

Programação Orientada a Objetos

Tipos, Ponteiros e Wrapper

André Santanchè

Laboratory of Information Systems – LIS

Instituto de Computação – UNICAMP

Junho 2020

Problema

```
vetor[0] = 15;  
vetor[1] = 5.6f;  
vetor[2] = "dinossauro";
```

```
for (int v = 0; v < vetor.length; v++)  
    System.out.println(vetor[v]);
```

Problema

- Considere que você precisa criar um vetor que armazene tipos diferentes de dados:

```
vetor[0] = 15;
```

```
vetor[1] = 5.6f;
```

```
vetor[2] = "dinossauro";
```

```
for (int v = 0; v < vetor.length; v++)
```

```
    System.out.println(vetor[v]);
```



Como fazer isso em Java?

Isto está em Java

```
7      vetor[0] = 15;  
8      vetor[1] = 5.6f;  
9      vetor[2] = "dinossauro";  
10  
11      for (int v = 0; v < vetor.length; v++)  
12          System.out.println(vetor[v]);  
13  }  
14 }
```

Problems @ Javadoc Declaration Console

<terminated> AppVetorNumeros [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java

15
5.6
dinossauro



Como?



Tipos de Dados Primitivos X Classes

Voltando ao Exemplo do Círculo




```
public class Circulo {  
    private int centroX, centroY;  
    private int raio;  
  
    Circulo(int centroX, int centroY, int raio) {  
        this.centroX = centroX;  
        this.centroY = centroY;  
        this.raio = raio;  
    }  
  
    public int getCentroX() {  
        return centroX;  
    }  
  
    public int getCentroY() {  
        return centroY;  
    }  
  
    public double area() {  
        return Math.PI * raio * raio;  
    }  
}
```

Circulo

- centroX: int
- centroY: int
- raio: int

- + getCentroX(): int
- + getCentroY(): int
- + area(): double

Duplos Papel das Variáveis

- Variáveis que contêm dados de **tipos primitivos**
- Variáveis que são **ponteiros** para objetos

```
int raio = 10;
```

```
Circulo circ = new Circulo(5, 3, raio);
```

Duplo Papel das Variáveis

```
int raio = 10;
```

```
Circulo circ = new Circulo(5, 3, raio);
```

Duplo Papel das Variáveis

```
int raio = 10;
```

raio

10

Duplo Papel das Variáveis

```
int raio = 10;
```

```
Circulo circ = new Circulo(5, 3, raio);
```

raio

10

circ

centroX

5

centroY

3

raio

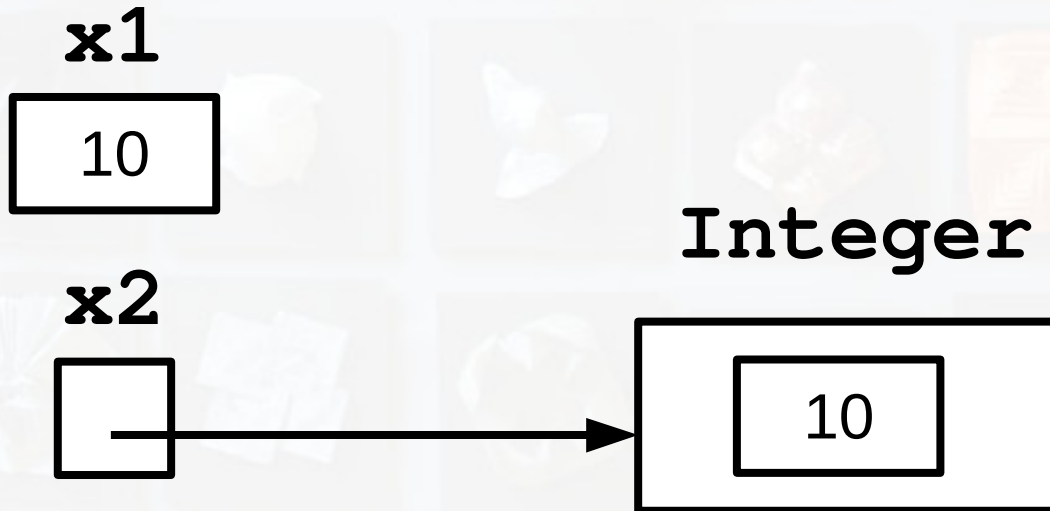
10

Wrapper

- Tipos primitivos tratados como objetos:

```
int x1 = 10;
```

```
Integer x2 = new Integer(10);
```



Wrapper Classes

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
long	Integer
float	Float
double	Double
boolean	Boolean

Tarefa

- Liste quais as vantagens e desvantagens de representar tipos primitivos na forma de objetos.

