

Linguagem de Programação Orientada à Objetos

Orientação à objetos

Objeto

É a representação de um conceito por meio de seu estado e comportamento

→ Estado: conjunto de dados que armazenam informações do conceito.

→ Comportamento: conjunto de operações que são realizadas pelo conceito.

Exemplos:

❶ Conta bancária

Dados: número, saldo

Operações: creditar, debitar

❷ Aluno do IFPE (cadastrado no Q-Acadêmico)

Dados: nome, cpf, endereço

Operações: corrigir nome, atualizar endereço

Como criar? Definindo classes!

Uma classe é um modelo para se criar objetos. Dizemos que um objeto é uma instância de uma classe. Então para criar um objeto basta instanciar uma classe.

Classes são como fôrmas de objetos. Uma vez definida uma classe pode-se instanciar N objetos baseado naquela classe.

Classe

Sintaxe:

```
public class ContaBancaria {  
}
```

Em Java, os nomes das classes devem seguir o padrão camel-case: Cada palavra deve começar com maiúscula unidas SEM espaço ou underline, como o exemplo acima.

Uma classe é definida em um arquivo java que deve ter exatamente o mesmo nome da classe.

Atributos e métodos

Nas classes declaram-se os atributos e métodos.

O conjunto de atributos correspondem ao estado do objeto que será criado.

→ Cada atributo tem um tipo: int, double, char, boolean, String, etc.

O conjunto de métodos corresponde ao comportamento do objeto que será criado.

→ Métodos realizam operações que modificam os valores dos atributos.

→ Métodos são “funções” que pertencem a um objeto.

Métodos

Sintaxe:

```
retorno      nome      parâmetros  
public double metodo(int par1, double par2) {  
    // instruções  
    return 2.5;  
}             corpo
```

Declarando método que não retornam nada:

Ao invés de ser public double, como mostrado acima, é definido como public void.

Declarando método que não recebe parâmetros:

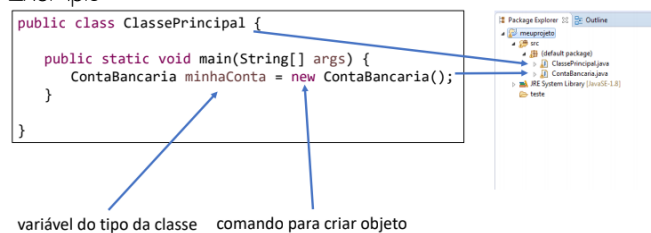
Ao invés de definir os parâmetros, como mostrado acima, apenas deixe em branco os parênteses.

Exemplo de uma classe com os métodos e atributos definidos:

```
public class ContaBancaria {  
    String numero;  
    double saldo;  
    public void creditar(double valor) {  
        saldo = saldo + valor;  
    }  
    public void debitar(double valor) {  
        saldo = saldo - valor;  
    }  
}  
Atributos  
Métodos
```

Para criar um objeto usamos o comando new e armazenamos o objeto em uma variável do tipo da classe.

Exemplo:



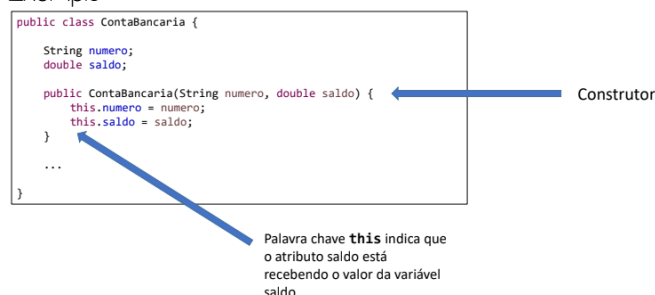
Construtores

As vezes precisamos definir o valor inicial dos atributos do objeto. Fazemos isso através de um método especial chamado construtor.

- Deve ter exatamente o mesmo nome da classe.
- É o único tipo de método que não tem retorno.

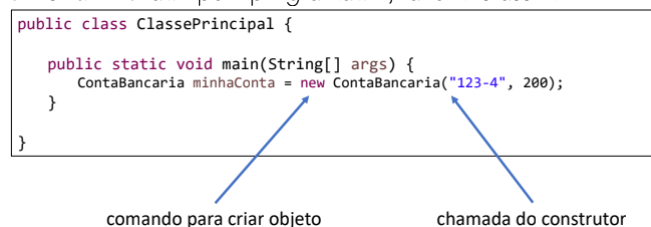
O construtor recebe parâmetros que serão usados para definir o valor dos atributos do objeto. Se o programador não criar um construtor o Java define implicitamente um construtor padrão sem parâmetros que não define os valores iniciais dos atributos.

Exemplo:



ATENÇÃO: Se você não colocar o this o compilador não sabe quem é quem.

Na verdade usamos o construtor padrão. Para usar o construtor criado pelo programador, fazemos assim:



Sabemos que é possível criar vários objetos a partir de uma classe. Podemos fazer assim:

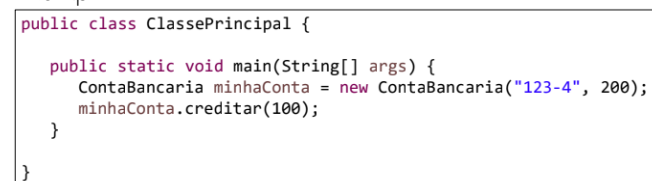


Os objetos servem para que possamos realizar ações com eles. Lembre que eles possuem comportamento?

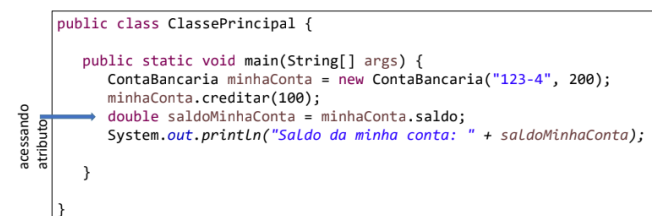
Ativamos o comportamento dos objetos chamando os métodos para realizarem ações em cima dos objetos.

Sintaxe de chamada de método: `minhaConta.creditar(100);`

Exemplo:



Sabemos que o saldo mudou de fato quando acessamos o atributo saldo. Em seguida, podemos armazenar o saldo em uma variável E, por fim, podemos imprimir o valor da variável com no exemplo abaixo:



E o que apareceria na tela?

Saldo da minha conta: 300.0