

Interpretador Hall

Resolvendo uma equação do 2º grau

Apresentaremos aqui três formas de se calcular as raízes de uma equação do 2º grau.

- Em um único programa com o algoritmo na função principal
- Em um programa com funções para se calcular o delta e as raízes
- Em um programa com o controle na apenas na função principal e funções para se calcular o delta e as raízes

Veja abaixo a versão 1 do programa. O programa que contém o algoritmo em uma única função, na função principal denominada algoritmo.

Equação do 2º Grau – versão 1

```
algoritmo ()
{
    // equacao do 2 grau

    real a;           // coeficiente a
    real b;           // coeficiente b
    real c;           // coeficiente c
    real d;           // delta
    real x1;          // primeira raiz
    real x2;          // segunda raiz
    caracter resp;    // deseja continuar ?

    repita
    {
        escreva ("-----");
        escreva ("equacao da forma ax2 + bx + c = 0");
        escreva ("-----");

        leia ("coeficiente a: ", a);
        leia ("coeficiente b: ", b);
        leia ("coeficiente c: ", c);

        // consideracoes sobre o coeficiente a

        se (a <> 0)
        {
            // calculo do delta

            d := b*b - (4*a*c);

            // consideracoes sobre delta

            se ( d > 0 )
            {
```

Equação do 2º Grau - versão 1

```
        escreva ("delta = ", d, " => raizes distintas.");
        x1 := (-b + raiz(d)) / (2*a);
        x2 := (-b - raiz(d)) / (2*a);
        escreva ("x1 = ", x1);
        escreva ("x2 = ", x2);
    }
senao
{
    se ( d == 0 )
    {
        escreva ("delta = ", d, " => raizes iguais.");
        x1 := (-b + raiz(d)) / (2*a);
        escreva ("x1 = ", x1);
        escreva ("x2 = ", x1);
    }
senao
{
    escreva ("delta = ", d, " => raizes imaginarias.");
    escreva ("as raizes nao sao reais");
}
}
}
senao
{
    escreva ("o coeficiente a nao pode ser nulo.");
}

leia ("deseja continuar (s/n): ", resp);

escreva();          // apenas para avancar uma linha

} ate (resp=='n');
```

Veja abaixo a versão 2 do programa

Nesta versão usamos 4 funções:

- **Algoritmo**
- **CalcularDelta(real a, real b, real c)**
- **CalcularR1(real a, real b, real d)**
- **CalcularR2(real a, real b, real d)**

A função **CalcularDelta** recebe como parâmetros todos os coeficientes da equação (três valores). As funções **CalcularR1** e **CalcularR2** recebem como argumentos os coeficientes **a** e **b** e o valor de **delta** que foi calculado anteriormente.

Note que os parâmetros que as funções esperam receber são todos do tipo **real**.

Equação do 2º Grau – versão 2

```
algoritmo ()
{
    // equacao do 2 grau

    real a;           // coeficiente a
    real b;           // coeficiente b
    real c;           // coeficiente c
    real d;           // delta
    real x1;          // primeira raiz
    real x2;          // segunda raiz
    caracter resp;    // deseja continuar ?

    repita
    {
        escreva ("-----");
        escreva ("equacao da forma ax2 + bx + c = 0");
        escreva ("-----");

        leia ("coeficiente a: ", a);
        leia ("coeficiente b: ", b);
        leia ("coeficiente c: ", c);

        // consideracoes sobre o coeficiente a

        se (a <> 0)
        {
            // calculo do delta

            d := CalcularDelta(a,b,c);

            // consideracoes sobre delta

            se ( d > 0 )
```

Equação do 2º Grau - versão 2

```

    {
        escreva ("delta = ", d, " => raizes distintas.");
        x1 := CalcularR1(a,b,d);
        x2 := CalcularR2(a,b,d);
        escreva ("x1 = ", x1);
        escreva ("x2 = ", x2);
    }
senao
{
    se ( d == 0 )
    {
        escreva ("delta = ", d, " => raizes iguais.");
        d := -1*d;
        x1 := CalcularR1(a,b,d);
        escreva ("x1 = ", x1);
        escreva ("x2 = ", x1);
    }
senao
{
    escreva ("delta = ", d, " => raizes imaginarias.");
    escreva ("as raizes nao sao reais");
}
}
}
senao
{
    escreva ("o coeficiente a nao pode ser nulo.");
}

leia ("deseja continuar (s/n): ",resp);

escreva ();          // apenas para avancar uma linha

} ate (resp=='n');
}
//-----
CalcularDelta(real a, real b, real c)
{
    // calculo do delta

    real d;

    d := b*b - (4*a*c);
    retorne d;
}
//-----
CalcularR1(real a, real b, real d)
{
    real x1;

    x1 := (-b + raiz(d)) / (2*a);
    retorne x1;
}
```

