

Programação Orientada a Objetos

FUNÇÕES

2021 / Ago

Prof. Dr. Clarimundo Machado Moraes Júnior

```
package eqgrau2semfuncao;

import java.util.Scanner;

public class EqGrau2SemFuncao {

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        Scanner coeficiente = new Scanner(System.in);

        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
        a = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente B:");
        b = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente C:");
        c = coeficiente.nextDouble();
        d = Math.pow(b, 2) - 4*a*c;
        if(d<0)
            System.out.println("Não existem raízes reais!");
        else{
            x1=(-b+Math.sqrt(d))/(2*a);
            x2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
        }
    } //fim do main()
} //fim da classe
```

```
package eqgrau2semfuncao;
```

```
import java.util.Scanner;
```

```
public class EqGrau2SemFuncao {
```

```
    public static void main(String[] args) {  
        double a, b, c, d, x1, x2;
```

```
        Scanner coeficiente = new Scanner(System.in);
```

Leitura dos coeficientes

```
        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
```

```
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
```

```
        a = coeficiente.nextDouble();
```

```
        System.out.println("Digite o coeficiente B:");
```

```
        b = coeficiente.nextDouble();
```

```
        System.out.println("Digite o coeficiente C:");
```

```
        c = coeficiente.nextDouble();
```

```
        d = Math.pow(b, 2) - 4*a*c;
```

```
        if(d<0)
```

```
            System.out.println("Não existem raízes reais!");
```

```
        else{
```

```
            x1=(-b+Math.sqrt(d))/(2*a);
```

```
            x2=(-b-Math.sqrt(d))/(2*a);
```

```
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
```

```
        }
```

```
    } //fim do main()
```

```
} //fim da classe
```

```

package eqgrau2semfuncao;

import java.util.Scanner;

public class EqGrau2SemFuncao {

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        Scanner coeficiente = new Scanner(System.in);

        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
        a = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente B:");
        b = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente C:");
        c = coeficiente.nextDouble();
        d = Math.pow(b, 2) - 4*a*c;
        if (d < 0)
            System.out.println("Não existem raízes reais!");
        else{
            x1 = (-b + Math.sqrt(d)) / (2*a);
            x2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
        }
    } //fim do main()
} //fim da classe

```

Cálculo do delta

```

package eqgrau2semfuncao;

import java.util.Scanner;

public class EqGrau2SemFuncao {

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        Scanner coeficiente = new Scanner(System.in);

        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
        a = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente B:");
        b = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente C:");
        c = coeficiente.nextDouble();
        d = Math.pow(b, 2) - 4*a*c;
        if(d<0)
            System.out.println("Não existem raízes reais!");
        else{
            x1=(-b+Math.sqrt(d))/(2*a);
            x2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
        }
    } //fim do main()
} //fim da classe

```

Cálculo das raízes

```

package eqgrau2semfuncao;

import java.util.Scanner;

public class EqGrau2SemFuncao {

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        Scanner coeficiente = new Scanner(System.in);

        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
        a = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente B:");
        b = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente C:");
        c = coeficiente.nextDouble();
        d = Math.pow(b, 2) - 4*a*c;
        if(d<0)
            System.out.println("Não existem raízes reais!");
        else{
            x1=(-b+Math.sqrt(d))/(2*a);
            x2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
        }
    } //fim do main()
} //fim da classe

```

Exibição das raízes

```

package eqgrau2semfuncao;

import java.util.Scanner;

public class EqGrau2SemFuncao {

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        Scanner coeficiente = new Scanner(System.in);

        System.out.println("Digite os coeficientes da equação com A diferente de zero!");
        System.out.println("Digite o coeficiente A (deve ser diferente de zero):");
        a = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente B:");
        b = coeficiente.nextDouble();
        System.out.println("Digite o coeficiente C:");
        c = coeficiente.nextDouble();
        d = Math.pow(b, 2) - 4*a*c;
        if(d<0)
            System.out.println("Não existem raízes reais!");
        else{
            x1=(-b+Math.sqrt(d))/(2*a);
            x2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("As raízes da equação são: X1= "+x1+"      X2= "+x2);
        }
    } //fim do main()
} //fim da classe

```

Exibição das raízes

Função

OBJETIVOS

- Modularização
- Separação de Funcionalidades
- Facilidade de manutenção
- Economia de código

ESTRUTURA DE UMA FUNÇÃO

```
[visibilidade] [tipo de retorno] <nome da função> ( [parâmetros] ) {  
    <escopo ou corpo da função>  
}
```

EXEMPLO

```
public void exhibe( ){  
    System.out.println("\nBoa noite turma");  
}
```



```

public class EqGrau2ComFuncao {

    public static double le(int coef) {...5 linhas }

    public static double delta(double a, double b, double c) {...3 linhas }

    public static double calculaX1(double a, double b, double d) {...3 linhas }

    public static double calculaX2(double a, double b, double d) {...3 linhas }

    public static void exhibe(double x1, double x2) {...3 linhas }

    public static void main(String[] args) {
        double a, b, c, d, x1, x2;

        System.out.println("Digite os 3 coeficientes da equação com A diferente de zero!");
        a = le(1);
        b = le(2);
        c = le(3);
        d = delta(a,b,c);
        if(d<0)
            System.out.println("Não existem raízes reais!");
        else{
            x1= calculaX1(a,b,d);
            x2= calculaX2(a,b,d);
            exhibe(x1,x2);
        }
    } //fim do main()
} //fim da classe

```

Função

VISIBILIDADE DA FUNÇÃO

- É possível limitar o acesso às funções
- Public, Private e Protected
- **Public** → todos podem acessar
- **static** → a própria classe executa a função sem uso de objetos (instância)

TIPO DE RETORNO DA FUNÇÃO

- **void** → não retorna valor
- **int** → retorna um valor inteiro
- **char** → retorna um valor char
- **string** → retorna um valor String
- **float** → retorna um valor float
- **double** → retorna um valor double

Função

Quando há retorno, via de regra, a função retorna apenas 1 valor por vez !

PARÂMETROS DA FUNÇÃO

- Função pode ter ou não ter parâmetros
- A função precisa receber dados externos ?

Função

EXEMPLO1 - uso de função que não retorna valor e não tem parâmetros

- Faça um programa que leia um número inteiro e exibe sua paridade
- **SUGESTÃO:** função que sinaliza uma violação de restrição

EXEMPLO2 - uso de função que não retorna valor e tem parâmetro

- Faça um programa que leia um número inteiro e exibe sua paridade
- **SUGESTÃO :** função que exibe a paridade

EXEMPLO3 - uso de função que retorna valor e não tem parâmetros

- Faça um programa que leia um número inteiro e exibe sua paridade
- **SUGESTÃO :** função que lê o número inteiro

EXEMPLO4 - uso de função que retorna valor e tem parâmetros

- Faça um programa que leia um número inteiro e exibe sua paridade
- **SUGESTÃO :** função que verifica a paridade