

## TRABALHO 1 – INF335 – Ambientes para Concepção de Software

Nome: Caroline Augusti – RG: 50.466.131-0

Nome: Paulo Mellin Gimenes - RG: 30.280.035-9

### 2.1. Depuração:

Esse código contém pelo menos 5 defeitos (bug). O objetivo é usar o depurador do Java (no Eclipse ou ferramenta equivalente) para identificar esses defeitos e responder às perguntas seguintes.

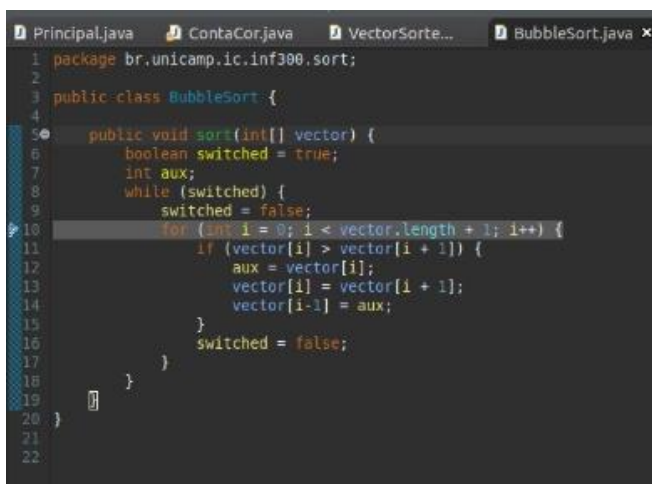
1 - Quais defeitos contém o código original? Considere ambas as classes. Responda à pergunta da forma “Bug #1: Linha XX, o código [aaa] deveria ser substituído por [bbb]”.

No arquivo Bubble Sort.java:

- Bug #1: Linha 10 - Código “for (int i = 0; i < vector.length + 1; i++) {” deveria ser substituído por “for (int i = 0; i < vector.length - 1; i++) {”.

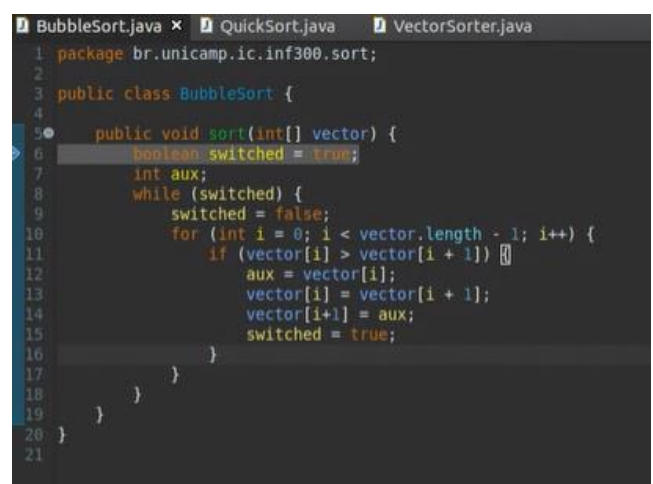
- Bug #2: Linha 14 – Código “vector[i - 1] = aux;” deveria ser substituído por “vector[i + 1] = aux;”.

- Bug#3: Linha 16 – Código “switched = false;” deveria ser substituído por “switched = true;”, e transferida para dentro do if (na linha 15).



```
1 package br.unicamp.ic.inf300.sort;
2
3 public class BubbleSort {
4
5     public void sort(int[] vector) {
6         boolean switched = true;
7         int aux;
8         while (switched) {
9             switched = false;
10            for (int i = 0; i < vector.length + 1; i++) {
11                if (vector[i] > vector[i + 1]) {
12                    aux = vector[i];
13                    vector[i] = vector[i + 1];
14                    vector[i - 1] = aux;
15                }
16                switched = false;
17            }
18        }
19    }
20 }
21
22
```

