

```

1  /* Teste Prático 1, sessão 1
2   * 8-nov-2019, Duração 1:30
3   *
4   */
5  import java.util.Scanner;
6
7  public class TP1A {
8      static Scanner teclado = new Scanner(System.in);
9
10     public static void main(String[] args) {
11
12         double xI, xF, deltaX; // intervalo de x, incremento de x
13         int npMax; // nº pontos máximo
14         int npValidos = 0; // nº pontos válidos
15         double[] funcao1; // array com pontos de uma função
16         int escolha;
17         do {
18             escolha = (int)lerNumero("1- ler pontos da Funcao do teclado; 2 - Funcao f():");
19         } while ((escolha != 1) && (escolha != 2));
20         do {
21             xI = lerNumero("Valor inicial de x: ");
22             xF = lerNumero("Valor final de x: ");
23         } while (xI >= xF);
24         npMax = (int)lerNumero("Número de pontos da função :");
25         if (escolha == 2) {
26             funcao1 = f(xI, xF, npMax);
27             npValidos = npMax;
28         } else {
29             funcao1 = new double[npMax];
30             npValidos = lerFuncao(funcao1);
31         }
32         deltaX = (xF - xI)/npValidos;
33
34         int[] raizes = zeros(funcao1, npValidos);
35
36         for (int r = 1; r <= raizes[0]; r++)
37             System.out.printf("Raiz %d x = %f\n", r, raizes[r] * deltaX + xI);
38
39         System.out.printf("Máximo para x = %.5f\n",
40             xI + xMaximo(funcao1, npValidos) * deltaX);
41
42         printFuncao(funcao1, npValidos, xI, deltaX);
43     }
44
45     /* 1) Função para ler um número decimal do teclado.
46      * Parâmetro: string com mensagem a mostrar
47      * Retorna: valor lido
48      */
49     static double lerNumero(String mensagem) {
50         System.out.print(mensagem);
51         double n = teclado.nextDouble();

```

```

52         return n;
53     }
54     /* 2) Função para ler do teclado vários pontos de uma funcao e guardá-los
num array.
55     *     Deve parar quando encher o array ou quando aparecerem dois 0
seguidos (0 0), que não contam para a sequência.
56     *     Parâmetros:   funcao - array com os pontos da função.
57     *     Retorna:      o nº de pontos lidos.
58     */
59     public static int lerFuncao(double[] funcao) {
60         System.out.printf("Introduza %d pontos de uma função ou até 0 0:%n",
funcao.length);
61         int y0 = -1, i = 0;
62         double y = -1;
63         do {
64             y0 = (int)y;
65             y = teclado.nextDouble();
66             funcao[i++] = y;
67         } while (!(y0 == 0 && (int)y == 0) && i < funcao.length);
68         if (y0 == 0 && (int)y == 0) i-=2; //2 últimos 0 não contam
69         return i;
70     }
71     /* 3) Função f que gera um array com com os pontos da função seno (x) no
intervalo dado
72     *     Parâmetros:   xI, xF (valores inicial e final do intervalo), np (nº
de pontos)
73     *     Retorna:      array com os np pontos da função
74     */
75     static double[] f(double xI, double xF, int np) {
76         double[] a = new double[np];
77         double dx = (xF - xI)/np;
78         for (int i = 0; i < a.length; i++) {
79             a[i] = Math.sin(xI + dx * i);
80         }
81         return a;
82     }
83     /* 4) Função xMaximo calcula posição no array que tem o valor maximo
84     *     Parâmetros:   funcao - array com os pontos; npv - nº de pontos válidos
85     *     Retorna:      posição do array onde está o máximo
86     */
87     static int xMaximo(double[] funcao, int npv) {
88         int xMax = 0;
89         for (int i = 0; i < npv; i++) {
90             if (funcao[i] > funcao[xMax]) xMax = i;
91         }
92         return xMax;
93     }
94
95     /* 5) Função que dado um array com vários pontos de uma função calcula os
zeros da função.
96     *     Os zeros são os pontos onde a função muda de sinal.
97     *     Parâmetros:   funcao - array com os pontos da funcao; npv - nº de
pontos válidos

```

```

98      *   Retorna:      array inteiro com a posição dos zeros.
99      *               0 primeiro elemento do array[0] tem o nº de zeros encontrados.
100     */
101     static int[] zeros(double[] funcao, int npv) {
102         int nraizes = 0;
103         int[] raiz = new int[npv];      // raiz[0] = nº de raizes
104
105         for (int i = 1; i < npv; i++) {
106             if (funcao[i-1] * funcao[i] < 0 || funcao[i] == 0){
107                 nraizes = nraizes + 1;
108                 raiz[nraizes] = i;
109             }
110         }
111         raiz[0] = nraizes;
112         return raiz;
113     }
114     /* 6) Função para listar no ecrã os pontos da funcao escolhida de acordo com o formato indicado na fig.
115     *   Parâmetros:   funcao - array com os pontos da funcao; npv - nº de pontos válidos;
116     *               xI - valor de x inicial; dx - incremento de x
117     *
118     */
119     public static void printFuncao(double[] funcao, int npv, double xI, double dx) {
120         for (int x = 0; x < npv ; x++) {
121             System.out.printf("Pos=%3d x= %8.5f y= %8.5f%n",x, xI + x*dx,funcao[x]);
122         }
123     }
124 }
125

```