Equação do 2º Grau – versão 2

```
}  
//-----  
CalcularR2(real a, real b, real d)  
{  
    real x2;  
  
    x2 := (-b - raiz(d)) / (2*a);  
    retorne x2;  
}  
//-----
```

Veja abaixo a versão 3 do programa

Nesta versão usamos 5 funções:

- **Algoritmo**
- **ResolverEq2Grau(real a, real b, real c)**
- **CalcularDelta(real a, real b, real c)**
- **CalcularR1(real a, real b, real d)**
- **CalcularR2(real a, real b, real d)**

A função **CalcularDelta** recebe como parâmetros todos os coeficientes da equação (três valores). As funções **CalcularR1** e **CalcularR2** recebem como argumentos os coeficientes **a** e **b** e o valor de **delta** que foi calculado anteriormente.

Note que os parâmetros que as funções esperam receber são todos do tipo **real**.

Observe que o trabalho da função principal (algoritmo) foi consideravelmente reduzido, o que facilitará a manutenção do código. Veja que essa função agora, controla apenas a chamada às funções que irão realizar o “trabalho pesado” de calcular os valores. Observe também com houve uma redução na quantidade de variáveis que são declaradas nessa função.

Equação do 2º Grau – versão 3

```
algoritmo ()
{
    // equacao do 2 grau

    real a;           // coeficiente a
    real b;           // coeficiente b
    real c;           // coeficiente c
    caracter resp;    // deseja continuar ?

    repita
    {
        escreva ("-----");
        escreva ("equacao da forma ax2 + bx + c = 0");
        escreva ("-----");

        leia ("coeficiente a: ", a);
        leia ("coeficiente b: ", b);
        leia ("coeficiente c: ", c);

        ResolverEq2Grau(a,b,c);

        leia ("deseja continuar (s/n): ",resp);

        escreva ();           // apenas para avançar uma linha
    }
}
```

Equação do 2º Grau - versão 3

```
    } ate (resp=='n');
}
//-----
ResolverEq2Grau(real a, real b, real c)
{
    real d;      // delta
    real x1;     // primeira raiz
    real x2;     // segunda raiz

    // consideracoes sobre o coeficiente a

    se (a <> 0)
    {
        // calculo do delta

        d := CalcularDelta(a,b,c);

        // consideracoes sobre delta

        se ( d > 0 )
        {
            escreva ("delta = ", d, " => raizes distintas.");
            x1 := CalcularR1(a,b,d);
            x2 := CalcularR2(a,b,d);
            escreva ("x1 = ", x1);
            escreva ("x2 = ", x2);
        }
        senao
        {
            se ( d == 0 )
            {
                escreva ("delta = ", d, " => raizes iguais.");
                d := -1*d;
                x1 := CalcularR1(a,b,d);
                escreva ("x1 = ", x1);
                escreva ("x2 = ", x1);
            }
            senao
            {
                escreva ("delta = ", d, " => raizes imaginarias.");
                escreva ("as raizes nao sao reais");
            }
        }
    }
    senao
    {
        escreva ("o coeficiente a nao pode ser nulo.");
    }
}
//-----
CalcularDelta(real a, real b, real c)
{
```

Equação do 2º Grau – versão 3

```
// calculo do delta
real d;

d := b*b - (4*a*c);
retorne d;
}
//-----
CalcularR1(real a, real b, real d)
{
  real x1;

  x1 := (-b + raiz(d)) / (2*a);
  retorne x1;
}
//-----
CalcularR2(real a, real b, real d)
{
  real x2;

  x2 := (-b - raiz(d)) / (2*a);
  retorne x2;
}
//-----
```

Comentários