

# AfroDev

Java Básico - Aula 3 - Parte 1

# Casting - Definição e regras

- Casting (formalmente type casting) é o fato de você atribuir um tipo primitivo a outro tipo primitivo
- Regras e exemplos: [https://www.w3schools.com/java/java\\_type\\_casting.asp](https://www.w3schools.com/java/java_type_casting.asp)

# Casting - Exercício

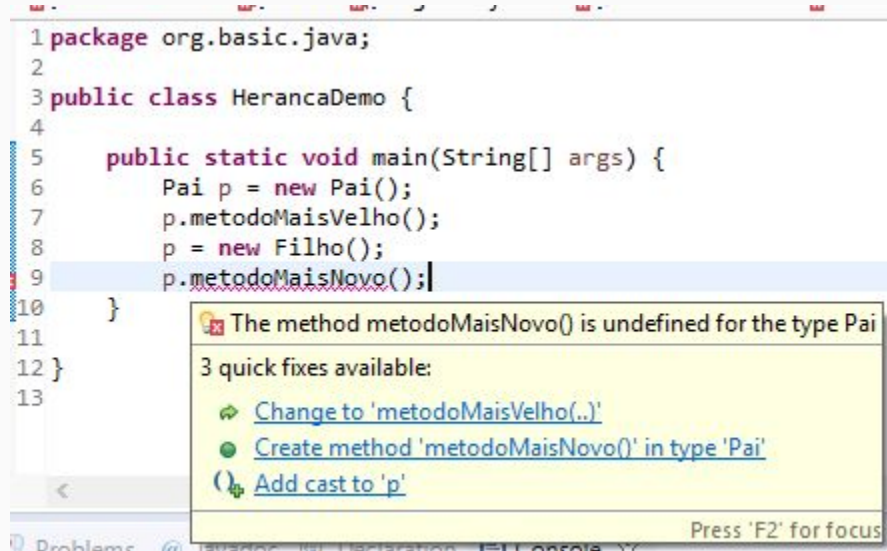
- Para aquecer, escreva o exemplo do slide anterior localmente
- Agora, atribua um tipo “maior” a um tipo “menor” SEM o cast manual. Exemplo: um `double` -> `int`. O que aconteceu?
- Mesmo manualmente permitido, qual problema existe ao fazer um cast manual?
- Na sua opinião, cast automático tem problema?

# Casting de tipos (Herança) - Exemplo

```
1 package org.basic.java;
2
3 public class Pai {
4
5     public void metodoMaisVelho() {
6         System.out.println("algum metodo Pai");
7     }
8
9 }
10
```

```
1 package org.basic.java;
2
3 public class Filho extends Pai {
4
5     public void metodoMaisNovo() {
6         System.out.println("algum metodo Filho");
7     }
8
9 }
10
```

# Casting de tipos (Herança) - Erro Upcast



# Casting de tipos (Herança) - Erro Downcast

```
1 package org.basic.java;  
2  
3 public class HerancaDemo {  
4  
5     public static void main(String[] args) {  
6         Filho f = new Filho();  
7         f.metodoMaisVelho();  
8         f = new Pai();  
9     }  
10  
11 }  
12
```

Type mismatch: cannot convert from Pai to Filho

2 quick fixes available:

- [Add cast to 'Filho'](#)
- [Change type of 'f' to 'Pai'](#)

Press 'F2' for focus

# Casting de tipos (Herança) - OK

```
1 package org.basic.java;
2
3 public class HerancaDemo {
4
5     public static void main(String[] args) {
6         Pai p = new Pai();
7         p.metodoMaisVelho();
8         p = new Filho();
9         p.metodoMaisVelho();
10    }
11
12 }
13
```

Problems @ Javadoc Declaration Console

<terminated> HerancaDemo [Java Application] C:\Program Files\

algum metodo Pai

algum metodo Pai

# Casting de tipos (Herança) - Sobrecarga

```
1 package org.basic.java;
2
3 public class Filho extends Pai {
4
5     public void metodoMaisNovo() {
6         System.out.println("algum metodo Filho");
7     }
8
9     @Override
10    public void metodoMaisVelho() {
11        //super.metodoMaisVelho();
12        System.out.println("metodoMaisVelho do Filho");
13    }
14 }
```

```
1 package org.basic.java;
2
3 public class HerancaDemo {
4
5     public static void main(String[] args) {
6         Pai p = new Pai();
7         p.metodoMaisVelho();
8         p = new Filho();
9         p.metodoMaisVelho();
10    }
11
12 }
13
```

Problems @ Javadoc Declaration Console

<terminated> HerancaDemo [Java Application] C:\Program File:  
algum metodo Pai  
metodoMaisVelho do Filho



# Instanceof

- Interroga se um determinado objeto é de um determinado tipo (classe)
- Retorna “true” (boolean) se verdadeiro, “false” caso contrário

# Instanceof - Exercício

- Qual a saída do programa abaixo?

```
3 public class HerancaDemo {  
4  
5     public static void main(String[] args) {  
6         Pai p = new Pai();  
7         Filho f = new Filho();  
8  
9         if (p instanceof Pai) {  
10             System.out.println("p instanceof Pai");  
11         }  
12         if (p instanceof Filho) {  
13             System.out.println("p instanceof Filho");  
14         }  
15         if (f instanceof Pai) {  
16             System.out.println("f instanceof Pai");  
17         }  
18         if (f instanceof Filho) {  
19             System.out.println("f instanceof Filho");  
20         }  
21     }  
22 }
```

# Instanceof - Exercício

- Qual a saída do programa abaixo?

```
3 public class HerancaDemo {  
4  
5     public static void main(String[] args) {  
6         Pai p = new Pai();  
7         p = new Filho();  
8  
9         Filho f = new Filho();  
10  
11         if (p instanceof Pai) {  
12             System.out.println("p instanceof Pai");  
13         }  
14         if (p instanceof Filho) {  
15             System.out.println("p instanceof Filho");  
16         }  
17         if (f instanceof Pai) {  
18             System.out.println("f instanceof Pai");  
19         }  
20         if (f instanceof Filho) {  
21             System.out.println("f instanceof Filho");  
22         }  
23  
24     }  
25  
26 }
```

# Final

- Palavra reservada
- Significa literalmente um “fim” (um basta) para as coisas
- Classes final
- Métodos final
- Variáveis final

# Final - Exercícios

- Torne a classe Pai como final. O que aconteceu?
- Torne o método metodoMaisVelho da classe Pai como final. O que aconteceu?
- Em um método qualquer crie uma variável final e tenta atribuir um valor a ela pela segunda vez. O que aconteceu?
- Na sua opinião, pra que serve algo “final”?

# Revisão

- Reforçamos conceitos muito importantes de OO sobre Herança
- Vimos que upcast e downcast, tanto para primitivos quanto para tipos, possuem regras bem específicas
- Ainda aprendemos que classes filhos podem modificar o comportamento da classe pai através da sobrecarga
- Por fim, vimos que “final” põe um “ponto final nas coisas”.