

## Como Aplicar *Array* multidimensional no NetBeans

TESTE - NetBeans IDE 8.1

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projeto: TESTE

Arquivos: teste, TESTE.java, Bibliotecas, TESTE2

Navegador: TESTE, main(String[] args)

Código-Fonte

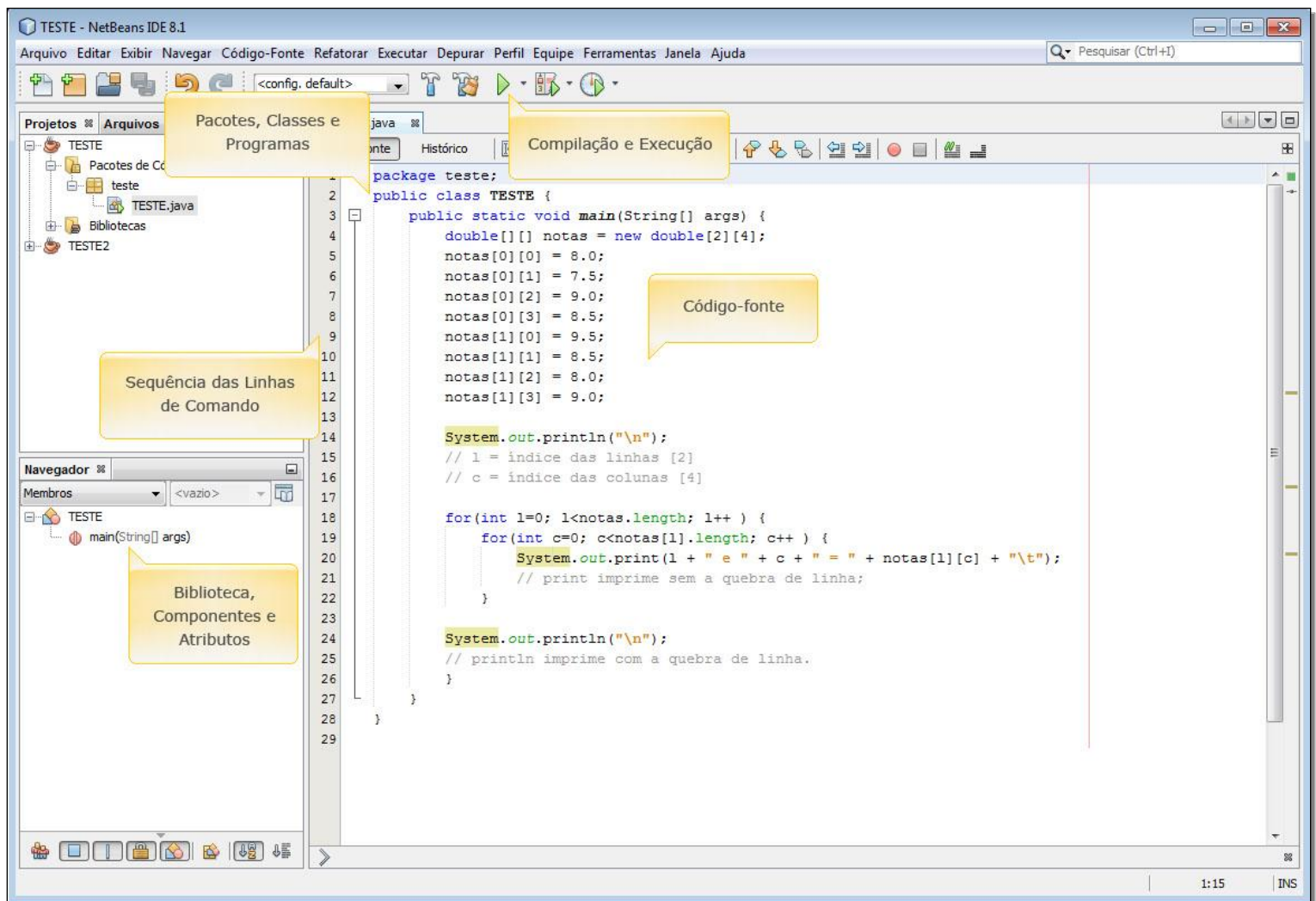
```

1 package teste;
2 public class TESTE {
3     public static void main(String[] args) {
4         double[][] notas = new double[2][4];
5         notas[0][0] = 8.0;
6         notas[0][1] = 7.5;
7         notas[0][2] = 9.0;
8         notas[0][3] = 8.5;
9         notas[1][0] = 9.5;
10        notas[1][1] = 8.5;
11        notas[1][2] = 8.0;
12        notas[1][3] = 9.0;
13
14        System.out.println("\n");
15        // l = índice das linhas [2]
16        // c = índice das colunas [4]
17
18        for(int l=0; l<notas.length; l++) {
19            for(int c=0; c<notas[l].length; c++) {
20                System.out.print(l + " e " + c + " = " + notas[l][c] + "\t");
21                // print imprime sem a quebra de linha;
22            }
23
24            System.out.println("\n");
25            // println imprime com a quebra de linha.
26        }
27    }
28 }
29