Antes



```
1 package br.unicamp.ic.inf300.sort;
2
3 public class BubbleSort {
4
5     public void sort(int[] vector) {
6         boolean switched = true;
7         int aux;
8         while (switched) {
9             switched = false;
10            for (int i = 0; i < vector.length - 1; i++) {
11                if (vector[i] > vector[i + 1]) {
12                    aux = vector[i];
13                    vector[i] = vector[i + 1];
14                    vector[i + 1] = aux;
15                    switched = true;
16                }
17            }
18        }
19    }
20 }
21
22
```

Depois

No arquivo VectorSorter.java:

- Bug #4: Linha 53 – Código “for (int i = 1; i < vector.length ; i++) {” deveria ser substituído por “for (int i = 0; i < vector.length ; i++) {”.

- Bug #5: Linha 64 – Código “int i = 1;” deveria ser substituído por “int i = 0;”.

```

52
53     for (int i = 1; i < vector.length; i++) {
54         vector[i] = (int) (Math.random()*100 + 1);
55     }
56
57     return vector;
58 }
59
60 public void print() {
61     System.out.print("[ ");
62     System.out.print(vector[0]);
63
64     int i = 1;
65     do {
66         i++;
67         System.out.print(", ");
68         System.out.print(vector[i]);
69     } while (i < vector.length - 1);
70
71     System.out.println(" ]");
72 }
73 }
74

```

Antes

```

34
35
36 public static int[] parseParameters() {
37     int[] numbers;
38     if (parameters.length > 0) {
39         numbers = new int[parameters.length];
40         for (int k = 0; k < parameters.length; k++) {
41             numbers[k] = Integer.parseInt(parameters[k]);
42         }
43     } else {
44         numbers = generateRandomVector(10);
45     }
46     return numbers;
47 }
48
49 private static int[] generateRandomVector(int size) {
50
51     int[] vector = new int[size];
52
53     for (int i = 0; i < vector.length; i++) {
54         vector[i] = vector.length - i;
55     }
56
57     return vector;
58 }
59
60 public void print() {
61     System.out.print("[ ");
62     System.out.print(vector[0]);
63
64     int i = 1;
65     do {
66         i++;
67         System.out.print(", ");
68         System.out.print(vector[i]);
69     } while (i < vector.length - 1);
70

```

Depois

2 - Considere o vetor [10, 9, 8, 7, 6, 5, 4, 3, 2, 1] como entrada da função sort:

2.1 - Qual é o conteúdo do vetor a terceira vez que o programa entrar no ciclo for?

Name	Value
no method return value	
this	BubbleSort (id=16)
vector	(id=17)
vector[0]	8
vector[1]	7
vector[2]	6
vector[3]	5
vector[4]	4
vector[5]	3
vector[6]	2
vector[7]	1
vector[8]	9
vector[9]	10
switched	true

A saída da segunda vez que o programa entra no ciclo for, correspondente a entrada do vetor na terceira vez que ele entra no ciclo do for. Sendo seu conteúdo, portanto, [8, 7, 6, 5, 4, 3, 2, 1, 9, 10].

