Coleções

Coleções

Tratando múltiplos objetos de forma eficiente!

Coleções

Veremos neste assunto:

- O que são coleções
- ArrayList
- Exemplo

Coleções

O que são coleções ?
Estruturas que armazenam múltiplos objetos.
São parecidas com vetores ou arrays, mas vetores (ou arrays) são fixos.
Coleções **são dinâmicas.**

Coleções

Principais implementações:

- ArrayList (Lista dinâmica)
- HashSet (Conjunto de elementos únicos)
- HashMap (Chave-Valor)

Para uma lista completa, acesse a documentação do Java sobre collections disponível em: https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html

Coleções

Algumas vantagens em utilizar coleções

Por que usar coleções?

- Permitem armazenar **quantidades variáveis** de objetos.
- Oferecem **métodos prontos** para adicionar, remover e buscar elementos.
- Melhor organização e desempenho comparado a arrays fixos.
- Menor esforço de aprendizado e de implementação

ArrayList

Estudaremos a classe ArrayList

ArrayList é uma das implementações de coleções disponíveis em Java. É fácil de aprender por sua semelhança com vetores e arrays.

ArrayList

ArrayList não é array ...
Apenas parece com array

ArrayList

Como ArrayList parece Arrays, vamos começar por Arrays ....
E vocês já viram arrays (ou matrizes) e vetores em outras linguagens

Lembre-se que já estudamos anteriormente quando tratamos do assunto do parâmetro args, que é um array de Strings.

Arrays em Java

Array

A declaração de um vetor (ou array) pode ser feita de dois modos:

1) Sem inicialização
2) Com inicialização

Arrays em Java

1) Declarando sem inicialização

Sintaxe:

```
Tipo nomeDaVariavel[] = new Tipo[tamanho desejado];
```

Neste caso, precisará preencher cada posição do array

Exemplo

```
int  numerosPares[] = new int[6];
```

numeroPares é o nome do array

Ele foi inicializado com tamanho de 6 posições, iniciando em 0

Arrays em Java (cont.)

2) Com inicialização

Sintaxe

```
Tipo nomeDaVariavel[] = {valor1, valor2, valor3 }
```

{ valor1, valor2, valor3} é chamada de lista de inicializadores

Exemplos

```
int  numerosPares[] = {0,2,4,6,8,10}
double valores[]  = {1.201, 2.102, 3.193}
```

Arrays em Java (cont.)

Qualquer elemento de um vetor é acessado via índice inteiro. Este índice inicia em 0.

Exemplo

```
int numerosPares = {0,2,4,6,8,10};
int v = numerosPares[2];
```

No exemplo, v recebe o valor 4;

Arrays em Java (cont.)

O comprimento do vetor é dado pelo atributo **length**

```
int[]  numerosPares = {0,2,4,6,8,10};
System.out.println(numerosPares.length)
```

Mostrará 6 no console

Arrays em Java (cont.)

Arrays não alteram a sua estrutura.
Pode alterar os valores em cada uma das posições, mas não alteram de comprimento.

ArrayList

Agora que você já lembrou de arrays em Java, vamos aprender sobre o arrayList
ArrayList é uma das implementações de coleções disponíveis em Java.
É fácil de aprender por sua semelhança com arrays.

ArrayList (cont.)

Mas então , o que ArrayList?
ArrayList é uma classe que é lista dinâmica do pacote **java.util**.

Características principais

- Pode crescer ou diminuir automaticamente.
- Aceita elementos repetidos.
- Usa índices para acessar elementos.
- Oferece métodos como .add(), .remove(), .get(), .size().

ArrayList (cont.)

Sintaxe
(precisa importar o java.util.ArrayList

```
import java.util.ArrayList;

ArrayList<Tipo> nomeDaVariavel = new ArrayList<>();
```

Onde:
Tipo pode ser qualquer Tipo primitivo ou classe

ArrayList (cont.)

Para obter o comprimento do objeto ArrayList utiliza-se o método **size()**

Exemplo:
(Supondo que o ArrayList arr1 existe e contém 3 elementos)

```
System.out.println(arr1.size());
```

Mostrará 3 no console.

ArrayList (cont.)

Para obter o conteúdo de um posição dentro do ArrayList utiliza-se o método **get(índice)**

Exemplo:
(Supondo que o ArrayList de Strings arr1 existe e contém 3 elementos)

```
String s = arr1.get(2);
```

A string s conterá o valor da posição 2 de arr1

ArrayList (cont.)

Para alterar o conteúdo de um posição dentro do ArrayList utiliza-se o método **set(índice , novo valor)**

Exemplo:
(Supondo que o ArrayList de Strings arr1 existe e contém 3 elementos do tipo String)

```
arr1.set(1, "novo valor da string");
```

O conteúdo da posição 1 de arr1 foi alterado

ArrayList (cont.)

A classe ArrayList possui diversos outros métodos que auxiliam o uso eficiente dos seus recursos.

Para obter maiores informações, acesse a documentação da classe https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

ArrayList (cont.)

Exemplo 1

```
import java.util.ArrayList;

public class ExemploArrayList {
    public static void main(String[] args) {
        ArrayList<String> frutas = new
ArrayList<>();
```

ArrayList (cont. do Exemplo)

Exemplo 1 - continuação

```
frutas.add("Maçã");
frutas.add("Banana");
frutas.add("Laranja");
frutas.add("Abacaxi");
frutas.add("Uva");
```

ArrayList (cont. do exemplo)

Exemplo 1 Continuação

```
    System.out.println("Frutas: " + frutas);
    System.out.println("Comprimento :
"+frutas.size());
    frutas.remove("Banana");
    System.out.println("Após remover Banana: " +
frutas);
```

ArrayList (cont. do exemplo)

Exemplo 1 Continuação

```
    System.out.println("Comprimento :
"+frutas.size());
    System.out.println(frutas.get(2));
    frutas.set(3,"Abacate");;
    System.out.println("Frutas: " + frutas);
  }
}
```

Exercício: Atividade Formativa 3 – Estoque de produtos

Acesse o Canvas

Leia os requisitos cuidadosamente e faça a atividade

Dúvidas?