

Paradigmas de Programação

Aula 05 - ArrayList



Conteúdo



- Java ArrayList
- Java ArrayList vs Array Tradicional
- Sintaxe do ArrayList
- Adicionar, Editar e Remover Elementos
- ArrayList Size
- Iterar um ArrayList
- Ordenar um ArrayList

Java ArrayList



• A classe ArrayList é um array redimensionável, que pode ser encontrado no pacote java.util.

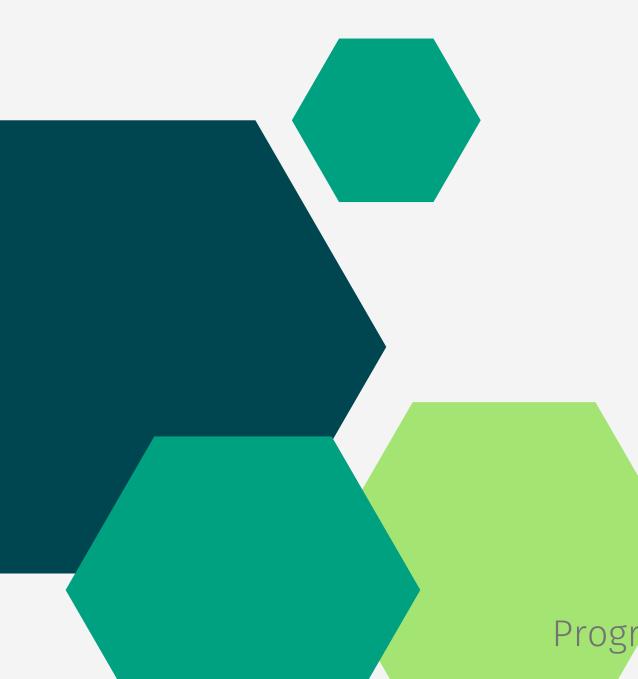
Java ArrayList vs Array



• A diferença entre um array *built-in* e um ArrayList em Java, é que o tamanho de um array não pode ser modificado (se quisermos adicionar ou remover elementos de/para um array, deve ser criado um novo). Enquanto os elementos podem ser adicionados e removidos de um ArrayList sempre que quisermos. A sintaxe também é um pouco diferente:

// import the ArrayList class
import java.util.ArrayList;
// Create an ArrayList object
ArrayList<TipoDados> nome_array = new ArrayList<TipoDados>();

Exemplo de Declaração



```
// import the ArrayList class
import java.util.ArrayList;
```

```
// Create an ArrayList object
ArrayList<String> cars = new ArrayList<String>();
```

ArrayList<Pessoa> agenda = new ArrayList<Pessoa>();

ArrayList<Atleta> competicao = new ArrayList<Atleta>();

Adicionar Elementos



• A classe ArrayList tem muitos métodos úteis. Por

Aceder um Elemento



 Para aceder um elemento no ArrayList, usamos o método get() e consulte o número do índice: cars.get(0);

Editar um Elemento



 Para modificar um elemento, usamos o método set() e indicamos o número do índice: cars.set(0, "Opel");

Remover um Elemento



 Para remover um elemento, usamos o método remove() e indicamos o número do índice: cars.remove(0);

Remover todos os Elementos



 Para remover todos os elementos do ArrayList, usamos o método clear(): cars.clear();

