

Classes e Objetos

Emerson C. Lima

Programação Orientada à Objetos e Algoritmos em C++

Objetivos dessa lição

- Saber declarar classes em Java
- Saber inicializar e utilizar objetos em Java
- Saber utilizar herança em Java
- Saber e utilizar Interfaces em Java


Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos  
    // construtores  
    // métodos  
  
}
```


Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos   
    // construtores  
    // métodos  
}
```

Classes


- Declarando classes em Java

```
class MinhaClasse {  
    // atributos   
    // construtores  
    // métodos  
}
```

Atributos são variáveis que guardam o estado do objeto

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos   
    // construtores  
    // métodos  
  
}
```


Atributos são variáveis que guardam o estado do objeto

Exemplo:

```
class Ponto {  
    int x;  
    int y;  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos   
    // construtores  
    // métodos  
  
}
```


Atributos são variáveis que guardam o estado do objeto

Exemplo:

```
class Data {  
    int dia;  
    int mes;  
    int ano;  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos   
    // construtores  
    // métodos  
  
}
```


Atributos são variáveis que guardam o estado do objeto

Exemplo:

```
class Cliente {  
    String nome;  
    Data aniversario;  
}
```


Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos   
    // construtores  
    // métodos  
  
}
```


Atributos são variáveis que guardam o estado do objeto

Exemplo:

```
class Aluno {  
    String matricula;  
    String nome;  
    char sexo;  
    boolean aprovado;  
    double nota;  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos   
    // construtores  
    // métodos  
  
}
```

Atributos são variáveis que guardam o estado do objeto

Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos  
    // construtores ←  
    // métodos  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos  
    // construtores ←  
    // métodos  
}
```

Construtores são métodos especiais utilizados para inicializar o estado do objeto

Exemplo:

```
class Ponto {  
    Ponto() {  
        x = 0;  
        y = 0;  
    }  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos  
    // construtores  
    // métodos  
}
```

Construtores são métodos especiais utilizados para inicializar o estado do objeto

Exemplo:

```
class Ponto {  
    Ponto(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {
```

```
// atributos
```

Construtores são métodos

Exemplo:

```
class Ponto {
```

```
    Ponto(int x, int y) {  
        x = x;  
        y = y;  
    }
```

```
    Ponto() { // construtor padrão  
        x = 0;  
        y = 0;  
    }
```

```
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {
```

```
// atributos
```

Construtores são métodos

Exemplo:

```
class Ponto {
```

```
    Ponto(int x, int y) {  
        x = x;  
        y = y;  
    }
```

```
    Ponto() { // construtor padrão  
        this(0, 0);  
    }
```

```
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos  
    // construtores ←  
    // métodos  
}
```

Construtores são métodos especiais utilizados para inicializar o estado do objeto

Classes

- Declarando classes em Java

```
class MinhaClasse {  
    // atributos  
    // construtores  
    // métodos ←  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos  
    // construtores  
    // métodos ←  
  
}
```

Métodos são funções que atuam sempre sobre um objeto específico

Exemplo:

```
class Ponto {  
    double distancia(Ponto outro) {  
        double dx = outro.x - this.x;  
        double dy = outro.y - this.y;  
        return Math.sqrt(dx*dx + dy*dy);  
    }  
}
```

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos  
    // construtores  
    // métodos ←  
  
}
```

Métodos são funções que atuam sempre sobre um objeto específico

Classes

- Declarando classes em Java

```
class MinhaClasse {  
  
    // atributos  
    // construtores  
    // métodos  
  
}
```

Objetos

- Objetos são instâncias de uma classe de objetos na memória

Objetos

- Objetos são instâncias de uma classe de objetos na memória

Bicicleta bike;

String nome;

Aluno maria;

Promocao promocao;

Objetos

- Objetos são instâncias de uma classe de objetos na memória

Bicicleta bike;

String nome;

Aluno maria;

Promocao promocao;

Declaração de objetos

Objetos

- Objetos são instâncias de uma classe de objetos na memória

```
Bicicleta bike = new Bicicleta();
```

```
String nome = "Foo";
```

```
Aluno maria = new Aluno("Maria", 1234);
```

```
Promocao promocao = new Promocao();
```

Instânciação e
Inicialização

Objetos

- Objetos são instâncias de uma classe de objetos na memória

```
Ponto p = new Ponto(4, 5);  
Ponto a = new Ponto(0, -1);
```

```
a.distancia(p);
```

Chamando métodos de outros objetos

Objetos

- Objetos são instâncias de uma classe de objetos na memória
- O operador **new** retorna uma referência para o novo objeto

Classes (2ª parte)

- Declarando variáveis de instância (de objetos)

[lista_modificadores] **tipo nome**

```
class Ponto {  
    int x;  
    int y;  
}
```

Classes (2ª parte)

- Declarando variáveis de instância (de objetos)

```
[lista_modificadores] tipo nome
```

```
class Ponto {  
    int x;  
    int y;  
}
```

```
class Retangulo {  
    Ponto origem;  
    int largura;  
    int altura;  
}
```

Classes (2ª parte)

- Declarando variáveis de instância (de objetos)

```
[lista_modificadores] tipo nome
```

```
class Ponto {
```

```
    ➡ int x;
```

```
    ➡ int y;
```

```
}
```

```
class Retangulo {
```

```
    ➡ Ponto origem;
```

```
    ➡ int largura;
```

```
    ➡ int altura;
```

```
}
```

Os tipos de variáveis podem ser primitivos ou outras classes

Classes (2ª parte)

- Declarando métodos de instância (de objetos)

```
[lista_modificadores] tipo_do_retorno nome(lista_parâmetros)
```

Classes (2ª parte)

- Declarando métodos de instância (de objetos)

```
[lista_modificadores] tipo_do_retorno nome(lista_parâmetros)
```

```
class Ponto {  
    int x;  
  
    int getX() {  
        return x;  
    }  
}
```

Classes (2ª parte)

- Declarando métodos de instância (de objetos)

```
[lista_modificadores] tipo_do_retorno nome(lista_parâmetros)
```

```
class Ponto {  
    int x;  
  
    int getX() {  
        return x;  
    }  
}
```

```
Ponto a = //...;
```

```
Ponto b = //...;
```

```
a.getX();
```

```
b.getX();
```


Classes (2ª parte)

- Declarando métodos de instância (de objetos)

```
[lista_modificadores] tipo_do_retorno nome(lista_parâmetros)
```

```
class Ponto {  
    int x;  
  
    int getX() {  
        return x;  
    }  
}
```

```
class Retangulo {  
  
    int getArea() {  
        return altura * largura;  
    }  
}
```

Classes (2ª parte)

- Declarando métodos de instância (de objetos)

```
[lista_modificadores] tipo_do_retorno nome(lista_parâmetros)
```

```
class FabricaCarro {  
    Carro criaCarro() {  
        return new Carro("Mercedes");  
    }  
}
```

Objetos construídos dentro de métodos não são problemas porque a memória é recolhida automaticamente pelo Java quando ele detecta que ela não é mais utilizada

Classes (2ª parte)

- Declarando variáveis e métodos estáticos (de classe)

```
class Retangulo {  
  
    static int quantidade = 0;  
    //... outras variáveis de instância  
  
    Retangulo() {  
        //...  
        quantidade += 1;  
    }  
  
    static int getQuantidade() {  
        return quantidade;  
    }  
  
}
```

Classes (2ª parte)

- Declarando variáveis e métodos estáticos (de classe)
- Modificador **static**
- Membros estáticos só podem acessar membros estáticos
- Membros de instância podem acessar membros estáticos

(desenho)