Java ArrayList

AGENDA

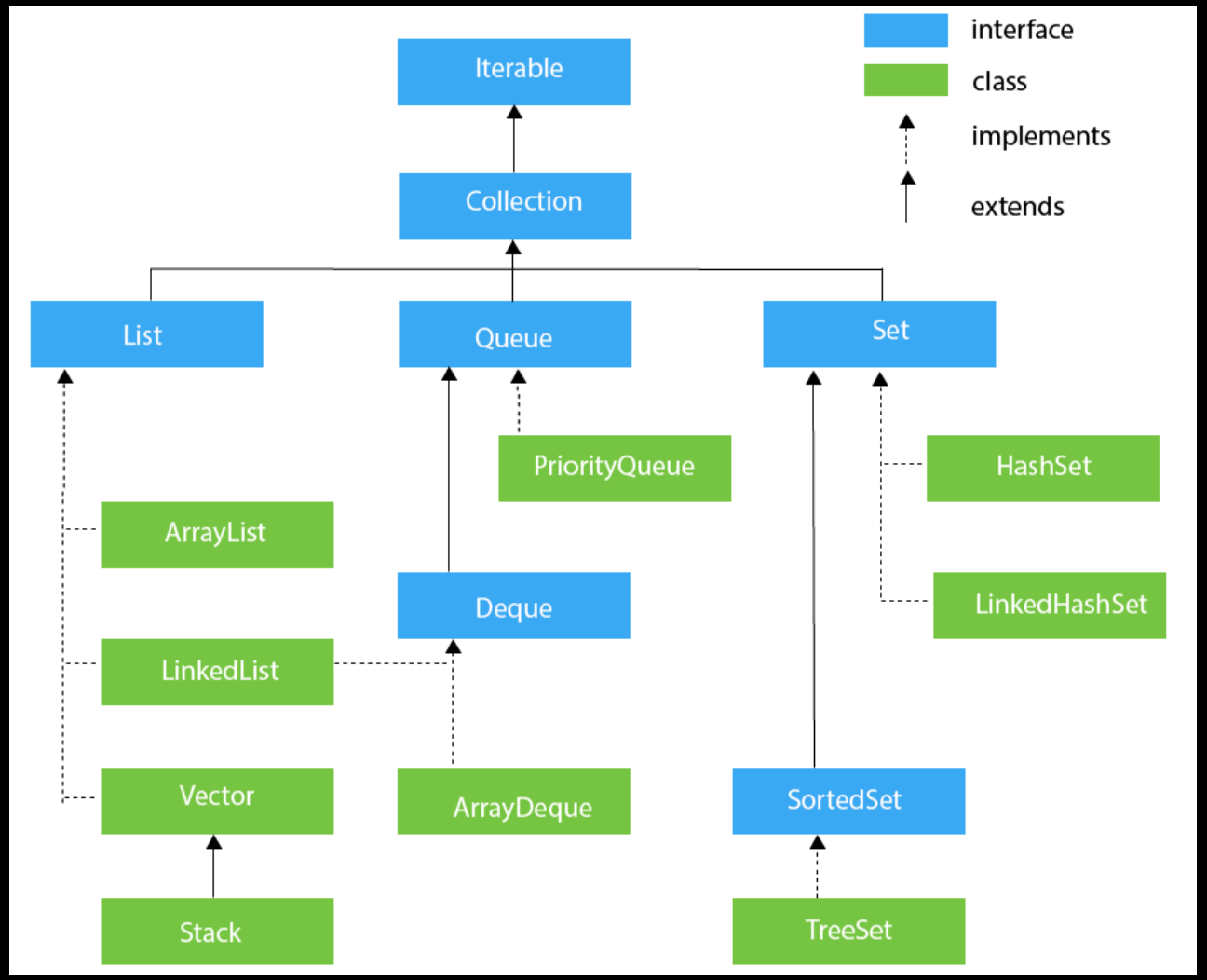
1. Introdução a ArrayList
2. Manipulação de dados com ArrayList
3. Principais métodos integrados
4. *Generics*
5. Introdução a *Collections*
6. Ordenação - método sort()
7. Aplicações práticas