Vantagens x Desvantagens

■ Vantagens

- Polimorfismo com tipos primitivos

■ Desvantagens

- Memória
- Tempo de processamento

Conversões Automáticas

- Simplifica o uso
- Torna o wrapper transparente

Criações Explícitas

// criação explícita de tipo primitivo

```
int x1 = 10;
```

// criação explícita de objeto wrapper

```
Integer x2 = new Integer(10);
```

```
Integer x3 = new Integer("10");
```

Conversões Explícitas

// conversão explícita tipo primitivo -> wrapper

```
Integer x4 = new Integer(x1);
```

// conversão explícita wrapper -> tipo primitivo

```
int x5 = x2.intValue();
```

Conversões Implícitas

// conversão implícita tipo primitivo -> wrapper

```
Integer x6 = 10;
```

// conversão implícita wrapper -> tipo primitivo

```
int x7 = new Integer(10);
```

```
int x8 = new Integer("10");
```

Conversões Implícitas

// convertidos explicitamente e somados

```
System.out.println("soma: " +  
    (x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8));
```

The background of the slide is a dense collage of various paper crafts, including folded papers, origami, and cutouts, arranged in a grid-like pattern. The colors are muted and earthy, with shades of beige, cream, and light brown. The text is centered over this collage.

O mesmo tipo de conversão acontece na
passagem de parâmetros

Wrapper no vetor

AppVetorNumeros.java

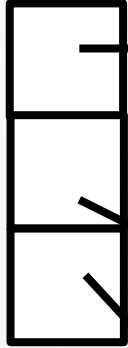
```
1 package pt.c03java.s06wrapping.s03vetor;  
2  
3 public class AppVetorNumeros {  
4     public static void main(String args[]) {  
5         Object vetor[] = new Object[3];  
6  
7         vetor[0] = 15;  
8         vetor[1] = 5.6f;  
9         vetor[2] = "dinossauro";  
10  
11         for (int v = 0; v < vetor.length; v++)  
12             System.out.println(vetor[v]);  
13     }  
14 }
```

Problems @ Javadoc Declaration Console

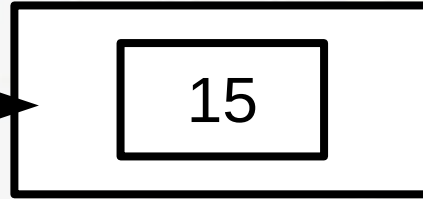
<terminated> AppVetorNumeros [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java

```
15  
5.6  
dinossauro
```

vetor
(Object)



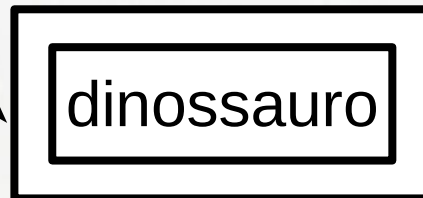
Integer



Float



String



Tarefa

- Criar um vetor que:
 - receba qualquer tipo de objeto
 - comece com 10 posições
 - depois de preencher as 10 posições, cresça na medida em que você acrescenta elementos
- Importante: o Java já possui um classe que faz isso, você não deve usá-la, deve criar a sua

André Santanchè

<http://www.ic.unicamp.br/~santanche>

Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.
- Mais detalhes sobre a referida licença Creative Commons veja no link:
<http://creativecommons.org/licenses/by-nc-sa/3.0/>
- Agradecimento a Goran Konjevod [<https://www.flickr.com/photos/23913057@N05/>] por sua fotografia “50-50 Show III” usada na capa e nos fundos, disponível em [<https://flic.kr/p/advD33>] vide licença específica da fotografia.