

NºMec:

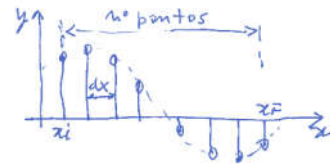
Nome:

Informações:

- O ficheiro criado deve ter um nome do tipo TP1A**12345**.java, em que **12345** deve ser substituído pelo número mecanográfico do aluno. Deve gravar o ficheiro no ambiente de trabalho (Desktop).
- Coloquem o nome e número mecanográfico no início do ficheiro como comentário.
- A prova é realizada sem consulta. Podem consultar a documentação de Java disponível no computador através do comando `view-javadoc "classe"` (exemplo: `view-javadoc Math`).

Problema:

Completar o programa dado que determina as raízes de uma função num intervalo $[X_i, X_f]$, dados os valores inicial e final do intervalo e o número de pontos que devem ser calculados. Pode-se escolher entre usar uma função dada ($f()$) ou ler os pontos da função do teclado. As raízes são os valores de x para os quais a função tem valor zero, ou seja, quando o valor da função (y) num determinado ponto muda de sinal relativamente ao ponto anterior (y_0). O incremento da variável x depende do intervalo e do número de pontos usados para o cálculo: $dx = (X_f - X_i) / \text{nº pontos}$. Também é calculado o valor de x que corresponde ao máximo da função e listada no ecrã uma tabela com a função de acordo com o formato indicado abaixo (resultados).



Funções a implementar (sugere-se que as vá implementando e testando por passos):

1) `double lerNumero(String mensagem) {}`

(2 valores)

Função que lê um número decimal do teclado e o retorna. O parâmetro é a mensagem que deve ser apresentada ao utilizador, indicativa da leitura.

2) `int lerFuncao(double[] funcao) {}`

(4 valores)

Função para ler do teclado vários pontos de uma funcao e guardá-los num array. Deve parar quando encher o array ou quando aparecerem dois 0 seguidos (considere só a parte inteira para esta comparação), que não contam para a sequência. Parâmetros: funcao - array com os pontos da função. Retorna: o nº de pontos lidos.

3) `double[] f(double xI, double xF, int np) {}`

(4 valores)

Função que calcula vários pontos (np) da função $f(x) = \text{seno}(x)$ no intervalo xI, xF . Devolve um array com os pontos.

4) `int xMaximo(double[] funcao, int npv) {}`

(4 valores)

Função `xMaximo` calcula posição no array que tem o valor máximo. Parâmetros: funcao - array com os pontos; npv - nº de pontos válidos. Retorna: posição do array onde está o máximo.

5) `int[] zeros(double[] funcao, int npv) {}`

(4 valores)

Função que determina os índices do array que correspondem às raízes da função (zeros ou pontos em que há mudança de sinal). Tem como argumentos um array com os pontos da função (funcao) e o número de pontos da amostra (npv). A função deve devolver um array com a posição das raízes encontradas, tendo o primeiro elemento do array `[0]` o nº de raízes válidas.

6) void **printFuncao**(double[] funcao,int npv,double xI,double dx) {}

(2 valores)

Função para listar no ecrã os pontos da função escolhida de acordo com o formato indicado abaixo (resultados). Parâmetros: funcao - array com os pontos da função; npv - nº de pontos válidos; xI - valor de x inicial; dx - incremento de x

Penalizações (mesmo que os resultados sejam corretos)

1. O código não está corretamente alinhado por blocos.
2. O programa não tem comentários adequados.
3. Erros de compilação.
4. Funções necessárias não são criadas ou não são chamadas corretamente.
5. O programa tem complexidade desnecessária.

