

INF335 – AMBIENTES PARA CONCEPÇÃO DE SOFTWARE

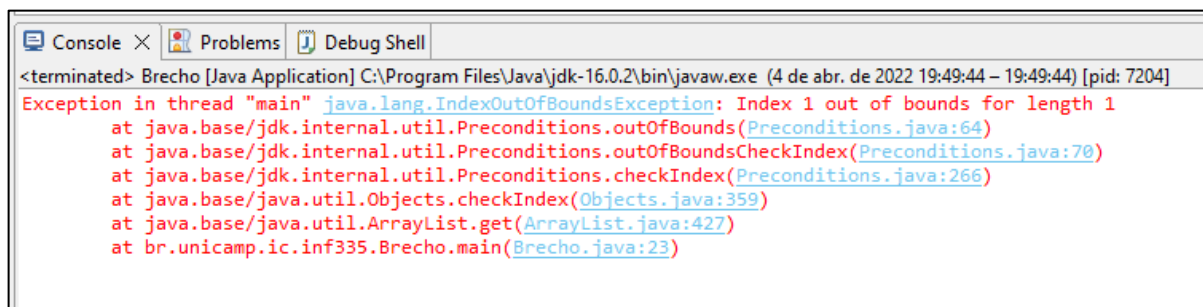
TRABALHO AULA 01 – IDE's E DEBUGGER

Acesso ao código corrigido: https://github.com/pedrolivr/inf_335_tarefa1

1 Problemas encontrados ao executar a classe *main* pela primeira vez

Ao executar a classe *main*, recebemos a exceção da Figura 1, onde o erro consiste na tentativa de leitura do index 1, em uma lista que não possui esse index.

Figura 1

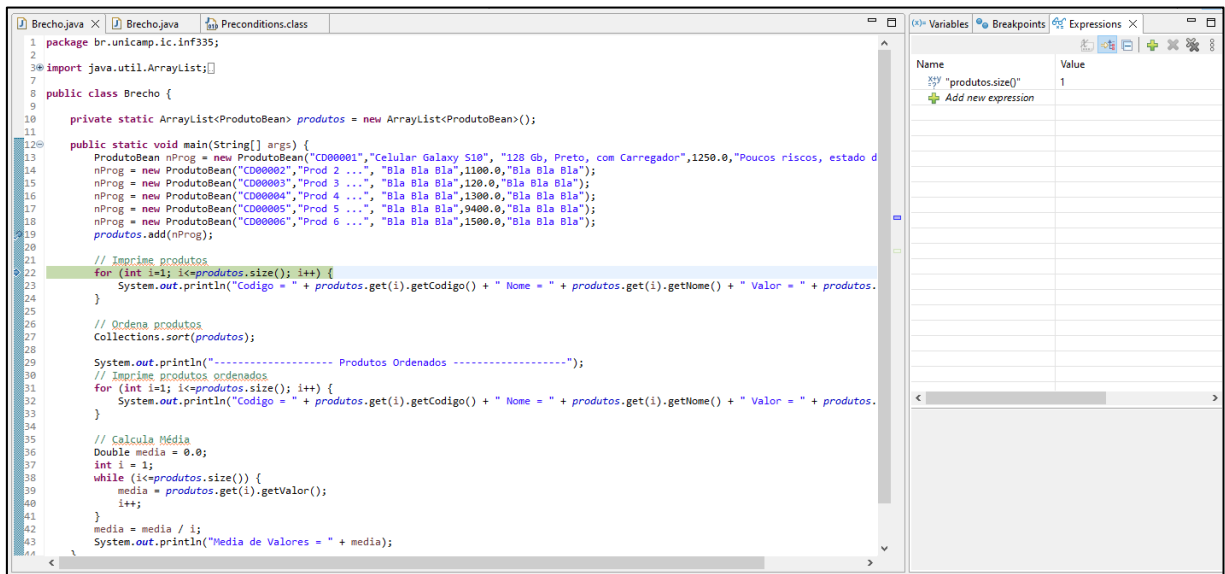


```
Console X Problems Debug Shell
<terminated> Brecho [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (4 de abr. de 2022 19:49:44 – 19:49:44) [pid: 7204]
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 1 out of bounds for length 1
    at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)
    at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:70)
    at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:266)
    at java.base/java.util.Objects.checkIndex(Objects.java:359)
    at java.base/java.util.ArrayList.get(ArrayList.java:427)
    at br.unicamp.ic.inf335.Brecho.main(Brecho.java:23)
```

