

Aula 03

Métodos e Array's

Pauta

- Métodos e campos
- Métodos e campos static
- import static
- Regras de escopo
- Sobrecarga de métodos

Métodos e campos

- Até agora:
 - acessíveis a partir de Instâncias da classe;
- Como acessá-los sem a necessidade de instâncias?
- Exemplo

Métodos e campos static

- Métodos e campos da classe (static);
- Métodos e campos do objeto (não static);
- Exemplos:
 - A classe Math;
 - `System.out.println();`
 - `Integer.parseInt();`
 - `Float.parseFloat();`

Métodos e campos static – a classe Math

- `abs(x)` valor absoluto
- `cos(x)` co-seno
- `exp(x)` exponencial
- `log(x)` logaritmo natural
- `max(x, y)` maior valor de x e y
- ... outros ...

import static

- Foi implementando no JSE 5.0
- Importa métodos e campos declarados como static em uma classe.
- Sem imports:
 - `double ang = java.lang.Math.PI / 2.0;`
 - `double x = java.lang.Math.sin(ang)`
-

... import static

- Com import normal:
 - `import java.lang.Math;`
 - `double ang = Math.PI / 2.0;`
 - `double x = Math.sin(ang)`
- Com imports
 - `import java.lang.Math;`
 - `Import static java.lang.Math.PI;`
 - `double ang = PI / 2.0;`
 - `double x = Math.sin(ang);`

Promoção de argumentos

- Na passagem de parâmetros para um método o Java converte o valor automaticamente.Ex.:
 - `Math.sqrt(4);`
- o método `sqrt` espera receber um tipo `double`. O tipo `int` então será convertido para um tipo `double`. Nunca o contrário.

Regras de escopo

- O escopo de uma declaração é a parte do programa que pode referenciar a entidade declarada pelo seu nome;
- O escopo é definido por { }

Sobrecarga de método

- Assinatura do método;
- Sobrecarga de métodos são varios métodos com o mesmo nome mas assinaturas diferentes;
- Exemplo

Escopo de variáveis

Campos de dados (declarados no bloco da classe): podem ser usadas em qualquer lugar (qualquer bloco) da classe

- *Uso em outras classes depende de modificadores de acesso (public, private, etc.)*
- *Existem enquanto o objeto existir (ou enquanto a classe existir, se declarados static)*

Variáveis locais (declaradas dentro de blocos de procedimentos)

- *Existem enquanto procedimento (método, bloco de controle de execução) estiver sendo executado*
- *Não podem ser usadas fora do bloco*
- *Não pode ter modificadores de acesso (private, public, etc.)*

EXEMPLO

variáveis visíveis dentro da classe, apenas

```
public class Circulo {
```

```
    private int raio;
```

```
    private int x, y;
```

```
    public double area() {
```

```
        return Math.PI * raio * raio;
```

```
    }
```

novoRaio é variável local ao método mudaRaio

```
    public void mudaRaio(int novoRaio) {
```

```
        int maxRaio = 50;
```

```
        if (novoRaio > maxRaio) {
```

```
            raio = maxRaio;
```

```
        }
```

```
        if (novoRaio > 0) {
```

```
            int inutil = 0;
```

```
            raio = novoRaio;
```

```
        }
```

```
    }
```

```
}
```

maxRaio é variável local ao método mudaRaio

raio é variável de instância

inutil é variável local ao bloco if

Revisão

- Métodos e campos
- Métodos e campos static
- import static
- Regras de escopo
- Sobrecarga de métodos

Dúvidas?

Array

Pauta

- O que são array's
- Declaração de array
- Array multidimensionais
- For aprimorado – foreach
- Passagem de parâmetros
- Lista de parâmetros de comprimento variável

O que são array's

- É um grupo de variáveis que contém valores que são todos do mesmo tipo;
- É um objeto;
- Possui um índice que deve ser inteiro e não negativo;

Declaração de array

- `int [] i = new int[12];`
- `int i[] = new int[12];`
- `int i[] = {1,2,3,4,5,6,7,8,9,10};`
- `int i[];`
- `i = new int[12];`

Declaração de array

- NUNCA Assim:
 - `int i[12]; (c/c++)`

Declaração de array

- Exemplo
- Exemplo com array de clientes

For aprimorado - foreach

```
for( parâmetro : nomeDoArray ) {  
  
}
```

For aprimorado - foreach

- Itera pelos elementos de um array;
- Usada somente para ler os valores do array, nunca para modifica-los;

