

Prova 4

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) Escreva uma interface Menu.Escreva uma interface **Arquivo**, filha da interface **Menu**, que contenha:

- um método **Novo**,
- um método **Abrir**,
- um método **Salvar**.

Escreva uma interface **Editar**, filha da interface **Menu**, que contenha:

- um método **Cortar**,
- um método **Copiar**,
- um método **Colar**.

Implemente uma classe abstrata **Aplicação** que:

- contenha um método abstrato **Run**,
- seja classe da interface **Arquivo**.

Implemente uma classe **Leitor** que seja filha da classe **Aplicação** e seja filha da interface **Editar**.

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

```
1 import java.lang.Math;
2 public class Numeros {
3     public static void main (String[] args){
4         final int tam = 5;
5         Real VR[] = new Real[tam];
6         VR[0] = new Real(10);
7         VR[1] = new Real(8);
8         VR[2] = new Real(6);
9         VR[3] = new Real(10);
10        VR[4] = new Real(5);
11        Complexo VC[] = new Complexo [10];
12        VC[0] = new Complexo(5,10);
13        VC[1] = new Complexo(8,8);
14        VC[2] = new Complexo(7,3);
15        VC[3] = new Complexo(7,4);
16        VC[4] = new Complexo(1,5);
17        Ordena.bolha(VR);
18        Ordena.contagem();
19    }
20 }
21 public interface Base {
22     public void print();
23 }
24 public interface OperacoesBasicas extends Base {
25     public OperacoesBasicas Adicao (OperacoesBasicas num);
26     public OperacoesBasicas MultiplicacaoPorEscalar (double num);
27 }
28 public interface OperacoesAvancadas extends OperacoesBasicas {
29     public OperacoesAvancadas Multiplicacao (OperacoesAvancadas num);
30 }
31 public interface Comparavel extends Base {
32     public int comparacao (Comparavel valor);
33 }
34 public class Ordena {
35     private static long trocas, comparacoes;
36     public static void bolha (Comparavel[] vetor) {
37         trocas = 0;
38         comparacoes = 0;
39         for (int i = 0; i < vetor.length; i++) {
40             for (int j = i+1; j < vetor.length; j++) {
41                 comparacoes++;
42                 if (vetor[i].comparacao(vetor[j]) == -1) {
43                     trocas++;
44                     Comparavel temp = vetor[i];
45                     vetor[i] = vetor[j];
46                     vetor[j] = temp;
47                 }
48             }
49         }
50     }
51     public static void contagem () {
52         System.out.println ("Última ordenação: Trocas: " + trocas + " Comparações: "
53             + comparacoes);
```

```

53     }
54 }
55 public class Real implements OperacoesAvancadas, Comparavel {
56     public double r;
57     public Real (double a) {
58         r = a;
59     }
60     public void print(){
61         System.out.println("Número real: " + r);
62     }
63     public OperacoesBasicas Adicao (OperacoesBasicas num) {
64         return new Real(r + ((Real)num).r);
65     }
66     public OperacoesBasicas MultiplicacaoPorEscalar (double a) {
67         return new Real(a*r);
68     }
69     public OperacoesAvancadas Multiplicacao (OperacoesAvancadas num) {
70         return new Real(r * ((Real) num).r);
71     }
72     public int comparacao (Comparavel valor) {
73         Real comparado = (Real) valor;
74         if (r == comparado.r)
75             return 0;
76         else if (r < comparado.r)
77             return 1;
78         else
79             return -1;
80     }
81 }
82 public class Complexo implements OperacoesAvancadas {
83     public double r,i;
84     public Complexo (double a, double b) {
85         r = a;
86         i = b;
87     }
88     public void print (){
89         System.out.println("Parte real: " + r + " parte imaginária: " + i);
90     }
91     public OperacoesBasicas Adicao (OperacoesBasicas num) {
92         Complexo C = (Complexo) num;
93         return new Complexo(r + C.r, i + C.i);
94     }
95     public OperacoesBasicas MultiplicacaoPorEscalar (double a) {
96         return new Complexo(a*r, a*i);
97     }
98     public OperacoesAvancadas Multiplicacao (OperacoesAvancadas num) {
99         Complexo C = (Complexo) num;
100         return new Complexo(r * C.r - i * C.i, r * C.i + r * C.i);
101     }
102     public double Modulo(){
103         return Math.sqrt(r*r+i*i);
104     }
105 }
106

```

Responda:

- (a) Qual a função da palavra reservada **interface** antes de **Base**?
- (b) Qual a função da palavra reservada **implements** depois da classe **Real**?
- (c) Desenhe o diagrama de classes do código acima.
- (d) O que é impresso na tela ao longo de toda a execução do método **main(String[])**?
- (e) Qual o valor do conteúdo de **VR[]** e de **VC[]** ao final do método **main(String[])**?
- (f) Como você faria para ordenar o conteúdo do vetor **VC[]**? Descreva as operações necessárias e escreva o código.

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