Variáveis

Profa. Karen Selbach Borges

Variáveis de Instância

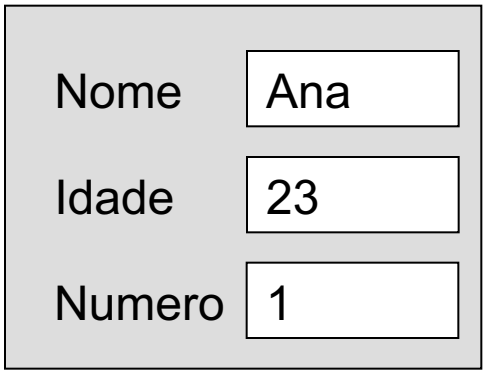
- Os atributos de um objeto também são chamados de variáveis de instância
- Cada objeto possui valores distintos para estas variáveis.

```
public class Pessoa {

   String nome;
    int idade;
    int numero;

    public Pessoa (String nome, int idade){
        this.nome = nome;
        this.idade = idade;
        numero ++;
    }
}
```

Execução 1



Variáveis de Instância

- Os atributos de um objeto também são chamados de variáveis de instância
- Cada objeto possui valores distintos para estas variáveis.

```
public class Pessoa {

   String nome;
    int idade;
    int numero;

    public Pessoa ( String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
        numero ++;
    }
}
```

Execução 1

Nome	Ana
Idade	23
Numero	1

Execução 2

Nome	Luis
Idade	51
Numero	1

Variáveis de Instância

- Os atributos de um objeto também são chamados de variáveis de instância
- Cada objeto possui valores distintos para estas variáveis.

```
public class Pessoa {

   String nome;
    int idade;
    int numero;

  public Pessoa (String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
        numero ++;
    }
}
```

Execução 1

Nome	Ana
Idade	23
Numero	1

Execução 2

Nome	Luis
Idade	51
Numero	1

Execução 3

Nome	João
Idade	35
Numero	1

Variáveis

- As variáveis de instância são inicializadas com valores default (padrão)

Variable Type	Default Value
Object reference	`null` (not referencing any object)
`byte`, `short`, `int`, `long`	`0`
`float`, `double`	`0.0`
`boolean`	`false`
`char`	`'\u0000'`

Variáveis Locais

- São aquelas declaradas na passagem de parâmetros ou dentro do método.
- Neste segundo caso, as variáveis devem obrigatoriamente ser inicializadas.
- Em ambos os casos, as variáveis deixam de existir após a execução do método.

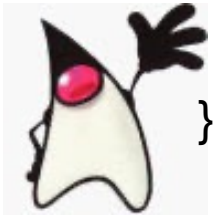
Variáveis Locais

```
public class Pessoa {
   String nome;
   int idade;
   String cpf;

 public Pessoa (String nome, int idade, String cpf) {
        this.nome = nome;
        this.idade = idade;
        this.cpf = cpf;
  }

 public boolean validaTamanhoCPF(){
       int tamanho = cpf.length();
       if (tamanho != 11) return false;
             else return true;
        } //fecha método
     } // fecha classe
```

Variáveis Locais



Variáveis de Referência

- São aquelas que correspondem as instâncias
- Funcionam como ponteiros para os objetos
- Então, são referências para áreas de memória

Variáveis de Referência

```
public class Pessoa {
   String nome;
   int idade;

   public Pessoa () { }

   public Pessoa (String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
   }

   public void copiaDados (Pessoa outraPessoa){
        System.out.println("OID outra = "+outraPessoa);
        this.nome =outraPessoa. nome;
        this.idade = outraPessoa.idade;
   }
}
```

```
public class Principal {

   public static void main (String[] args){
      Pessoa p1 = new Pessoa ("Ana", 33);
      System.out.println("OID p1 = "+p1);
      Pessoa p2 = new Pessoa();
      System.out.println ("OID p2 = "+p2);
      p2.copiaDados(p1);
   }
}
```

Variáveis de Referência

```
public class Pessoa {
   String nome;
   int idade;

   public Pessoa () { }

   public Pessoa (String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
   }

   public void copiaDados (Pessoa outraPessoa){
        System.out.println("OID outra = "+outraPessoa);
        this.nome =outraPessoa. nome;
        this.idade = outraPessoa.idade;
   }
}
```

```
public class Principal {

   public static void main (String[] args){
      Pessoa p1 = new Pessoa ("Ana", 33);
      System.out.println("OID p1 = "+p1);
      Pessoa p2 = new Pessoa();
      System.out.println ("OID p2 = "+p2);
      p2.copiaDados(p1);
   }
}
```

Saída:

OID p1 = Pessoa@276
OID p2 = Pessoa@861
OID outra = Pessoa@276

Variáveis de Referência

```
public class Pessoa {
   String nome;
   int idade;

   public Pessoa () { }

   public Pessoa (String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
   }

   public void copiaDados (Pessoa outraPessoa){
        System.out.println("OID outra = "+outraPessoa);
        this.nome =outraPessoa. nome;
        this.idade = outraPessoa.idade;
   }
}
```

```
public class Principal {

   public static void main (String[] args){
      Pessoa p1 = new Pessoa ("Ana", 33);
      System.out.println("OID p1 = "+p1);
      Pessoa p2 = new Pessoa();
      System.out.println ("OID p2 = "+p2);
      p2.copiaDados(p1);
   }
}
```

outraPessoa = p1

Observe que:

O OID de p1 e de outraPessoa é o mesmo ! Isso significa que ambas as variáveis apontam para o mesmo de endereçamento de memória.

Variáveis - outras

- No Java existe a possibilidade de declarar variáveis associadas a uma estrutura de controle.
- Essa variável só existirá durante a execução do código associado àquela estrutura.

Variáveis - outras

```
public class Main {

   public static void main (String[] args){

      for (int i = 0; i<10; i++){
         System.out.println(i);
       }

       System.out.println(i);
  }
}
```

Variável **i** só existe enquanto o comando for estiver sendo executado

```
Main.java:15: error: cannot find symbol
        System.out.println(i);
                           ^
  symbol:   variable i
  location: class Main
```