Array multidimensionais

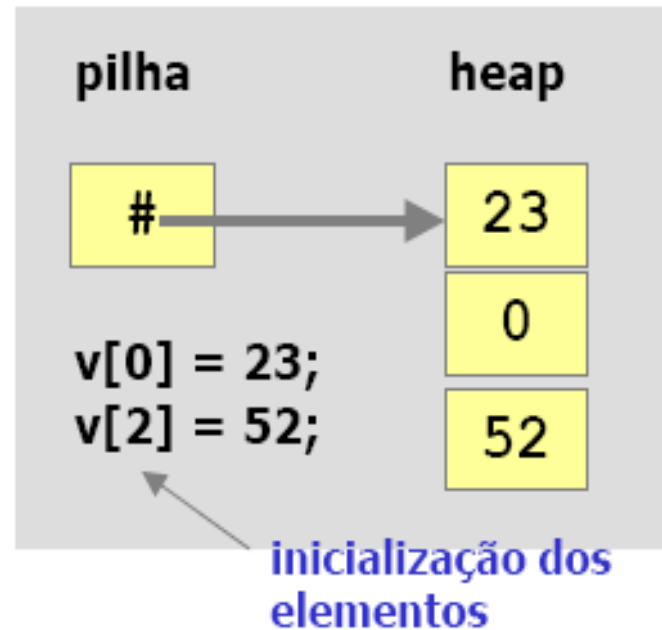
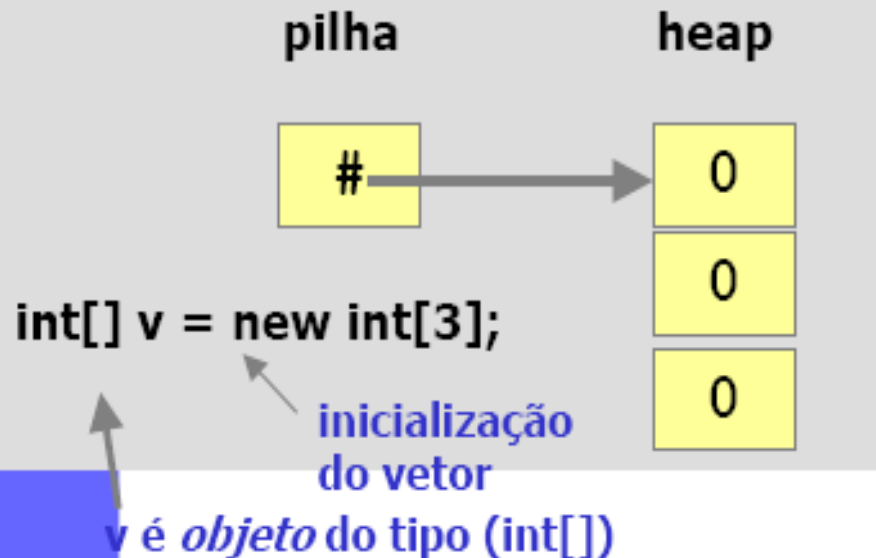
- Um array pode ter quantas dimensões forem necessárias para o programa;
- Declara da seguinte forma:
 - `int [][]...[] variavel = new int[2][4]...[8]`
- Não é uma boa prática de programação

Passagem de parâmetros

- Tipos **primitivos** são copiados;
- **Objetos** tem sua referência copiada;
- Passagem de array para métodos
- Exemplos

