

**LPOO**  
**1º semestre de 2020**  
**Lista 01**  
**20/03/2020**

**Nome:** \_\_\_\_\_  
**RA:** \_\_\_\_\_

Professores: Elio Lavisi, Wellington Felix,  
Natanael, Vitor Oliveira,  
Vinicius Pereira

---

Esta lista contém 5 páginas e 6 questões.

Para exibir algo na saída padrão você pode usar o termo `sout()`.

---

1. Sobre o conceito de *Constructor*, responda Verdadeiro ou Falso para as seguintes afirmações, justificando a sua resposta
  - I) Um constructor pode ter qualquer nome?
  - II) Um constructor pode retornar um objeto da classe em que ele está definido?
  - III) Um constructor pode ser declarado fora da classe?
  - IV) Um constructor tem um tipo de retorno?
2. Considere a seguintes classes chamadas `Aluno`, `AlunoTeste01` e `AlunoTeste02`

---

```
1  class Aluno{
2      String nome;
3      double nota;
4
5      Aluno(String aNome, double nota){
6          this.nome = aNome;
7          nota = nota;
8      }
9
10     @Override
11     public String toString(){
12         String resultado = "";
13
14         resultado += "Nome: " + this.nome + "\n";
15         resultado += "Nota: " + this.nota + "\n";
16
17         return resultado;
18     }
19
20 }
```

---

---

```
1  class AlunoTeste01{
2
3      public static void main(String[] args){
4          Aluno aluno01 = new Aluno("Pedro", 2.0);
5          System.out.println(aluno01);
6      }
7
8  }
```

---

```
1  class AlunoTeste01{
2
3      public static void main(String[] args){
4
5          Aluno aluno01 = new Aluno("Pedro", 2.0);
6          System.out.println(aluno01);
7
8          Aluno aluno02 = new Aluno();
9          System.out.println(aluno02);
10
11     }
12
13 }
```

---

Responda às seguintes questões:

- a) Qual será o resultado de executar `AlunoTeste01`?
- b) Existe algum erro lógico na classe `Aluno`? O que você mudaria nesta classe? Caso você mude alguma linha, qual será o resultado de `AlunoTeste01`?
- c) A classe `AlunoTeste02` tem algum erro de compilação? Qual?

## 3. Considerando a classe Conta

---

```
1  class Conta{
2
3      double saldo;
4
5      Conta(double aSaldo){
6          this.saldo = aSaldo;
7      }
8
9      void deposita(double valor){
10         this.saldo += valor;
11     }
12
13     double saca(double valor){
14         if(valor <= this.saldo){
15             this.saldo -= valor;
16             return valor;
17         }
18         else {
19             valor = this.saldo;
20             this.saldo = 0.0;
21             return valor;
22         }
23     }
24
25     @Override
26     public String toString(){
27         String resposta = "Saldo na conta: " + this.saldo;
28         return resposta;
29     }
30
31 }
```

---

Considere os blocos

- a. Da linha 2 a 4.
- b. Da linha 4 a 8.
- c. Da linha 8 a 30.

Estes blocos representam, respectivamente

- I) Atributos, constructor e métodos.
- II) Métodos, atributos e constructor.
- III) Constructor, atributos e métodos.
- IV) Constructor, métodos e atributos.

4. Considere a classe `ContaTeste` e a classe `Conta` da questão anterior

---

```
1  class ContaTeste{
2      public static void main(String[] args){
3
4          Conta conta = new Conta(100);
5          System.out.println(conta);
6
7          System.out.println("Tentando sacar " + 70);
8          System.out.println("Sacando " + conta.saca(70));
9          System.out.println(conta);
10
11         System.out.println("Depositando " + 20);
12         conta.deposita(20);
13         System.out.println(conta);
14
15         System.out.println("Tentando sacar " + 70);
16         System.out.println("Sacando " + conta.saca(70));
17         System.out.println(conta);
18     }
19 }
```

---

Escreva qual será a saída do método *main* de `ContaTeste`

5. Crie uma classe `ContaTelefone` com os seguintes requisitos:

- Atributos:
  - numero (*String*),
  - codigo (*int*),
  - minutosFalados (*double*).
- Todos atributos devem ser inicializados pelo construtor da classe
- Método com a assinatura `double getValorDaConta()` que retorna o valor da conta de telefone (em reais) seguindo a seguinte regra:
  - Se a quantidade de tempo falado (em minutos) for menor ou igual a 30, o valor da conta é 0.6 vezes o número de minutos.
  - Caso contrário, o valor da conta é 0.4 vezes o número de minutos.
- Sobrescreva `@Override` do método `public String toString()` que retorna uma *String* que contém o estado do objeto (cada atributo com seu valor)

Em seguida crie uma classe `ContaTelefoneTeste` com um *main* que cria duas contas de telefone, com todos os valores diferentes, e imprime cada uma usando `System.out.println(conta)`

6. Considerando a classe `ContaTelefone` da questão anterior, crie uma classe `ContaCelular`, filha de `ContaTelefone`, com os seguintes requisitos:

- `ContaCelular` herda de `ContaTelefone`
- Atributos:
  - operadora (*String*)
- Todos atributos devem ser inicializados pelo construtor da classe, sendo que os atributos da classe pai devem ser inicializados usando o construtor da classe pai (`super`). (Lembrando que o construtor da classe pai deve estar na primeira linha do construtor da classe filho.)
- Sobrescreva o método `double getValorDaConta()` onde
  - Se a quantidade de tempo falado for menor ou igual a 20, o valor da conta é 0.9 vezes o número de minutos.
  - Caso contrário, o valor da conta é 1.8 vezes o número de minutos.
  - Se o nome da operadora é “TC” acrescente 40 ao valor da conta. **Dica:** Não compare duas *Strings* com o operador `==`, mas sim usando o método `equals` da *String*
- Um `@Override` do método `public String toString()` que retorna uma *String* que contém o estado do objeto (cada atributo com seu valor)

Em seguida crie uma classe `ContaCelularTeste` com um *main* que cria duas contas de celular, com todos os valores diferentes, e imprime cada uma usando `System.out.println(conta)`