```

A seguir, podemos comparar o código de programação anterior desenvolvido em uma ferramenta própria de desenvolvimento: NetBeans.

1:15 | INS



TESTE - NetBeans IDE 8.1

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Pesquisar (Ctrl+I)

Projetos Arquivos Serviços

TESTE

Pacotes de Códigos-fonte

teste

TESTE.java

Bibliotecas

TESTE2

Navegador

Membros

<vazio>

TESTE

main(String[] args)

Código-Fonte Histórico

```

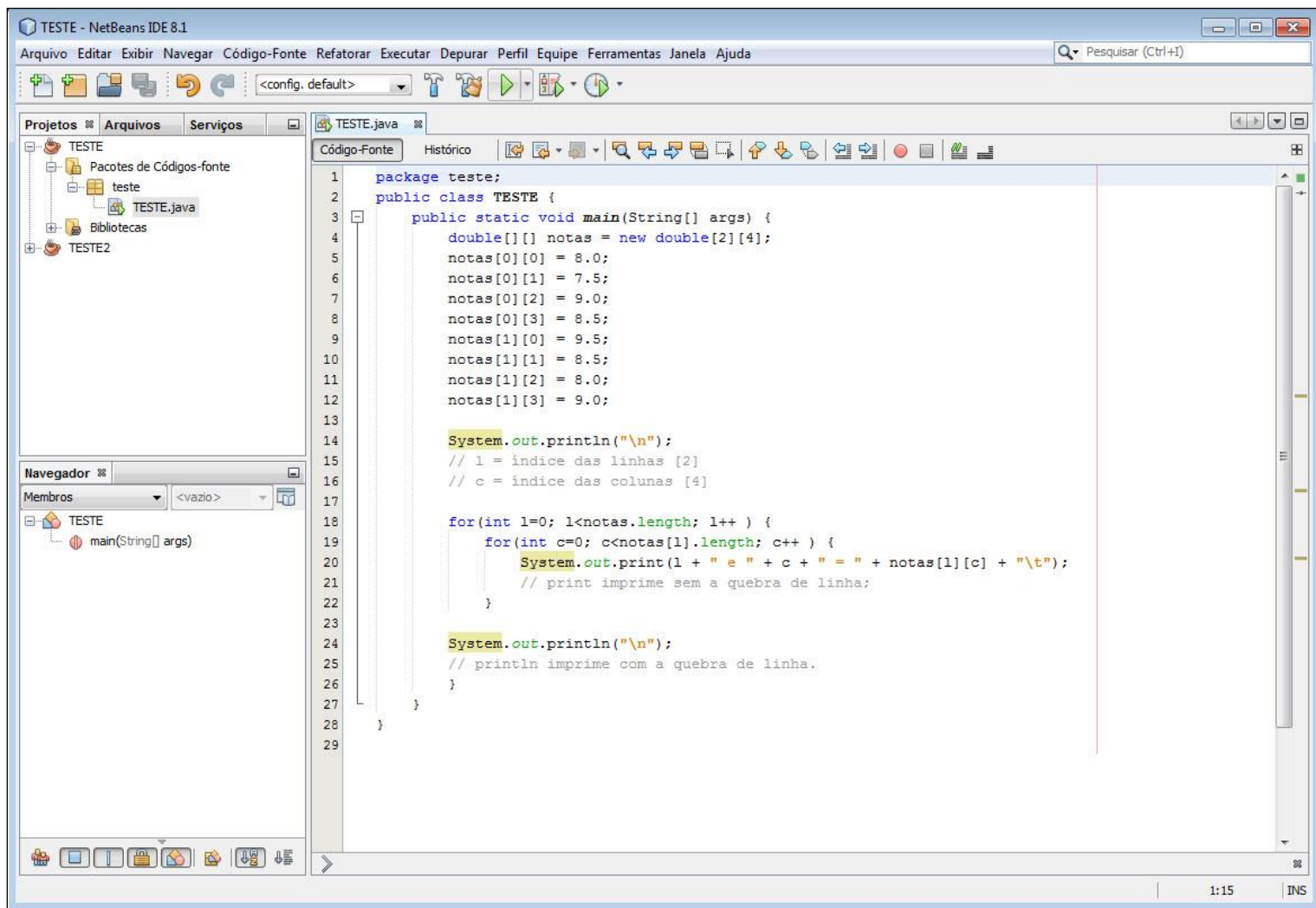
1 package teste;
2 public class TESTE {
3     public static void main(String[] args) {
4         double[][] notas = new double[2][4];
5         notas[0][0] = 8.0;
6         notas[0][1] = 7.5;
