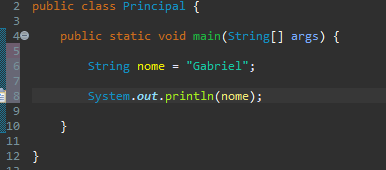
**String:**

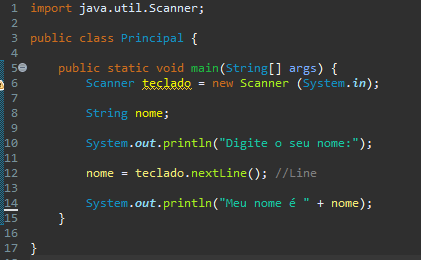
Usado como um tipo primitivo (mas não é), nos nossos 1º exemplos são usado para representar uma **cadeia de caractere**.



Para pegar o **nome ou algo escrito** usando o scanner eu uso o **.next ou .nextLine diferença** é: Gabriel de Jesus

O .next = Gabriel

O .nextline = Gabriel de Jesus

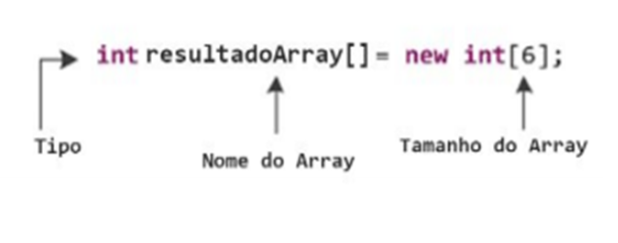


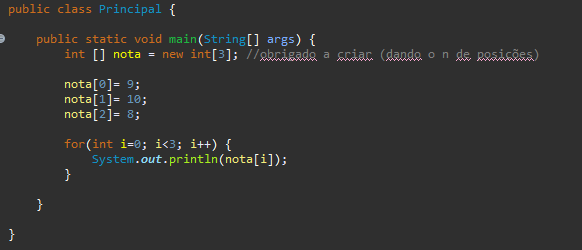
**Exercício: Pegar nome, matricula e calcular a media.**



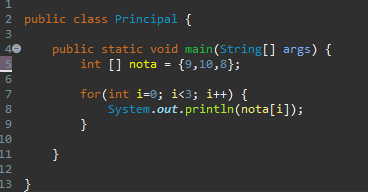
**Array:**

É tipo um armário, onde tem varias gavetas! No qual guarda variáveis.

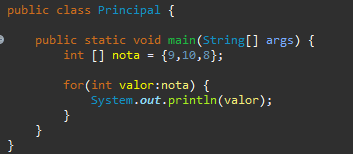




OU



E com vetores podemos fazer um **for diferente**!



Assim ele joga os valores da nota dentro do valor e mostra cada um.

**Literais de matriz:**

O literal nulo usado para representar a ausência de um objeto também pode ser usado para representar a ausência de uma matriz. Por exemplo:

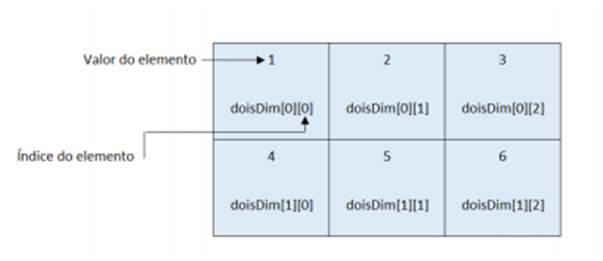
**String [] nome = null;**

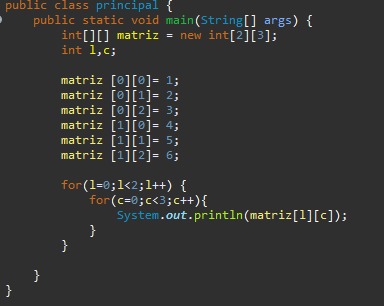
Além do literal nulo, Java também define uma sintaxe especial que permite especificar valores de matriz literalmente em seus programas. Essa sintaxe pode ser usada apenas ao declarar uma variável do tipo de matriz. Combina a criação do objeto de matriz com a inicialização dos elementos da matriz:

**String [] diasDaSemana = {“Domingo”, “Segunda-feira”, “Terça-feira”,“Quarta-feira”, “Quinta-feira”, “Sexta-feira”, “Sábado”};**

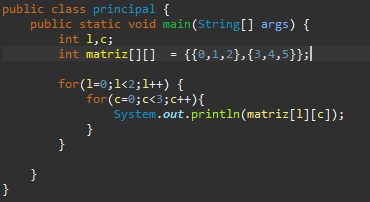
**Array Bidimensional:**

É um vetor de vetores! É um array que em vez de ter uma variável, tem outro array.



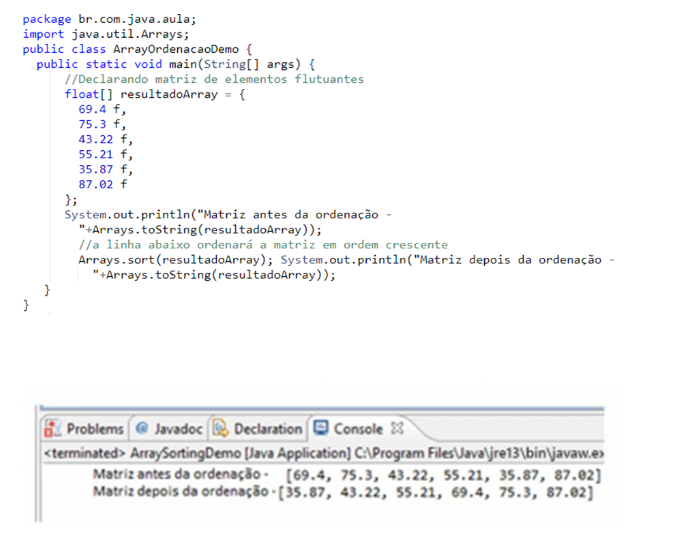


**OU**



**Classe auxiliar incorporada (java.util.Arrays) para manipulação de matrizes**

Java fornece classe auxiliar muito importante (java.util.Arrays) para manipulação de array. Essa classe possui muitos métodos utilitários, como classificação de matriz, impressão de valores de todos os elementos da matriz, pesquisa de um elemento, cópia de uma matriz em outra matriz, etc. Vamos ver um exemplo de programa para entender essa classe para melhor programação. No programa abaixo, a matriz flutuante foi declarada. Estamos imprimindo os elementos da matriz antes da classificação e após a classificação.



Semelhante ao “java.util.Arrays”, a classe **System** também possui uma funcionalidade de copiar dados de maneira eficiente de uma matriz para outra. Sintaxe como abaixo:

**public static void arraycopiaDados(Object src, int srcPos,Object dest, int destPos, int length)**

Os dois argumentos do objeto especificam a matriz da qual copiar e a matriz para a qual copiar. Os três argumentos int especificam a posição inicial na matriz de origem, a posição inicial na matriz de destino e o número de elementos da matriz a serem copiados.