2 DEBUGGER STEP OVER - LINHA A LINHA

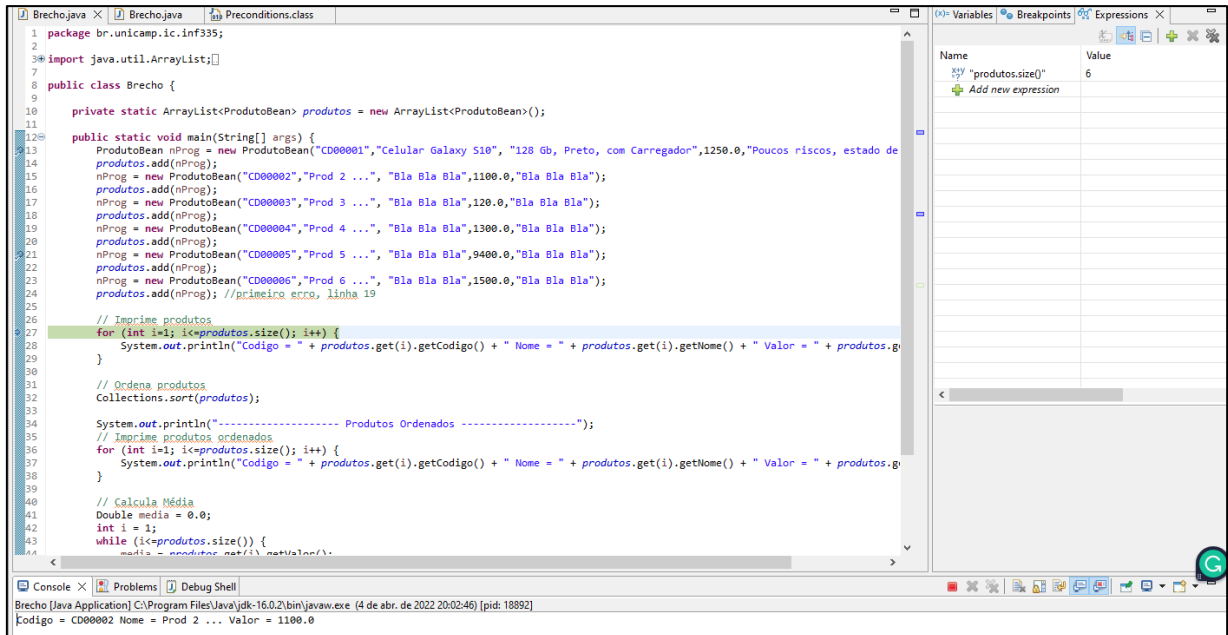
Um *markpoint* foi inserido para que fosse possível checar o tamanho da lista dos produtos. Nota-se na Figura 2 que a lista possui apenas um item, pois o método *produtos.add(object)* é aplicado para o ultimo produto apenas. Uma expressão foi adicionada para checar o tamanho da lista, que tem o tamanho de 01 (um item).

Figura 2



Para a correção, o método *produtos.add()* foi inserido logo abaixo de todos os objetos criados, para que integrassem a lista principal dos *produtos*. A mesma expressão foi validada ao retornar o valor de 6 itens.

Figura 3



```
1 package br.unicamp.ic.inf335;
2
3 import java.util.ArrayList;
4
5 public class Brecho {
6
7     private static ArrayList<ProdutoBean> produtos = new ArrayList<ProdutoBean>();
8
9     public static void main(String[] args) {
10         ProdutoBean nProd = new ProdutoBean("CD00001", "Celular Galaxy S10", "128 Gb, Preto, com Carregador", 1250.0, "Poucos riscos, estado de
11         produtos.add(nProd);
12         nProd = new ProdutoBean("CD00002", "Prod 2 ...", "Bla Bla Bla", 1100.0, "Bla Bla Bla");
13         produtos.add(nProd);
14         nProd = new ProdutoBean("CD00003", "Prod 3 ...", "Bla Bla Bla", 120.0, "Bla Bla Bla");
15         produtos.add(nProd);
16         nProd = new ProdutoBean("CD00004", "Prod 4 ...", "Bla Bla Bla", 1300.0, "Bla Bla Bla");
17         produtos.add(nProd);
18         nProd = new ProdutoBean("CD00005", "Prod 5 ...", "Bla Bla Bla", 9400.0, "Bla Bla Bla");
19         produtos.add(nProd);
20         nProd = new ProdutoBean("CD00006", "Prod 6 ...", "Bla Bla Bla", 1500.0, "Bla Bla Bla");
21         produtos.add(nProd); //primeiro erro, linha 19
22
23         // Imprime produtos
24         for (int i=1; i<produtos.size(); i++) {
25             System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.g
26         }
27
28         // Ordena produtos
29         Collections.sort(produtos);
30
31         System.out.println("----- Produtos Ordenados -----");
32         // Imprime produtos ordenados
33         for (int i=1; i<produtos.size(); i++) {
34             System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.g
35         }
36
37         // Calcula Média
38         Double media = 0.0;
39         int i = 1;
40         while (i<produtos.size()) {
41             media = produtos.get(i).getValor();
42         }
43     }
44 }
```

