

# Orientação a Objetos II

## Debugging

Prof. Felipe Scheidt  
2016

Instituto Federal do Paraná



# Introdução

- *Debugging* é um processo metódico de encontrar e reduzir ou eliminar *bugs* (defeitos) em um programa de computador, com o objetivo de fazer o código ter o resultados esperado.

# O que é um bug?

- Um *bug* de software pode ser um erro ou uma falha que leva um programa de computador a se comportar de maneira incorreta ou inesperada.
- A maioria dos *bugs* surge a partir da desatenção ou de erros cometidos por pessoas.
- *Bugs* raramente são ocasionados por compiladores ou interpretadores.

# Tipos de bugs

- Que causam uma exceção: sistema congela ou interrompe sua execução.
- Que levam a uma inconsistência lógica: sistema apresenta um comportamento ilógico.

# Exemplo #1

- Onde está o problema? O código abaixo compila, porém lançará uma exceção em tempo de execução.

```
1 package debugging;
2 public class Exemplo {
3     private String nome;
4     public static void main(String[] args) {
5         Exemplo e = new Exemplo();
6         System.out.println(e.getNome().toLowerCase());
7     }
8     public String getNome() {
9         return nome;
10    }
11    public void setNome(String nome) {
12        this.nome = nome;
13    }
14 }
```

# Exemplo #2

- O código abaixo não gera nenhum erro ou lançamento de exceção, porém possui um erro (bug) de lógica, que leva sempre ao mesmo resultado, independente do estado da variável *isMemoryFull*

```
public static void main(String[] args) {
    boolean isMemoryFull=false;
    if(isMemoryFull=true){
        System.out.println("Fazer upgrade no servidor!");
        System.out.println("Memoria Cheia: " + isMemoryFull);
    }else{
        System.out.println("Servidor com baixo uso.");
    }
}
```

# Debugging no Eclipse

The screenshot shows the Eclipse IDE interface during a debug session. The title bar reads "Debug - OrientacaoObjetos2/src/debugging/Exercicio3.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and debugging.

The Debug console on the left shows the following structure:

- Exercicio3 [Java Application]
  - debugging.Exercicio3 at localhost:51292
    - Thread [main] (Suspended)
      - Exercicio3.main(String[]) line: 6

The Variables view on the right shows the following data:

Name	Value
args	String[0] (id=16)
num1	Double (id=19)
num2	Double (id=22)

The Outline view on the right shows the following structure:

- debugging
  - Exercicio3
    - main(String[]): void
    - calculaOperacao(Object, Object, String): Object

The main editor window shows the source code of Exercicio3.java:

```
3 public class Exercicio3 {
4     public static void main(String[] args) {
5         Double num1 = 4d; Double num2 = 2d;
6         Double res = (Double)calculaOperacao(num1,num2,"sum");
7         System.out.println("o total da operacao eh: " + res);
8     }
9     public static Object calculaOperacao(Object o1, Object o2, String operacao
10        if(operacao.equals("soma"))
11        return (Double)o1 + (Double)o2;
12        else if(operacao.equals("subtracao"))
```

The Console view at the bottom shows the following text:

```
Exercicio3 [Java Application] C:\Program Files\Java\jre1.8.0_20\bin\javaw.exe (30/03/2015 12:33:10)
```

The status bar at the bottom right shows "Writable", "Smart Insert", and "6:1".

# Exercícios

- Instruções para responder cada um dos exercícios a seguir:
- Usando a ferramenta de *debug* do eclipse encontre o problema no código e responda:
  - 1) Em qual linha do código esta o(s) *bug(s)*?
  - 2) Se há, qual o nome da exceção lançada? Qual o seu significado?
  - 3) reescreva o código para que seu funcionamento fique correto.

# Exercício 1

```
1 package debugging;
2 /**
3  * Calcula a media dos valores de um array
4  */
5 public class Exercicio1 {
6     public static void main(String[] args) {
7         int[] notas = {6,4,8};
8         double media = 0d;
9         for(int p=0;p<=notas.length;p++){
10             System.out.println(notas[p]);
11             media=notas[p]+media;
12         }
13         media=media/notas.length;
14         System.out.println("Media: " + media);
15     }
16 }
```

# Exercício 2

```
1 package debugging;
2 /**
3  * Calcula o total de pixels numa região
4  * Ex.: se 4 linhas e 4 colunas, total de pixel = 16
5  */
6 public class Exercicio2 {
7     public static void main(String[] args) {
8         int i,j;
9         int numLinhas = 2;
10        int numColunas = 4;
11        int pixels=0;
12        for (i=0; i<numLinhas; i++)
13            for (j=0; j<numColunas; j++);
14            pixels++;
15        System.out.println("Quantidade de pixels: " + pixels);
16    }
17 }
```

# Exercício 3

```
1 package debugging;
2
3 public class Exercicio3 {
4     public static void main(String[] args) {
5         Double num1 = 4d; Double num2 = 2d;
6         Double res = (Double)calculaOperacao(num1,num2,"sum");
7         System.out.println("o total da operacao eh: " + res);
8     }
9     public static Object calculaOperacao(Object o1, Object o2, String operacao){
10         if(operacao.equals("soma"))
11             return (Double)o1 + (Double)o2;
12         else if(operacao.equals("subtracao"))
13             return (Double)o1 - (Double)o2;
14         else if(operacao.equals("divisao"))
15             return (Double)o1 / (Double)o2;
16         else if(operacao.equals("multiplicacao"))
17             return (Double)o1 * (Double)o2;
18         else
19             return "operacao desconhecida";
20     }
21 }
```

# Exercício 4

```
1 package debugging;
2 public class Exercicio4 {
3     private static int REFS = 0;
4     private static int a;
5     private static int b;
6     private static int c;
7
8     public Exercicio4(int a, int b, int c){
9         REFS++;
10        this.a = a;
11        this.b = b;
12        this.c = c;
13    }
14    public String toString() {
15        return "Valores do Objeto: ["
16            + a + ", " + b + ", " + c+"]";
17    }
18    public static void main(String[] args) {
19        Exercicio4 objA = new Exercicio4(2,31,8);
20        Exercicio4 objB = new Exercicio4(9,11,1);
21        System.out.println(objA);
22        System.out.println(objB);
23        System.out.println("Total de objetos criados: " + REFS);
24    }
25 }
```