Introdução a Array

Em lições anteriores, discutimos como declarar diferentes variáveis usando os tipos de dados primitivos.
Na declaração de variáveis, frequentemente utilizamos um identificador ou um nome e um tipo de dados. Para se utilizar uma variável, deve-se chamá-la pelo nome que a identifica.
Por exemplo, temos três variáveis do tipo
int
com diferentes identificadores para cada
variável:

```
int number1;
int number2;
int number3;
number1 = 1;
number2 = 2;
number3 = 3;
```

Como se vê, inicializar e utilizar variáveis pode tornar-se uma tarefa tediosa, especialmente se
elas forem utilizadas para o mesmo objetivo. Em Java, e em outras linguagens de
programação, pode-se utilizar uma variável para armazenar e manipular uma lista de dados
com maior eficiência. Este tipo de variável é chamado de array

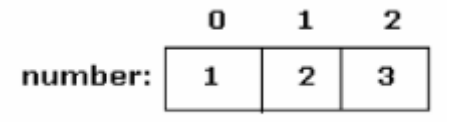

Figura 1: Exemplo de um array de inteiros

Um array armazena múltiplos itens de um mesmo tipo de dado em um bloco contínuo de
memória, dividindo-o em certa quantidade de posições.
Imagine um array como uma variável esticada – que tem um nome que a identifica e que pode conter mais de um valor para esta mesma variável

Declarando Array

Array precisa ser declarado como qualquer variável. Ao declarar um array, defina o tipo dedados deste seguido por colchetes [] e pelo nome que o identifica. Por exemplo:

```
int [] ages;
```

ou colocando os colchetes depois do identificador. Por exemplo:

```
int ages[];
```

Depois da declaração, precisamos criar o array e especificar seu tamanho. Este processo é

chamado de construção (a palavra, em orientação a objetos, para a criação de objetos)
Para se construir um objeto, precisamos utilizar um construtor
Por exemplo:

```
// declaração
int ages[];

// construindo
ages = new int[100];
```

ou, pode ser escrito como:

```
// declarar e construir
int ages[] = new int[100];
```

No exemplo, a declaração diz ao compilador Java que o identificador **ages** será usado como um nome de um array contendo inteiros, usado para criar, ou construir, um novo array contendo 100 elementos.

Em vez de utilizar uma nova linha de instrução para construir um array, também é possível automaticamente declarar, construir e adicionar um valor uma única vez.

Acessando um elemento do Array

Para acessar um elemento do array, ou parte de um array, utiliza-se um número inteiro chamado de índice.
Um índice é atribuído para cada membro de um array, permitindo ao programa e ao programador acessar os valores individualmente quando necessário. Os números dos índices são sempre inteiros.
Eles começam com zero e progridem sequencialmente por todas as posições até o fim do array. Lembre-se que os elementos dentro do array possuem índice de 0 a tamanhoDoArray-1.
Por exemplo, dado o array ages que declaramos anteriormente, temos:

```
// atribuir 10 ao primeiro elemento do array

ages[0] = 10;

// imprimir o último elemento do array

System.out.print(ages[99]);
```

Lembre-se que o array, uma vez declarado e construído, terá o valor de cada membro inicializado automaticamente. Conforme a seguinte tabela:

Tipo primitivo	Iniciado com
boolean	false
byte, short e int	0
char	'\u0000'
long	0L
float	0.0F
double	0.0

Tabela 1: Valor de inicialização automatica para os tipos primitivos

Entretanto, tipos de dados por referência, como as Strings, não serão inicializados caracteres em branco ou com uma string vazia "", serão inicializados com o valor null. Deste modo, o ideal é preencher os elementos do arrays de forma explícita antes de utilizá-los. A manipulação de objetos nulos pode causar a desagradável surpresa de uma exceção do tipo NullPointerException;

por exemplo, ao tentar executar algum método da classe String,
conforme o exemplo a seguir:

```
public class ArraySample {

   public static void main(String[] args){

      String [] nulls = new String[2];

      System.out.print(nulls[0]); // Linha correta, mostra null

      System.out.print(nulls[1].trim()); // Causa erro
   }
}
```

O código abaixo utiliza uma declaração for para mostrar todos os elementos de um array.

```
public class ArraySample {
   public static void main(String[] args){
      int[] ages = new int[100];
      for (int i = 0; i < 100; i++) {
         System.out.print(ages[i]);

                  }
         }
}
```

Tamanho de Array

Para se obter o número de elementos de um array, pode-se utilizar o atributo length
O atributo length de um array retorna seu tamanho, ou seja, a quantidade de elementos.

É utilizado como no código abaixo:

nomeArray.length

Por exemplo, dado o código anterior, podemos reescrevê-lo como:

```
public class ArraySample {
    public static void main (String[] args) {
        int[] ages = new int[100];
        for (int i = 0; i <
ages.length
; i++) {
            System.out.print(ages[i]);
        }
    }
}
```