Console: Brecho [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (4 de abr. de 2022 20:02:46) [pid: 18892]
Codigo = CD00002 Nome = Prod 2 ... Valor = 1100.0

Após as alterações realizadas, executou-se novamente o código, a mesma exceção foi apresentada, porém o problema se encontra na forma com que a lista é iterada, onde o contador deve ser iniciado em “0”.

Figura 4

The screenshot shows an IDE with a Java file named 'Brecho.java'. The code defines a 'ProdutoBean' class and a 'Brecho' class. The 'main' method in 'Brecho' creates a list of products and iterates over them. The iteration starts at index 1, which causes an 'Index 6 out of bounds for length 6' exception when it reaches the 6th element (index 5). The console output shows the first five products correctly, followed by the exception message and a stack trace pointing to line 28 of 'Brecho.java'.

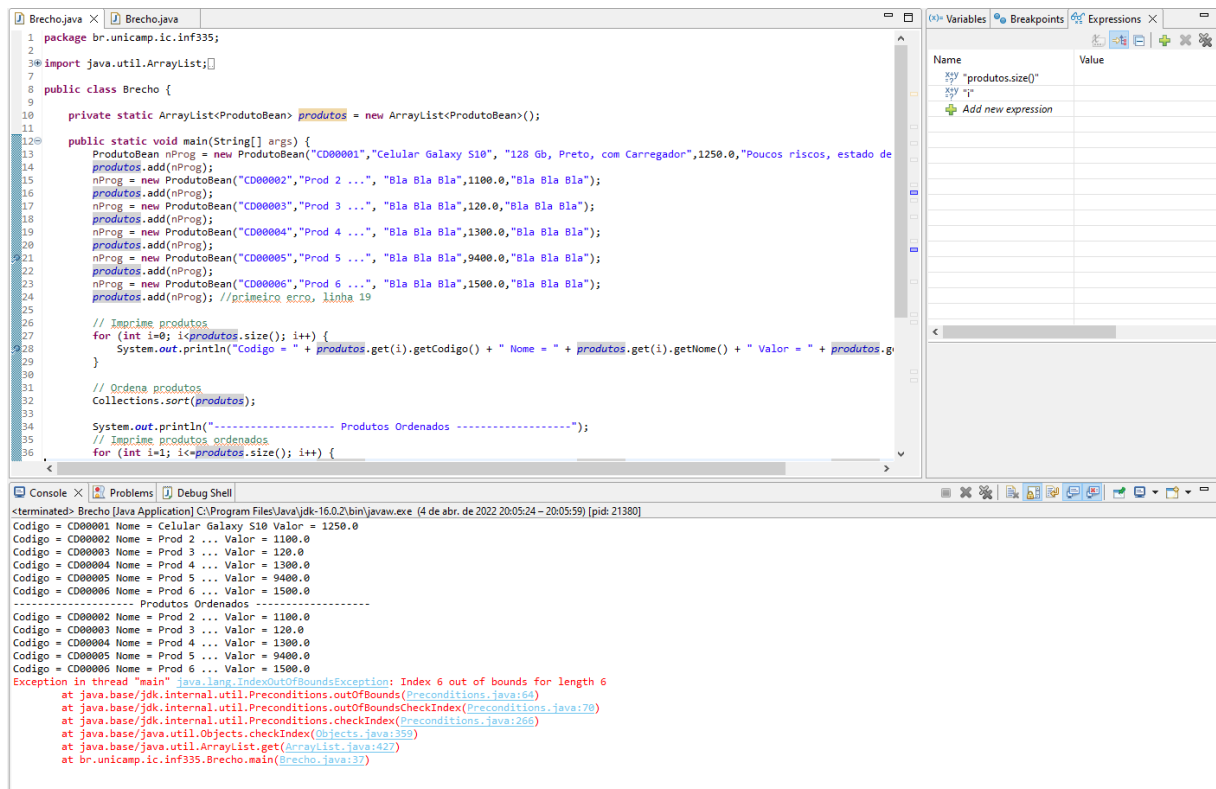
```
1 package br.unicamp.ic.inf335;
2
3 import java.util.ArrayList;
4
5 public class Brecho {
6
7     private static ArrayList<ProdutoBean> produtos = new ArrayList<ProdutoBean>();
8
9     public static void main(String[] args) {
10         ProdutoBean nProd = new ProdutoBean("CD000001", "Celular Galaxy S10", "128 Gb, Preto, com Carregador", 1250.0, "Poucos riscos, estado de novo.");
11         produtos.add(nProd);
12         nProd = new ProdutoBean("CD000002", "Prod 2 ...", "Bla Bla Bla", 1100.0, "Bla Bla Bla");
13         produtos.add(nProd);
14         nProd = new ProdutoBean("CD000003", "Prod 3 ...", "Bla Bla Bla", 120.0, "Bla Bla Bla");
15         produtos.add(nProd);
16         nProd = new ProdutoBean("CD000004", "Prod 4 ...", "Bla Bla Bla", 1300.0, "Bla Bla Bla");
17         produtos.add(nProd);
18         nProd = new ProdutoBean("CD000005", "Prod 5 ...", "Bla Bla Bla", 9400.0, "Bla Bla Bla");
19         produtos.add(nProd);
20         nProd = new ProdutoBean("CD000006", "Prod 6 ...", "Bla Bla Bla", 1500.0, "Bla Bla Bla");
21         produtos.add(nProd); //primeira erro, linha 19
22
23         // Imprime produtos
24         for (int i=1; i<produtos.size(); i++) {
25             System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.get(i).getValor());
26         }
27
28         // Ordena produtos
29         Collections.sort(produtos);
30
31         System.out.println("----- Produtos Ordenados -----");
32         // Imprime produtos ordenados
33         for (int i=0; i<produtos.size(); i++) {
34             System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.get(i).getValor());
35         }
36
37         // Calcula Média
38         Double media = 0.0;
39     }
40 }
```