Resultados e formatos (as entradas do utilizador estão apresentadas a negrito):

1- ler pontos da Funcao do teclado; 2 - Funcao f():**1**

Valor inicial de x: **-1**

Valor final de x: **9**

Número de pontos da função :**50**

Introduza 50 pontos de uma função ou até 0 0:

1.1 2.2 3.3 -1.5 -2 -1.2 0 0

Raiz 1 x = 4.000000

Máximo para x = 2.33333

Pos= 0 x= -1.00000 y= 1.10000

Pos= 1 x= 0.66667 y= 2.20000

Pos= 2 x= 2.33333 y= 3.30000

Pos= 3 x= 4.00000 y= -1.50000

Pos= 4 x= 5.66667 y= -2.00000

Pos= 5 x= 7.33333 y= -1.20000

1- ler pontos da Funcao do teclado; 2 - Funcao f():**2**

Valor inicial de x: **1**

Valor final de x: **7**

Número de pontos da função :**100**

Raiz 1 x = 3.160000

Raiz 2 x = 6.340000

Máximo para x = 1.60000

Pos= 0 x= 1.00000 y= 0.84147

Pos= 1 x= 1.06000 y= 0.87236

...

Pos= 9 x= 1.54000 y= 0.99953

Pos= 10 x= 1.60000 y= 0.99957

Pos= 11 x= 1.66000 y= 0.99602

...

Pos= 35 x= 3.10000 y= 0.04158

Pos= 36 x= 3.16000 y= -0.01841

Pos= 37 x= 3.22000 y= -0.07833

...

Pos= 87 x= 6.22000 y= -0.06314

Pos= 88 x= 6.28000 y= -0.00319

Pos= 89 x= 6.34000 y= 0.05678

Pos= 90 x= 6.40000 y= 0.11655

...

Pos= 98 x= 6.88000 y= 0.56201

Pos= 99 x= 6.94000 y= 0.61060

```
import java.util.Scanner;
public class TP1 {
    static Scanner teclado = new Scanner(System.in)
    public static void main(String[] args) {

        double xI, xF, deltaX;           // intervalo de x, incremento de x
        int npMax ;                       // n° pontos máximo
        int npValidos = 0;                // n° pontos válidos
        double[] funcao1;                 // array com pontos de uma função
        int escolha;
        do {
            escolha = (int)lerNumero("1- ler pontos da Funcao do teclado; 2 - Funcao f():");
        } while((escolha != 1) && (escolha != 2));
        do {
            xI = lerNumero("Valor inicial de x: ");
            xF = lerNumero("Valor final de x: ");
        } while (xI >= xF);
        npMax = (int)lerNumero("Número de pontos da função :");
        if (escolha == 2) {
            funcao1 = f(xI, xF, npMax);
            npValidos = npMax;
        } else {
            funcao1 = new double[npMax];
            npValidos = lerFuncao(funcao1);
        }
        deltaX = (xF - xI)/npValidos;
        int[] raizes = zeros(funcao1, npValidos);
        for (int r = 1; r <= raizes[0]; r++)
            System.out.printf("Raiz %d x = %f\n", r, raizes[r] * deltaX + xI);
        System.out.printf("Máximo para x = %.5f\n",
            xI + xMaximo(funcao1, npValidos) * deltaX);
        printFuncao(funcao1, npValidos, xI, deltaX);
    }

    // 1) Função para ler um número decimal do teclado.
    public static double lerNumero(String mensagem) {

    }

    // 2) Função para ler do teclado vários pontos de uma funcao e guardá-los num array.
    public static int lerFuncao(double[] funcao) {

    }

    // 3) Função f que gera um array double com com os pontos da função seno (x) num
    intervalo
    static double[] f(double xI, double xF, int np) {

    }

    // 4) Função xMaximo calcula posição no array que tem o valor maximo
    static int  xMaximo(double[] funcao, int npv) {

    }

    // 5) Função que dado um array com vários pontos de uma função calcula os zeros da
    função.
    static int[]  zeros(double[] funcao, int npv) {

    }

    // 6) Função para listar no ecrã os pontos da funcao escolhida
    public static void printFuncao(double[] funcao, int npv, double xI, double dx){

    }

}
```