7         notas[0][2] = 9.0;
8         notas[0][3] = 8.5;
9         notas[1][0] = 9.5;
10        notas[1][1] = 8.5;
11        notas[1][2] = 8.0;
12        notas[1][3] = 9.0;
13
14        System.out.println("\n");
15        // l = índice das linhas [2]
16        // c = índice das colunas [4]
17
18        for(int l=0; l<notas.length; l++)
19            for(int c=0; c<notas[l].length; c++)
20                System.out.print(notas[l][c] + " ");
21        // print imprime sem quebra de linha
22    }
23
24    System.out.println("\n");
25    // println imprime com a quebra de linha.
26 }
27
28
29

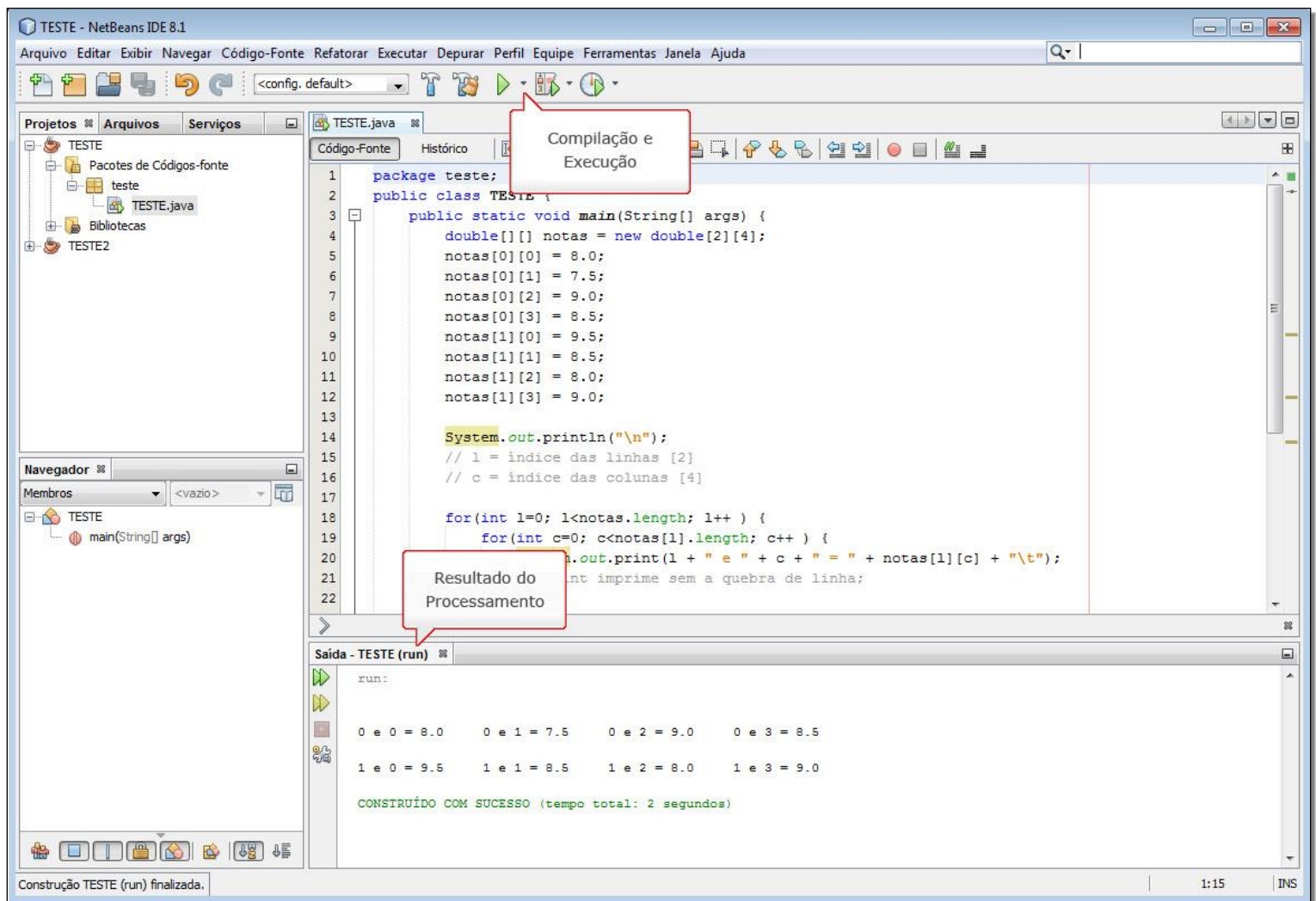
```

