse {
    // declaração de atributos
    // declaração de métodos
}
```

Classe

- ▶ Tipo para objetos
- ▶ Define variáveis (atributos) e operações (métodos) dos objetos
- ▶ Exemplo: Classe Conta na representação gráfica (UML)



Objeto

- ▶ Instância (exemplo) de uma classe
- ▶ Possui valor para cada atributo (variáveis de instância)
- ▶ Exemplo: Objetos do Tipo Conta na representação gráfica

c1 : Conta
numero = "1246-8"
saldo = 400.50
creditar(...) debitar(...)

Objeto c1

c2 : Conta
numero = "9654-3"
saldo = 780.50
creditar(...) debitar(...)

Objeto c2

Estado de um objeto

- ▶ Valor variáveis (ou atributos) de um objeto em dado instante

c1 : Conta
numero = "1246-8"
saldo = 400.50
...

Estado antes

c1.debitar(100)



c1 : Conta
numero = "1246-8"
saldo = 300.50
...

Estado depois

Atributos da classe



- ▶ Definem as variáveis dos objetos (instâncias)
- ▶ Possuem valor padrão caso nenhum seja definido
- ▶ Possuem um tipo de acesso definido:
 - ▶ Public
 - ▶ Private
 - ▶ Protected
 - ▶ Default (quando nenhum é definido)

Classe Conta



```
public class Conta {  
    public String numero;  
    public double saldo;  
    // métodos  
}
```

Definindo atributos

tipoAcesso tipo nome = valorInicial;

public ou private
(mais comuns)

Opcional

Classe Conta

- ▶ Criando objeto do tipo conta e mudando seus atributos

```
Conta c1 = new Conta();  
c1.numero = "1";  
c1.saldo = 100;
```

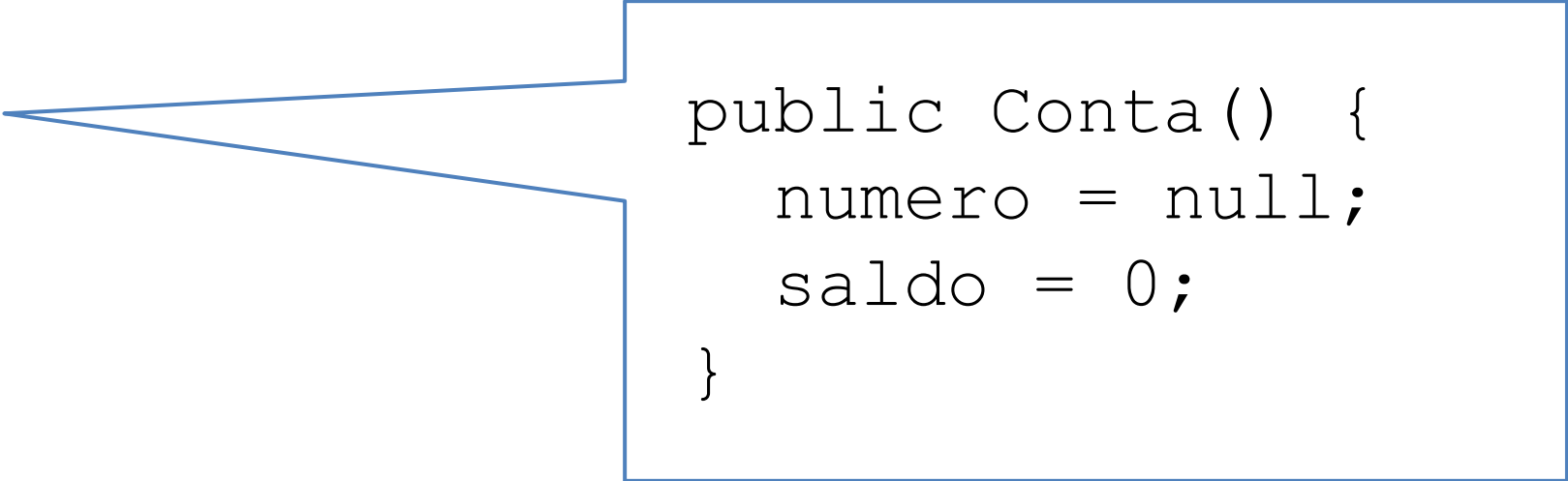
Construtor



- ▶ Método usado para criar instâncias da classe
 - ▶ Tem o mesmo nome da classe
 - ▶ Usado após a palavra **new**
- ▶ O construtor que não recebe parâmetros é chamado de construtor padrão
 - ▶ Se nenhum é definido o compilador cria um padrão
- ▶ Uma classe pode ter mais de um construtor

Construtor padrão para a Classe Conta

```
public class Conta {  
    public String numero;  
    public double saldo;  
    // métodos  
}
```



```
public Conta() {  
    numero = null;  
    saldo = 0;  
}
```

Valor padrão para atributos

- ▶ De tipo primitivo:
 - ▶ numérico é 0
 - ▶ booleano é igual a false
- ▶ De tipo de referência é null
- ▶ Exemplo: ao criar uma Conta com o construtor padrão número = null e saldo = 0.0;

Definindo construtor para Conta

```
public class Conta {  
    public String numero;  
    public double saldo;  
    public Conta(String num, double s) {  
        numero = num;  
        saldo = s;  
    }  
    // outros métodos  
}
```

Definindo construtor para Conta

- ▶ Passando parâmetros para o construtor definido para Conta

```
Conta c1 = new Conta("1246-8", 400.5);
```

```
Conta c2 = new Conta("9654-3", 780.5);
```


Construtor da Classe Conta

► Usando this no construtor de Conta

```
public class Conta {  
    // atributos  
    public Conta(String numero, double saldo) {  
        this.numero = numero;  
        this.saldo = saldo;  
    }  
}
```

this



- ▶ Referência que aponta para o próprio objeto
- ▶ Útil para remover ambiguidades
 - ▶ Ex: definir se uma variável é parâmetro do método ou atributo da classe
- ▶ Facilita a leitura do código

Exercício:



- ▶ Que outros construtores poderiam ser úteis para a Classe Conta?
- ▶ Qual o construtor de uma Classe Endereco que possui atributos do tipo String para rua, cidade e estado?