Arrays

DigitalHouse>

Certified Tech Developer

The Ultimate Degree

Índice

1 | Arrays

Arrays

Arrays são **estruturas** de dados **estáticas** que permitem armazenar elementos do mesmo tipo de forma contínua.

Eles permitem o acesso aos seus elementos de forma aleatória por meio de um índice que começa em 0 (zero).

A coleção ArrayList tem o mesmo comportamento, portanto, seu nome.

	nomes
0	Juan
1	Mario
2	Carlos
3	Marcelo
4	Marcelo

Arrays

Em Java, um **array** é um **objeto**, e como tal, o operador **new** deve ser usado para criar uma instância, mas ao contrário das coleções, os arrays são de **comprimento fixo**, que deve ser definido na criação, sendo imutáveis.

Com os colchetes "[ ]" indica que é uma array.

```
String[] nomes = new String[5];
```

O tipo pode ser um tipo primitivo ou qualquer tipo de objeto.

Aqui, devemos definir dentro dos colchetes [ ] o tamanho da estrutura.

nomes	
0	null
1	null
2	null
3	null
4	null

Arrays

Definimos valores para uma **array** por meio de seu **índice**. Dado que é uma estrutura fixa, os elementos não podem ser removidos.

```
nomes[0] = "Juan";
nomes[1] = "Mario";
nomes[3] = "Marcelo";
```

	nomes
0	Juan
1	Mario
2	null
3	Marcelo
4	null

Tentar acessar um índice fora do intervalo, como por exemplo, nomes [10] levanta uma exceção. Veremos o que é uma exceção em breve.

Arrays

Podemos percorrer um array por meio de **for, while ou for each** e também usar a propriedade **length** que indica o tamanho do array.

```
for(int i = 0; i < nomes.length; i++)
   System.out.println(nomes[i]);
```

```
int i = 0;
while(i < nomes.length) {
   System.out.println(nomes[i]);
   i++;
}
```

```
for(String nome : nomes)
   System.out.println(nome);
```

	nomes
0	Juan
1	Mario
2	null
3	Marcelo
4	null

2 | Array vs. Coleções

Array vs. Coleções

	Array	Coleções
Estrutura	Estática	Dinâmica
Tipos primitivos	Usa tipos primitivos	Você deve usar as classes Integer, Float, Double
Longitude	nomes.**lenght**	nomes.**size()**
Obter um valor	nomes**[2]**	nomes.**get(2)**
Estabelecer um valor	nomes**[2] = "Carlos"**	nomes.**set(2,"Carlos")**
Incluir um elemento	Não é possível	nomes.**add("Juan")**

Array vs. Coleções

	Array	Coleções
Remover um elemento	Não é possível	nomes.**remove(“Juan”)**
Acesso a um elemento fora do limite estabelecido	Erro Exceção	Não traz erro
Ordenamento	Não pode	Pode

Devido aos benefícios, facilidade de manutenção e para evitar bugs, sempre usaremos coleções ao armazenar um conjunto de objetos.


DigitalHouse>
Coding School