Observe que as linhas do código são as mesmas. No entanto, alguns fatores proporcionam um melhor acompanhamento do código, bem como um ambiente mais interativo e agradável para o profissional de desenvolvimento. Vejamos:

- Contribuição do ambiente com a indicação numérica e sequencial da cada linha
- Localização e estrutura do código em relação à classe e ao pacote
- Recursos de acesso
- Compilação
- Execução
- Resultado do processamento.

1:15 INS





TESTE - NetBeans IDE 8.1

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projetos Arquivos Serviços

TESTE

Pacotes de Códigos-fonte

teste

TESTE.java

Bibliotecas

TESTE2

Navegador

Membros

<vazio>

TESTE

main(String[] args)

Código-Fonte Histórico

TESTE.java

```

1 package teste;
2 public class TESTE {
3     public static void main(String[] args) {
4         double[][] notas = new double[2][4];
5         notas[0][0] = 8.0;
6         notas[0][1] = 7.5;
7         notas[0][2] = 9.0;
8         notas[0][3] = 8.5;
9         notas[1][0] = 9.5;
10        notas[1][1] = 8.5;
11        notas[1][2] = 8.0;
12        notas[1][3] = 9.0;
13
14        System.out.println("\n");
15        // l = índice das linhas [2]
16        // c = índice das colunas [4]
17
18        for(int l=0; l<notas.length; l++) {
19            for(int c=0; c<notas[l].length; c++) {
20                System.out.print(l + " e " + c + " = " + notas[l][c] + "\t");
21            }
22            System.out.println();
23        }
24    }
25 }

```

Compilação e Execução

Resultado do Processamento

Saída - TESTE (run)

```

run:
0 e 0 = 8.0    0 e 1 = 7.5    0 e 2 = 9.0    0 e 3 = 8.5
1 e 0 = 9.5    1 e 1 = 8.5    1 e 2 = 8.0    1 e 3 = 9.0

CONSTRUÍDO COM SUCESSO (tempo total: 2 segundos)

```

Construção TESTE (run) finalizada.

1:15 INS