Entendendo Melhor

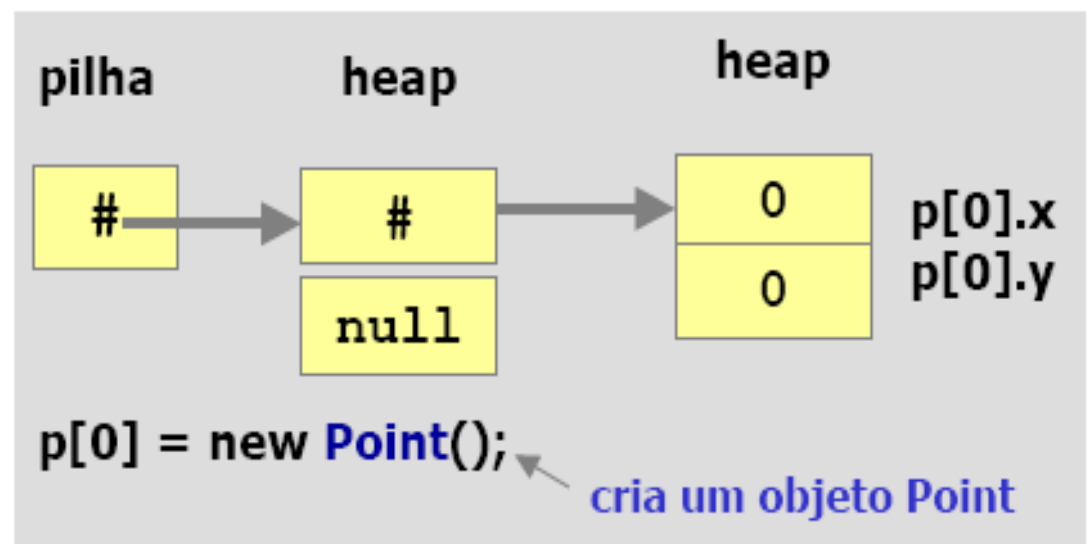
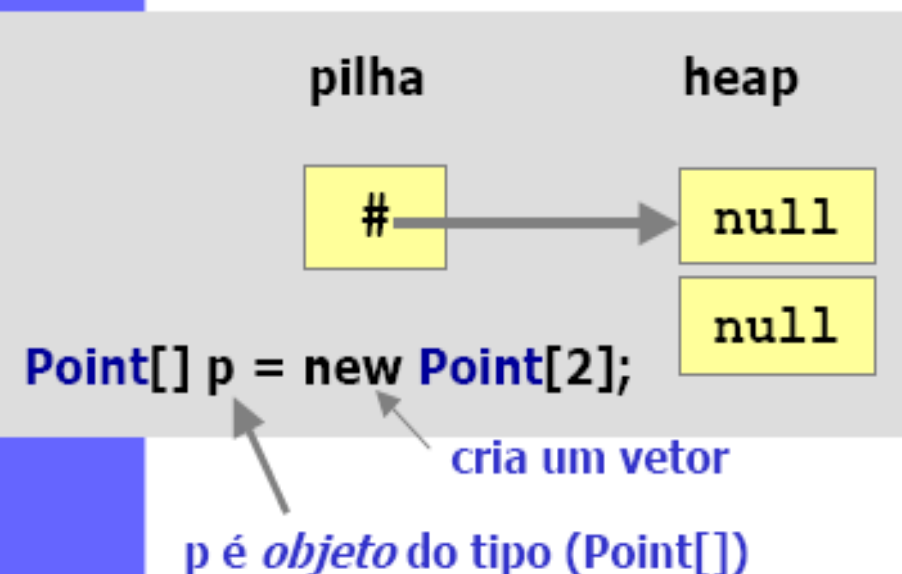
- de tipos primitivos



```
class Point {  
    public int x;  
    public int y;  
}
```

Point
+x: int
+y: int

- de objetos (*Point* é uma classe, com dois membros `x` e `y`, inteiros)



Lista de parâmetros de comprimento variável

- Pode ser utilizado no lugar de array para passagem de parâmetros do mesmo tipo, mas com quantidades variáveis;
- É declarado utilizando reticências (...)
- Deve ser colocado no fim da lista de parâmetros ;
- Ex.: `public void metodo(int ... variavel)`

Revisão

- O que são array's
- Declaração de array
- Array multidimensionais
- For aprimorado – foreach
- Passagem de parâmetros
- Lista de parâmetros de comprimento variável
- Escopo de variáveis

Duvidas??

Exercícios

- Construa uma classe chamada Calculadora com os métodos somar, multiplicar definidos de forma que aceitem parametros variáveis;
- Desenvolva um método main que execute os métodos somar e multiplicar com parâmetros variados e imprima o resultado.

Exercícios

- Escreva um aplicativo que calcule o salário dos funcionários abaixo em função das vendas;
 - Cálculo do salário: \$ 200,00 + 8% sobre as vendas brutas

Nome	Vendas Brutas	Salário
Maria	\$ 10000	
Antonio	\$ 17500	
Carlos	\$ 12350	
Joaquim	\$ 45210	

Fim