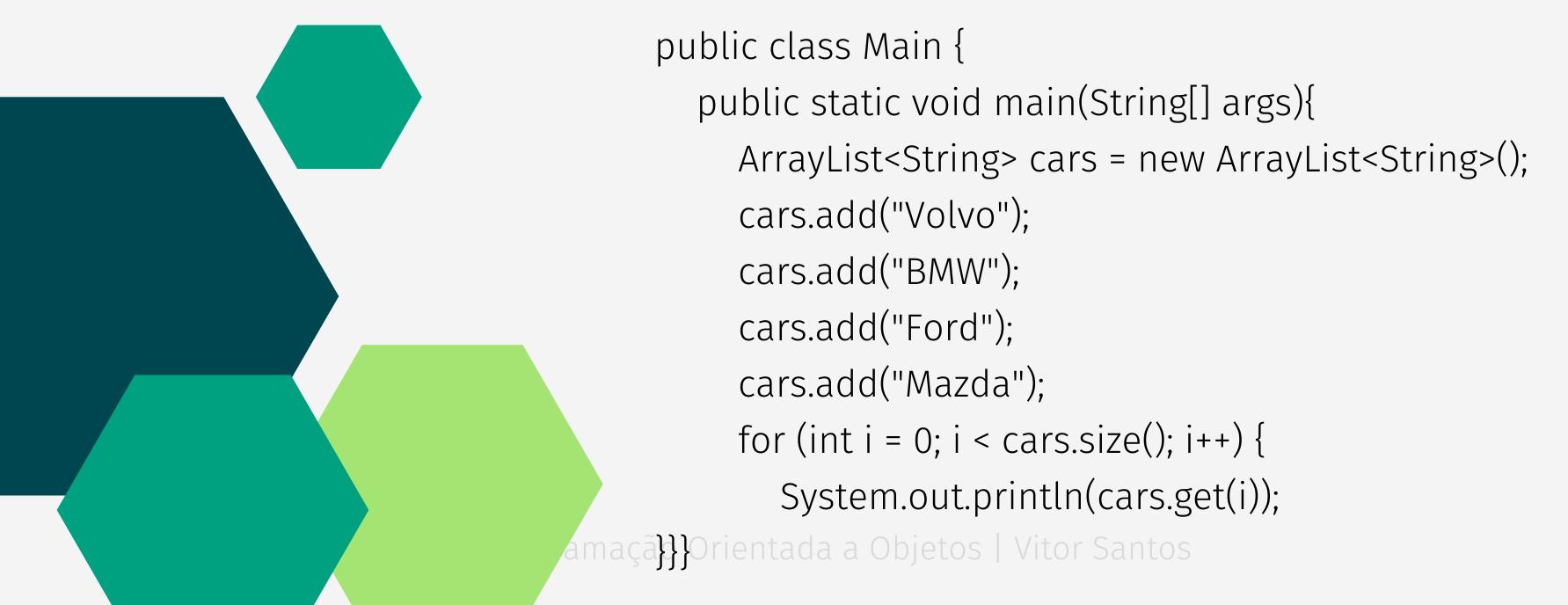
ArrayList Size



 Para descobrir quantos elementos um ArrayList tem, usamos o método size: cars.size();