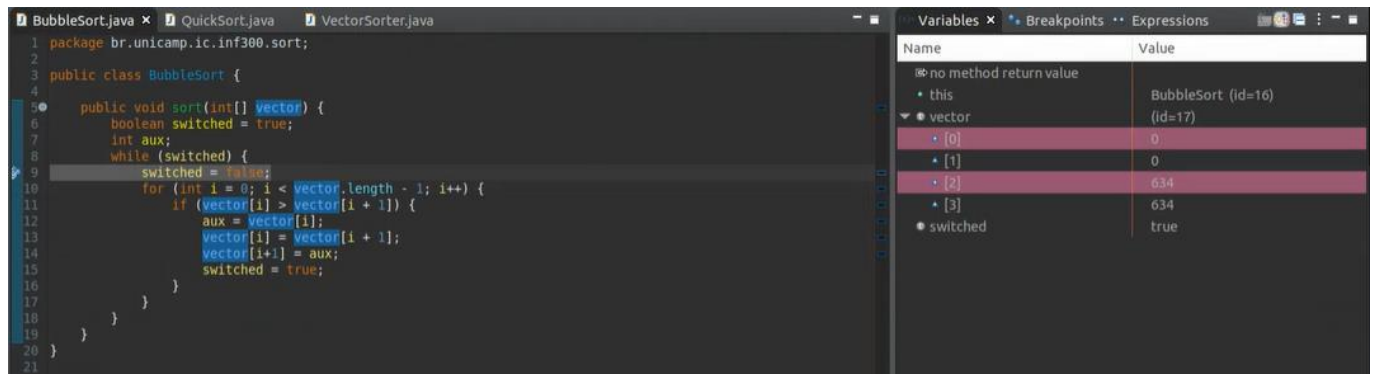
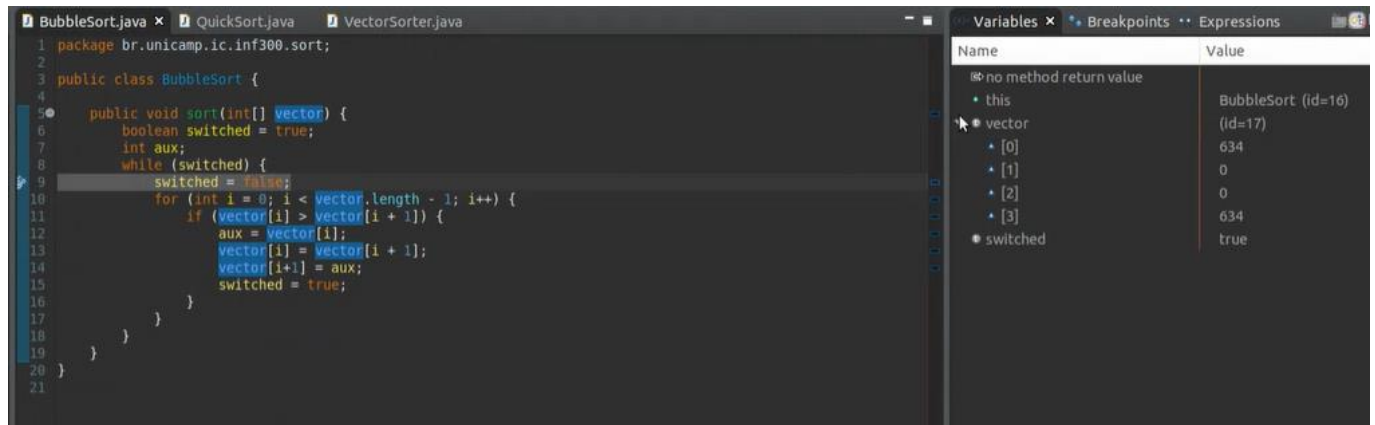
2.2 - Qual é o conteúdo do vetor a terceira vez que o programa sair do ciclo for?

Name	Value
no method return value	
this	BubbleSort (id=16)
vector	(id=17)
vector[0]	7
vector[1]	6
vector[2]	5
vector[3]	4
vector[4]	3
vector[5]	2
vector[6]	1
vector[7]	8
vector[8]	9
vector[9]	10
switched	true

A saída da terceira vez que o programa entra no ciclo for, correspondente a [7, 6, 5, 4, 3, 2, 1, 8, 9, 10].