Nesta aula iremos abordar a estrutura de dados ArrayList e como podemos *manipular dados* através dela

ArrayList

A Classe ArrayList permite a criação de uma lista dinâmica (Coleção de Objetos), e pode ser encontrada no pacote java.util

Hierarquia das classes e interfaces de *Collection*



- As classes e interfaces da estrutura de coleções são membros do pacote **java.util**

Diferença entre Array e ArrayList

- O tamanho de um Array não pode ser modificado
- Se houver a necessidade de adicionar ou remover um elemento será preciso criar um novo array
- No ArrayList, os elementos podem ser adicionados e removidos sempre que necessário
- Sintaxe:

```
import java.util.ArrayList;
```

Instanciando um ArrayList (lista) heterogêneo (lista heterogênea)

```
ArrayList lista = new ArrayList();
```

ArrayList com *Generics*

ArrayList com *Generics*

- Permite a criação de uma lista **homogênea** (tipo específico de objeto)
  - melhora a confiabilidade do programa
  - extremamente recomendado o uso de *Generics*
  - *adição de um elemento de outro tipo causará um erro, podendo ser capturado em tempo de compilação*
  - Sintaxe:

```
ArrayList<TipoObjeto> lista = new ArrayList<TipoObjeto>();
```

Instanciando um ArrayList (lista) do tipo String (lista homogênea)

```
ArrayList<String> lista = new ArrayList<String>();
```

Instanciando um ArrayList (alunos) do tipo Aluno (lista homogênea)

```
ArrayList<Aluno> alunos = new ArrayList<Aluno>();
```

ArrayList com *Generics*

- A utilização de *Generics* evita o *casting* de métodos e objetos
  - Exemplo (sem a utilização de *Generics*)

```
String nome = (String) lista.get(0);
```

  - Exemplo (com a utilização de *Generics*)

```
String nome = lista.get(0);
```

ArrayList - Principais métodos

Método	Descrição
boolean **add** (T obj)	Insere um elemento no final da lista e **sempre** retorna um *true* (boolean)
void **add**(int index, T obj)	Insere o elemento especificado na posição indicada da lista
int **size**()	Retorna o número de elementos (tamanho) armazenados na lista
void **set**(int index, T obj)	Coloca um elemento em um índice específico da lista, **sobrescrevendo** o elemento anterior
T **get**(int index)	Obtém o elemento armazenado em um índice específico
T **remove**(int index)	Remove um elemento e descola para baixo todos os elementos acima dele. O elemento removido é retornado
void **clear**()	Remove todos os elementos da lista
boolean **contains**(T object)	Retorna verdadeiro se a lista contém o elemento especificado e falso caso contrário
boolean **isEmpty**()	Retorna verdadeiro se a lista estiver vazia e false caso contrário

Manipulação de Dados com ArrayList

A Classe ArrayList possui vários métodos úteis para manipulação de dados

Método	Descrição
boolean **add** (T obj)	Insere um elemento no final da lista e **sempre** retorna um *true* (boolean)
void **add**(int index, T obj)	Insere o elemento especificado na posição indicada da lista
int **size**()	Retorna o número de elementos (tamanho) armazenados na lista
void **set**(int index, T obj)	Coloca um elemento em um índice específico da lista, **sobrescrevendo** o elemento anterior
T **get**(int index)	Obtém o elemento armazenado em um índice específico
T **remove**(int index)	Remove um elemento e descola para baixo todos os elementos acima dele. O elemento removido é retornado
void **clear**()	Remove todos os elementos da lista
boolean **contains**(T object)	Retorna verdadeiro se a lista contém o elemento especificado e falso caso contrário
boolean **isEmpty**()	Retorna verdadeiro se a lista estiver vazia e false caso contrário