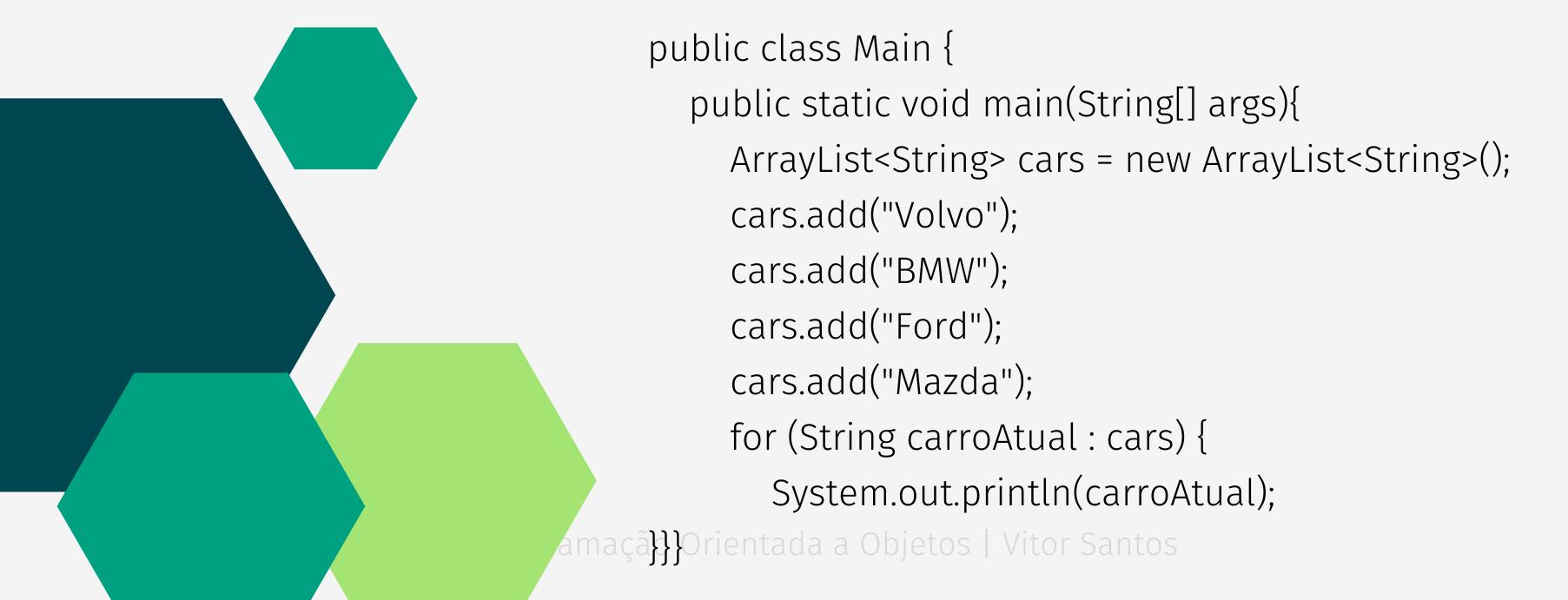
Iterar um ArrayList

• Percorremos os elementos de um ArrayList com um loop for e usamos o método size() para especificar quantas vezes o loop deve ser executado:



Iterar um ArrayList

• Também podemos percorrer um ArrayList com o loop for-each:



Ordenar um ArrayList

• Outra classe útil no pacote java.util é a classe Collections, que inclui o método sort() para ordenar listas em ordem alfabética ou numérica:



```
public class Main {
  public static void main(String[] args){
    ArrayList<String> cars = new ArrayList<String>();
     cars.add("Volvo");
     cars.add("BMW");
     cars.add("Ford");
     cars.add("Mazda");
     Collections.sort(cars); // Sort cars
     for (String carroAtual : cars) {
       System.out.println(carroAtual);
```

Ordenar um ArrayList

Outra classe útil no pacote java.util é a classe Collections, que inclui o método sort() para ordenar listas em ordem alfabética ou numérica:

```
public class Main {
  public static void main(String[] args){
    ArrayList<Integer> myNumbers = new ArrayList<Integer>();
    myNumbers.add(33);
    myNumbers.add(15);
    myNumbers.add(20);
    myNumbers.add(34);
    myNumbers.add(8);
    myNumbers.add(12);
    Collections.sort(myNumbers); // Sort myNumbers
    for (int i : myNumbers) {
       System.out.println(i);
```

Lista Métodos ArrayList

<u>Java ArrayList add()</u>

inserts the element to the arraylist

<u>Java ArrayList addAll()</u>

adds all elements of a collection to arraylist

<u>Java ArrayList clear()</u>

removes all the elements from arraylist

<u>Java ArrayList clone()</u>

makes a copy of the array list

<u>Java ArrayList contains()</u>

checks if the element is present in the arraylist

<u>Java ArrayList get()</u>

returns the element present in the specified index

Java ArrayList indexOf()

returns the position of the specified element

<u>Java ArrayList removeAll()</u>

removes multiple elements from the arraylist

<u>Java ArrayList remove()</u>

removes the single element from the arraylist

<u>Java ArrayList size()</u>

returns the length of an arraylist

<u>Java ArrayList isEmpty()</u>

checks if the arraylist is empty

Java ArrayList subList()

returns a portion of the arraylist

<u>Java ArrayList set()</u>

replace the single element from an arraylist

<u>Java ArrayList sort()</u>

sorts the arraylist according to specified order

<u>Java ArrayList toArray()</u>

converts an arraylist to an array

<u>Java ArrayList toString()</u>

converts the arraylist into a String

<u>Java ArrayList ensureCapacity()</u>

set the size of an araylist

Java ArrayList lastIndexOf()

returns position of last occurrence of the element

<u>Java ArrayList retainAll()</u>

retains only the common elements

<u>Java ArrayList containsAll()</u>

checks if a collection is a subset of arraylist

<u>Java ArrayList trimToSize()</u>

trims the capacity of arraylist equal to the size

<u>Java ArrayList removeRange()</u>

removes a portion of the arraylist

<u>Java ArrayList replaceAll()</u>

replace all elements from the arraylist

<u>Java ArrayList removelf()</u>

removes element that satisfy the condition

Java ArrayList forEach()

performs an action to all elements of arraylist

<u>Java ArrayList iterator()</u>

returns an iterate to loop through the ArrayList



Paradigmas de Programação

Aula 05 - ArrayList