Console Output:

```
<terminated> Brecho [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (5 de abr. de 2022 19:27:22 - 19:27:24) [pid: 13224]
Codigo = CD000002 Nome = Prod 2 ... Valor = 1100.0
Codigo = CD000003 Nome = Prod 3 ... Valor = 120.0
Codigo = CD000004 Nome = Prod 4 ... Valor = 1300.0
Codigo = CD000005 Nome = Prod 5 ... Valor = 9400.0
Codigo = CD000006 Nome = Prod 6 ... Valor = 1500.0
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 6 out of bounds for length 6
    at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:61)
    at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:76)
    at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:266)
    at java.base/java.util.Objects.checkIndex(Objects.java:359)
    at java.base/java.util.ArrayList.get(ArrayList.java:427)
    at br.unicamp.ic.inf335.Brecho.main(Brecho.java:28)
```

A solução foi inicializar a variável “i” como “0” e deixar o operador como “<” que condiz com o tamanho total do *array*. Após a execução, nota-se ainda que a média apresenta erros, novamente quanto a leitura de index inexistente na lista.

Figura 5



```
1 package br.unicamp.ic.inf335;
2
3 import java.util.ArrayList;
4
5 public class Brecho {
6
7     private static ArrayList<ProdutoBean> produtos = new ArrayList<ProdutoBean>();
8
9     public static void main(String[] args) {
10         ProdutoBean nProd = new ProdutoBean("CD00001", "Celular Galaxy S10", "128 Gb, Preto, com Carregador", 1250.0, "Poucos riscos, estado de
11         produtos.add(nProd);
12         nProd = new ProdutoBean("CD00002", "Prod 2 ...", "Bla Bla Bla", 1100.0, "Bla Bla Bla");
13         produtos.add(nProd);
14         nProd = new ProdutoBean("CD00003", "Prod 3 ...", "Bla Bla Bla", 120.0, "Bla Bla Bla");
15         produtos.add(nProd);
16         nProd = new ProdutoBean("CD00004", "Prod 4 ...", "Bla Bla Bla", 1300.0, "Bla Bla Bla");
17         produtos.add(nProd);
18         nProd = new ProdutoBean("CD00005", "Prod 5 ...", "Bla Bla Bla", 9400.0, "Bla Bla Bla");
19         produtos.add(nProd);
20         nProd = new ProdutoBean("CD00006", "Prod 6 ...", "Bla Bla Bla", 1500.0, "Bla Bla Bla");
21         produtos.add(nProd); //primeira exceção, linha 19
22
23         // Imprime produtos
24         for (int i=0; i<produtos.size(); i++) {
25             System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.g
26         }
27
28         // Ordena produtos
29         Collections.sort(produtos);
30
31         System.out.println("----- Produtos Ordenados -----");
32
33         // Imprime produtos ordenados
34         for (int i=1; i<produtos.size(); i++) {
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Console

```
<terminated> Brecho (Java Application) C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (4 de abr. de 2022 20:05:24 - 20:05:59) [pid: 21380]
Codigo = CD00001 Nome = Celular Galaxy S10 Valor = 1250.0
Codigo = CD00002 Nome = Prod 2 ... Valor = 1100.0