**3 - Considere o vetor [634, 0, 0, 634] como entrada da função sort:**

**3.1 - Quantas trocas de posição são feitas até obter um vetor ordenado?**



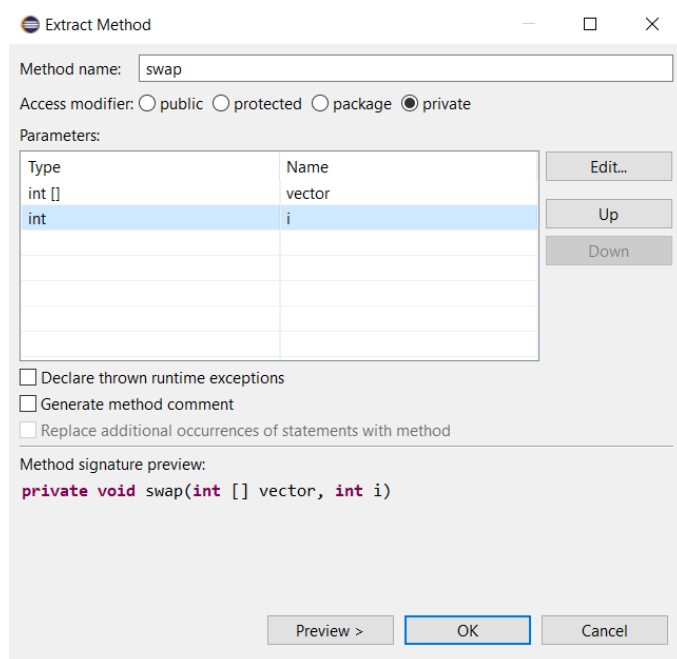
Foram necessárias duas trocas para que o vetor ficasse ordenado.

**Para cada pergunta anexe um screenshot mostrando o utilizzo do depurador configurado para tal pergunta.**

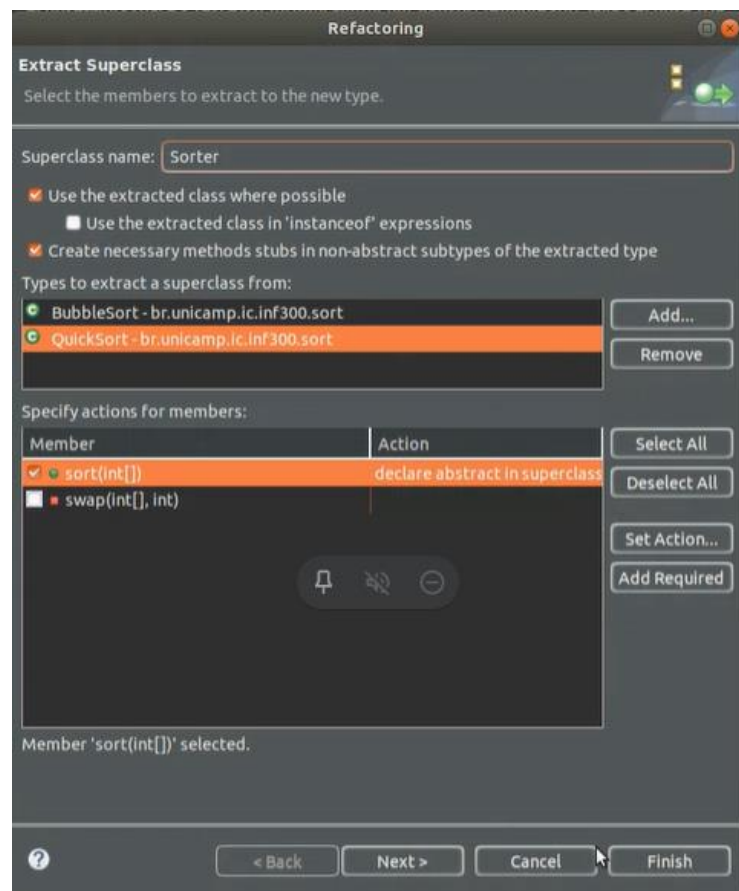
## 2.2 Refatoração

Use os recursos de refatoração do Eclipse para fazer as seguintes operações de refatoração no código:

- **Extract Method:** No método `sort` da classe `BubbleSort`, extrair o método que faz a troca de posição de elementos em um método privado `swap`.



**Extract Superclass:** Criar uma superclasse Sorter a comum entre BubbleSort e QuickSort, com o método abstrato sort.



```
1 package br.unicamp.ic.inf300.sort;
2
3 public abstract class Sorter {
4
5     public Sorter() {
6         super();
7     }
8
9     public abstract void sort(int[] vector);
10
11 }
```