

Inspeção de código

Arthur Gusmão de Matheus – 31949932

TIME: GP 1

Gabriel Zorzi - 32054531

Renato Nakano – 31851835

TicketMachine

1 – Erro

```
public class TicketMachine {  
    protected int valor;  
    protected int saldo;  
    protected int[] papelMoeda = {2, 5, 10, 20, 50, 100};  
  
    //Recebe precoDoBilhete, porém aqui esta sendo tratado como valor, alterar nome do parametro.  
  
    public TicketMachine(int valor) {  
        this.valor = valor;  
        this.saldo = 0;  
    }  
}
```

2 – Erro

```
    }  
    // Omissão falta atributos em getSaldo, getTroco.  
    public int getSaldo() {  
        return saldo;  
    }  
  
    public Iterator<Integer> getTroco() {  
        return null;  
    }  
}
```

3 – Erro

```

44         if (saldo < valor) {
45             throw new SaldoInsuficienteException();
46         }
47         String result = "*****\n";
48         result += "*** R$ " + saldo + ",00 ***\n";
49         result += "*****\n";
50         return result;
51     }
52 }

```

(Excesso)

Informações desnecessárias

Troco

1 – Erro

```

3  import java.util.Iterator;
4
5  /**
6   *
7   * @author Calebe de Paula Bianchini
8   */
9  class Troco {
10
11     protected PapelMoeda[] papeisMoeda;
12
13     public Troco(int valor) {
14         papeisMoeda = new PapelMoeda[6];
15         int count = 0;
16         while (valor % 100 != 0) {
17             count++;
18         }
19         papeisMoeda[5] = new PapelMoeda(100, count);
20         count = 0;
21         while (valor % 50 != 0) {
22             count++;
23         }
24         papeisMoeda[4] = new PapelMoeda(50, count);

```

(Comissão)

O atributo "papeisMoeda" não está definido no diagrama de classe.

2 – Erro

```

8    */
9    class Troco {
10
11        protected PapelMoeda[] papeisMoeda;
12
13        public Troco(int valor) {
14            papeisMoeda = new PapelMoeda[6];
15            int count = 0;
16            while (valor % 100 != 0) {
17                count++;
18            }
19            papeisMoeda[5] = new PapelMoeda(100, count);
20            count = 0;
21            while (valor % 50 != 0) {
22                count++;
23            }
24            papeisMoeda[4] = new PapelMoeda(50, count);
25            count = 0;

```

(Comissão)

A lógica está sendo executada no construtor.

3 – Erro

```

66        return false;
67    }
68
69    @Override
70    public PapelMoeda next() {
71        PapelMoeda ret = null;
72        for (int i = 6; i >= 0 && ret != null; i++) {
73            if (troco.papeisMoeda[i] != null) {
74                ret = troco.papeisMoeda[i];
75                troco.papeisMoeda[i] = null;
76            }
77        }
78        return ret;
79    }
80
81    @Override
82    public void remove() {
83        next();
84    }

```

(Computação)

A variável ret esta nula, ou seja, ela nunca entrara no loop abaixo

4 – Erro

```
66         return false;
67     }
68
69     @Override
70     public PapelMoeda next() {
71         PapelMoeda ret = null;
72         for (int i = 6; i >= 0 && ret != null; i++) {
73             if (troco.papeisMoeda[i] != null) {
74                 ret = troco.papeisMoeda[i];
75                 troco.papeisMoeda[i] = null;
76             }
77         }
78         return ret;
79     }
80
81     @Override
82     public void remove() {
83         next();
```

(Excesso)

O laço nunca será executado.

PapelMoeda

1 – Erro

```
14         this.quantidade = quantidade;
15     }
16
17     public int getValor() {
18         return valor;
19     }
20
21     public int getQuantidade() {
22         return quantidade;
23     }
24 }
```

(Comissão)

Falta dos métodos set dos atributos valor e quantidade.

2 – Erro

```
24 lines (19 sloc) | 443 Bytes

1  package br.calebe.ticketmachine.core;
2
3  /**
4   *
5   * @author Calebe de Paula Bianchini
6   */
7  public class PapelMoeda {
8
9      protected int valor;
10     protected int quantidade;
11
12     public PapelMoeda(int valor, int quantidade) {
13         this.valor = valor;
14         this.quantidade = quantidade;
15     }
16
17     public int getValor() {
18         return valor;
19     }
20
21     public int getQuantidade() {
22         return quantidade;
23     }
24 }
```

(Dados)

Falta da mensagem de exceção