Criação do ArrayList

```
import java.util.ArrayList; //importa a Classe ArrayList

...

//cria um ArrayList de Objetos (String)
ArrayList<String> carros = new ArrayList<String>();

//adicionando elementos em um ArrayList
carros.add("BMW");
carros.add("Ford");
carros.add("Mustang");

//imprimindo os elementos do ArrayList
System.out.println(carros);
```

Adição de um elemento - método add()

```
...

//adicionando elementos em um ArrayList
carros.add("BMW");
carros.add("Ford");
carros.add("Mustang");

//imprimindo os elementos do ArrayList
System.out.println(carros);
```

Acessando um elemento - método get()

```
...

//acessando um elementos através do seu índice
carros.get(0);
```

- índices dos Arrays iniciam em 0: [0] é o primeiro elemento, [1] é o segundo…

Alterando um elemento - método set()

```
...

//alterando um elementos através do seu índice
carros.set(0, "BMW");
```

Removendo um elemento - método remove()

```
...

//removendo um elementos através do seu índice
carros.remove(0);
```

- [0] é índice em que o elemento será removido

Removendo todos os elementos - método clear()

```
...

//removendo todos os elementos do ArrayList
carros.clear();
```

- [0] é índice em que o elemento será removido

Tamanho do ArrayList - método size()

```
...

//alterando um elementos através do seu índice
carros.size();
```

- [0] é índice em que o elemento será removido

Percorrendo um ArrayList - for loop e size()

```
 ...

 //percorrendo ArrayList
for(int i = 0; i < cars.size(); i++) {
    System.out.println(carros.get(i));
}
```

Percorrendo um ArrayList com for-each

```
 ...

 //percorrendo ArrayList
for(String i : cars) {
    System.out.println(i);
}
```

ArrayList - outros tipos de objetos

ArrayList - Outros tipos

- Os elementos em um ArrayList são **objetos**
- **String em Java é um objeto** (não um tipo primitivo)
- Para utilizar outros tipos, como int, por exemplo, é preciso especificar a classe wrapper equivalente: Integer
- Para outros tipos primitivos: Boolean para boolean, Character para char, Double para double…

Criando um ArrayList de Integer

```
  ...

  //ArrayList de Integers
ArrayList<Integer> listaNumeros = new ArrayList<Integer>();
    listaNumeros.add(10);
    listaNumeros.add(15);
    listaNumeros.add(20);
    listaNumeros.add(25);

   for (int i : listaNumeros) {
       System.out.println(i);
    }
```

Ordenando um ArrayList (Classe Collections)

Ordenando um ArrayList - método sort()

```
  ...
import java.util.ArrayList;
import java.util.Collections;  // Importa a Classe Collections


…
    ArrayList<String> carros = new ArrayList<String>();
    carros.add("Volvo");
    carros.add("BMW");
    carros.add("Ford");
    carros.add("Jaguar");

    Collections.sort(carros);  // Ordena a lista carros

    for (String i : carros) {
      System.out.println(i);
    }
 …
```

Ordenando um ArrayList - método sort()

```
  ...
import java.util.ArrayList;
import java.util.Collections;  // Importa a Classe Collections


  ...
    ArrayList<Integer> listaNumeros = new ArrayList<Integer>();
    listaNumeros.add(33);
    listaNumeros.add(15);
    listaNumeros.add(20);
    listaNumeros.add(34);
    listaNumeros.add(8);
    listaNumeros.add(12);

    Collections.sort(listaNumeros);  // Ordena a lista carros

    for (int i : listaNumeros) {
      System.out.println(i);
    }


  ...
```

Aplicação prática (eclipse)

Classes Java Eclipse

```
package aula17_ArrayList;
```

Aluno.java
ListaAlunos.java
ListaAlunosTeste.java

Obrigado e até a próxima aula!