

**Prova 3**

Professor: Gustavo Henrique Borges Martins

Aluno: \_\_\_\_\_ Matrícula: \_\_\_\_\_

**Instruções para a prova:**

- Preencha o cabeçalho e todas as folhas desta prova com seu nome e sua matrícula.
- Leia atentamente a todas as questões antes de resolvê-las.
- As questões desta prova foram planejadas para serem resolvidas em linguagem Java.
- Não deixe de responder nenhuma questão.
- Deixe comentários sobre as questões, eles podem ser considerados para a resolução da questão.

1. (10 pontos) Implemente uma classe abstrata **Alma** que contenha:

- um atributo privado do tipo inteiro,
- um construtor padrão que inicialize o atributo privado com o valor 0,
- um método que some 1 no atributo privado,
- um método abstrato e que sobrecarregue o método anterior.

Implemente uma classe filha **Corpo** que não possa ser herdada e que herde as características da classe **Alma** e que contenha:

- um construtor padrão que inicialize o atributo com o valor -1,
- um método que retorne o valor do atributo.

O atributo não pode ser manipulado diretamente fora da classe **Alma** nem da filha **Corpo**, portanto, não faça métodos *Sets* para o atributo privado (Se não fosse o caso, não haveria necessidade do atributo ser privado.).

2. (10 pontos) Analise o seguinte trecho de código:

```
1 public class Main{
2     public static void main (String[] args){
3         final int tam = 3;
4         Mae Objs[] = new Mae[tam];
5         for (int i = 0; i < tam; i++){
6             Objs[i] = new Filha(i);
7         }
8         for (int i = 0; i < tam; i++){
9             for (int j = i; j < tam; j++){
10                 Objs[j].s();
11             }
12         }
13         for (int i = 0; i < tam; i++){
14             Objs[i].print();
15         }
16     }
17 }
18 public abstract class Mae{
19     private int s;
20     public Mae(int a){
21         s = a % 2;
22     }
23     public abstract void print();
24     public void print(String palavra){
25         System.out.println("A sentença:" + palavra + " contém " + palavra.length()
26         + " caracteres.");
27     }
28     protected boolean par(){
29         if ((s%2) == 0){
30             return true;
31         } else {
32             return false;
33         }
34     }
35     protected void s(){
36         s++;
37     }
38     protected int s(String palavra){
39         if (s > 0){
40             return palavra.length() % s;
41         } else {
42             return 0;
43         }
44     }
45 }
46 public class Filha extends Mae{
47     public Filha(int n){
48         super(n);
49         String arroz = "Arroz";
50         super.s(arroz);
51         s(arroz);
52     }
53     public void print(){
```

```

53         if (par()){
54             System.out.println("Quer dizer que temos um par.");
55         } else {
56             System.out.println("Quer dizer que temos um ímpar.");
57         }
58         s();
59     }
60     public void print(String palavra){
61         if (palavra.length() == 0){
62             System.out.println("A sentença:'" + palavra + "' contém um número par de
63             caracteres.");
64         } else {
65             System.out.println("A sentença:'" + palavra + "' contém um número ímpar
66             de caracteres.");
67         }
68         s();
69     }

```

Responda:

- Qual a função da palavra reservada **abstract** antes da classe **Mae**?
- Qual a função da palavra reservada **abstract** antes do método **print()**?
- Existe algum método sobrecarregado na classe **Mae**? Se sim, quais?
- Existe algum método sobrecarregado na classe **Filha**? Se sim, quais?
- Existe polimorfismo em algum método das classes **Mae** e **Filha**? Se sim, quais?
- Existe algum método abstrato na classe **Mae** ou na classe **Filha**? Se sim, quais?
- O que é impresso na tela ao longo de toda a execução do método **main(String[])**?
- Qual o valor de **Objs[0].s** ao final do método **main(String[])**?
- Qual o valor de **Objs[1].s** ao final do método **main(String[])**?
- Qual o valor de **Objs[2].s** ao final do método **main(String[])**?

---

Questões	1	2	Total
Total de pontos	10	10	20
Pontos obtidos			