Codigo = CD00003 Nome = Prod 3 ... Valor = 120.0
Codigo = CD00004 Nome = Prod 4 ... Valor = 1300.0
Codigo = CD00005 Nome = Prod 5 ... Valor = 9400.0
Codigo = CD00006 Nome = Prod 6 ... Valor = 1500.0
----- Produtos Ordenados -----
Codigo = CD00002 Nome = Prod 2 ... Valor = 1100.0
Codigo = CD00003 Nome = Prod 3 ... Valor = 120.0
Codigo = CD00004 Nome = Prod 4 ... Valor = 1300.0
Codigo = CD00005 Nome = Prod 5 ... Valor = 9400.0
Codigo = CD00006 Nome = Prod 6 ... Valor = 1500.0
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 6 out of bounds for length 6
    at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)
    at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:70)
    at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:266)
    at java.base/java.util.Objects.checkIndex(Objects.java:359)
    at java.base/java.util.ArrayList.get(ArrayList.java:427)
    at br.unicamp.ic.inf335.Brecho.main(Brecho.java:37)
```

Um *markpoint* foi adicionado para que fosse checado o valor da variável “media”. Percebe-se que ela não está iterando os valores, apenas assumindo o valor do último produto listado no *array*.

Figura 6

The screenshot displays an IDE with a Java file named `Brecho.java`. The code defines a `Produto` class and a `main` method. It creates an array of `Produto` objects, sorts them by value, and calculates the average value. The `media` variable is calculated as the sum of all values divided by the number of products.

The Variables window on the right shows the state of the program. The `media` variable is highlighted, showing its value as 1500.0. The `value` variable is also visible, showing its value as 1500.0.

The Console window at the bottom shows the output of the program. It displays the sorted products and the calculated average value.

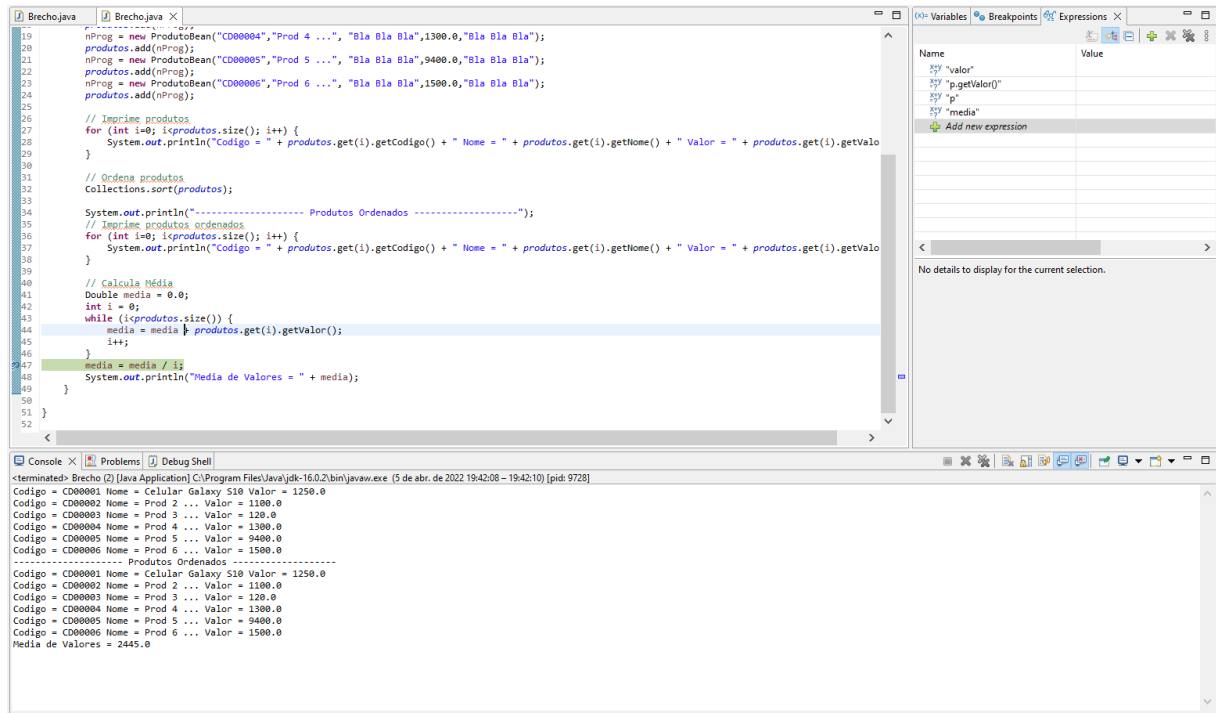
```
19 nProd = new ProdutoBean("CD00004", "Prod 4 ...", "Bla Bla Bla", 1300.0, "Bla Bla Bla");
20 produtos.add(nProd);
21 nProd = new ProdutoBean("CD00005", "Prod 5 ...", "Bla Bla Bla", 9400.0, "Bla Bla Bla");
22 produtos.add(nProd);
23 nProd = new ProdutoBean("CD00006", "Prod 6 ...", "Bla Bla Bla", 1500.0, "Bla Bla Bla");
24 produtos.add(nProd);
25
26 // Imprime produtos
27 for (int i=0; i<produtos.size(); i++) {
28     System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.get(i).getValor());
29 }
30
31 // Ordena produtos
32 Collections.sort(produtos);
33
34 System.out.println("----- Produtos Ordenados -----");
35
36 // Imprime produtos ordenados
37 for (int i=0; i<produtos.size(); i++) {
38     System.out.println("Codigo = " + produtos.get(i).getCodigo() + " Nome = " + produtos.get(i).getNome() + " Valor = " + produtos.get(i).getValor());
39 }
40
41 // Calcula Média
42 Double media = 0.0;
43 int i = 0;
44 while (i<produtos.size()) {
45     media = produtos.get(i).getValor();
46     i++;
47 }
48 media = media / i;
49 System.out.println("Média de Valores = " + media);
50 }
51 }
52 }
```

Console Output:

```
Brecho (2) [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (5 de abr. de 2022 19:38:01) [pid: 3468]
Codigo = CD00001 Nome = Celular Galaxy S10 Valor = 1250.0
Codigo = CD00002 Nome = Prod 2 ... Valor = 1100.0
Codigo = CD00003 Nome = Prod 3 ... Valor = 120.0
Codigo = CD00004 Nome = Prod 4 ... Valor = 1300.0
Codigo = CD00005 Nome = Prod 5 ... Valor = 9400.0
Codigo = CD00006 Nome = Prod 6 ... Valor = 1500.0
----- Produtos Ordenados -----
Codigo = CD00001 Nome = Celular Galaxy S10 Valor = 1250.0
Codigo = CD00002 Nome = Prod 2 ... Valor = 1100.0
Codigo = CD00003 Nome = Prod 3 ... Valor = 120.0
Codigo = CD00004 Nome = Prod 4 ... Valor = 1300.0
Codigo = CD00005 Nome = Prod 5 ... Valor = 9400.0
Codigo = CD00006 Nome = Prod 6 ... Valor = 1500.0
```

Após alterações, o código é executado até o fim sem apresentar erros, no entanto, o método *Collections.sort(array)* não está retornando a lista ordenada.

Figura 7



Para tanto, foram feitas correções nos operadores lógicos na função *compareTo(object)*. Com as alterações realizadas, o código não apresenta mais nenhuma exceção e suas saídas são coerentes com a proposta do programa.

Figura 8

