

```

package calculonumerico;

import java.io.IOException;
import java.text.DecimalFormat;
import java.util.Random;

/**
 *
 * @author ramon
 */
public class Cholesky {
    int n;
    double soma;
    double[][] a, l;
    double[] y, b, x;
    Random aleatorio;

    Cholesky() throws IOException{
        System.out.print("Metodo de Cholesky by Ramon");
        System.out.println();

        n = 3;

        a = new double[n][n];
        l = new double[n][n];

        //a = pegaMatriz();
        colocarMatrizNaMao();

        //primeiro passo:
        l[0][0] = Math.sqrt(a[0][0]);

        //segundo passo
        for(int j = 1; j < n; j++){
            l[j][0] = a[j][0] / l[0][0];
        }

        //terceiro passo
        for(int i = 1; i < n-1 ; i++){
            soma = 0;
            //quarto passo
            for(int k = 0; k < i; k++){
                soma = soma + (l[i][k] * l[i][k]);
            }
            l[i][i] = Math.sqrt(a[i][i] - soma);

            //quinto passo
            for(int j = i + 1; j < n; j++){
                soma = 0;
                for(int k=0; k<i; k++){

```

```

        soma = soma + (l[j][k]* l[i][k]);
    }
    l[j][i] = (a[j][i] - soma) / l[i][i];
}

}
//sexto passo
soma = 0;
for(int k =0; k < n-1; k++){
    soma = soma + (l[n-1][k] * l[n-1][k]);
}
l[n-1][n-1] = Math.sqrt(a[n-1][n-1] - soma);

//triangular inferior

y = new double[n];
b = pegaArrayb();

y[0] = b[0]/l[0][0];
for(int k =1; k<n ; k++){
    soma = 0;
    for(int i = 0; i< k; i++){
        soma = soma + (l[k][i] * y[i]);
    }
    y[k]= (b[k] - soma)/l[k][k];
}

//triangular superior
//pega a matriz inferior como se fosse a superior
x = new double[n];
x[n-1] = y[n-1]/l[n-1][n-1];
for(int k=n-2; k >= 0; k--){
    soma = 0;
    for(int i = n-1; i >= 0 ; i--){
        soma = soma + (l[i][k] * x[i]);
    }
    x[k] = (y[k] - soma)/l[k][k];
}

// //transpor matriz
// for(int k=0; k<n;k++){
//     for(int j = k+1; j < n; j++){
//         l[k][j] = l[j][k];
//     }
// }
//
// int k;
// for(int i = 0; i< n-1; i++){
//     k = (n-1) - i;
//     soma = 0;
//     for(int j = k+1; j < n; j++){
//         soma = soma + (l[j][k] * x[j]);

```

```

//      }
//      x[i]= (y[i] - soma)/l[i][i];
//      }

//metodos de impressao(saida)
System.out.println("Matriz proposta:");
imprimiMatriz(a);
System.out.println("Matriz fatorada:");
imprimiMatriz(l);
System.out.println("Array b:");
imprimiArray(b);
System.out.println("Array y:");
imprimiArray(y);
System.out.println("Array solucao X:");
imprimiArray(x);
}

public double[][] pegaMatriz(){
//vai gerar uma matriz aleatoria
a = new double[n][n] ;
aleatorio = new Random();
int nAleatorio = 1 + aleatorio.nextInt(42);
/*
Random aleatorio = new Random();
for(int j= 0; j< n; j++){
    for(int kMath.pow(2, i) = 0; k< n; k++){
        a[j][k] = 1+aleatorio.nextInt(10);
    }
}
*/
int j,cont1,k,cont2;
for(j= 0, cont1 =0; j< n; j++, cont1++){
    for(k = 0, cont2=1; k< n; k++, cont2++){
        a[j][k] = Math.pow( nAleatorio, cont1+cont2) ;
    }
}

return a;
}
public void imprimiMatriz(double[][] matriz){
    for(int j= 0; j< n; j++){
        for(int k = 0; k< n; k++){

            System.out.print(matriz[j][k]);
            System.out.print(" - ");
        }
        System.out.println();
    }
}

```

```
}  
}
```

```
private double[] pegaArrayb() {  
    b = new double[n];  
    /*  
    for(int i = 0; i < n ; i++){  
        //b[i] = Math.pow(2, i);  
  
        b[i]= 1 + aleatorio.nextInt(n);  
    }  
    */  
    b[0] = 2;  
    b[1] = 1;  
    b[2] = 5;  
  
    return b;  
}
```

```
private void imprimiArray(double[] y) {  
    // imprime vetor com arredondamento  
    DecimalFormat decimal = new DecimalFormat( "0.00" );  
    for(int i = 0; i < n ; i++){  
        System.out.println(decimal.format(y[i]));  
  
    }  
}
```

```
private void colocarMatrizNaMao() throws IOException {  
    /*  
    a = new double[n][n];  
    for(int j= 0; j < n; j++){  
        for(int k = 0; k < n; k++){  
            System.out.print("Digite o elemento: A" );  
            System.out.println();  
            a[j][k] = System.in.read();  
        }  
  
    }  
    */
```

```
    a[0][0] = 1;  
    a[0][1] = 1;  
    a[0][2] = 0;  
    //a[0][3] = ;  
    a[1][0] = 1;  
    a[1][1] = 2;  
    a[1][2] = -1;  
    //a[1][3] = ;  
    a[2][0] = 0;  
    a[2][1] = -1;  
    a[2][2] = 3;
```

```
//a[2][3] = ;  
//a[3][0] = ;  
//a[3][1] = ;  
//a[3][2] = ;  
//a[3][3] = ;
```

```
}  
}
```