Estruturas de dados

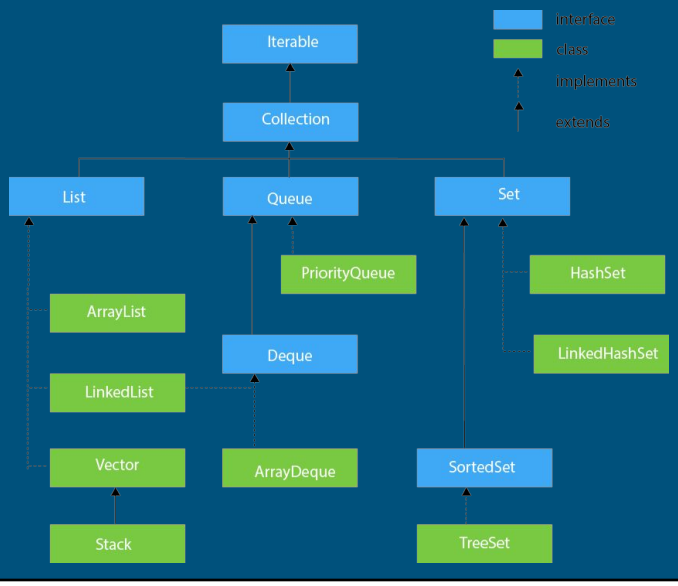
Coleções Java

emerson@paduan.pro.br

Coleções

emerson@paduan.pro.br

Classes para ED em Java



ArrayList

ArrayList<T> (pacote java.util) pode alterar dinamicamente seu tamanho para acomodar mais elementos.

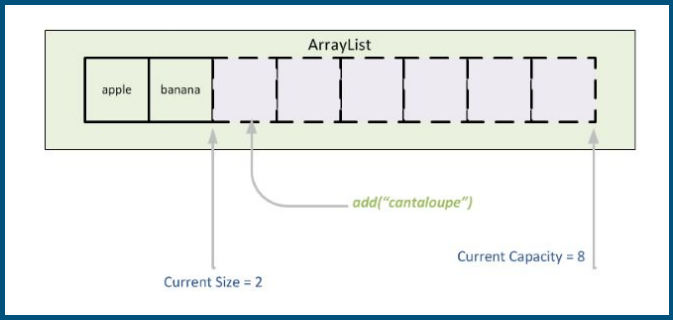
– *T* indica o tipo de elemento armazenado na coleção

– Isso é semelhante a especificar o tipo ao declarar um array, exceto que apenas tipos *não-primitivos* podem ser utilizados com essas classes de coleção.

Classes com essa espécie de marcador de lugar são chamadas *classes genéricas*.

ArrayList

Coleção em Java que permite armazenar elementos de forma DINÂMICA.



ArrayList

Exemplo:

```
import java.util.ArrayList;

public class Exemplo {
	public static void main(String args[])
	{
		ArrayList<String> nomes = new ArrayList<>();

		nomes.add("Huguinho");
		nomes.add("Zezinho");
		nomes.add("Luizinho");

		System.out.println(nomes);
	}
}
```

Principais métodos

Método	Descrição
add(Object o)	Adiciona um elemento ao fim do ArrayList
add(int index, Object o)	Adiciona um elemento no índice especificado do ArrayList
clear()	Remove todos os elementos do ArrayList
get(int index)	Retorna o elemento do índice especificado
indexOf(Object o)	Retorna o índice da primeira ocorrência do elemento especificado no ArrayList
remove(Object o)	Remove a primeira ocorrência do valor especificado
remove(int index)	Remove o elemento do índice especificado
size()	Retorna o número de elementos armazenados no ArrayList
isEmpty()	Retorna true se não existem elementos no ArrayList

foreach

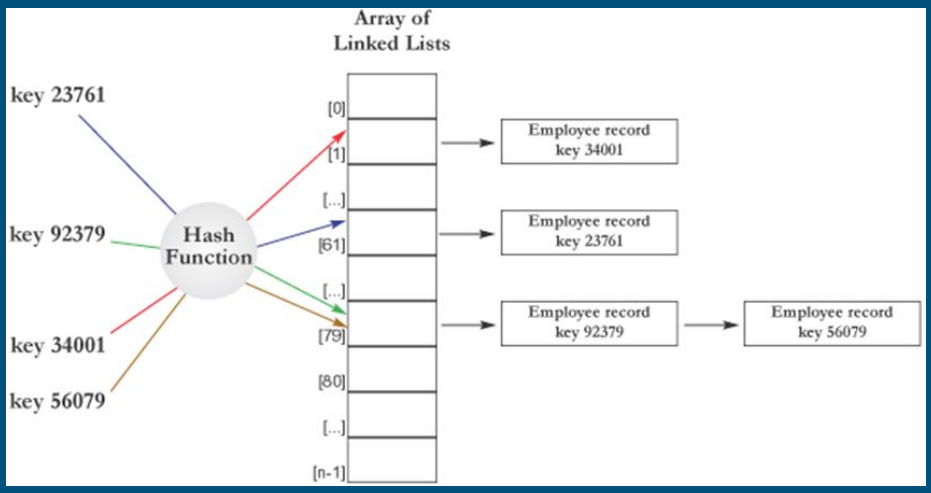
foreach – (para cada)

iterar sobre coleções de maneira simples e direta

Sintaxe:

```
for( tipo variavel : nomeArray ){
    //corpo do for
}
```

Hash



Em Java

```
public static void main(String[] args) {
    HashMap<Integer, String> mapa = new HashMap<>();

    mapa.put(1, "um");
    mapa.put(2, "dois");
    mapa.put(3, "três");
    mapa.put(4, "quatro");

    System.out.println(" 3 = " + mapa.get(3));
    System.out.println(" 3 = " + mapa.get(32));
}
```

Associação entre classes

emerson@paduan.pro.br

Associação

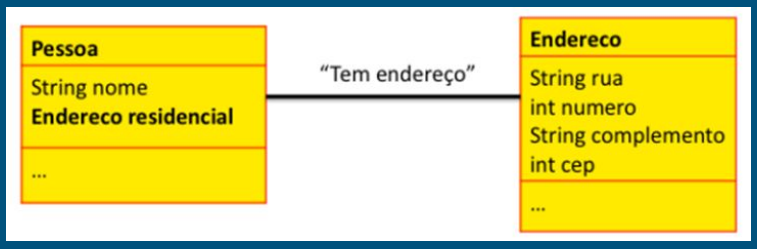
Um sistema é composto por várias Classes.

• As classes se conectam para poderem se comunicar por troca de mensagens (chamadas de métodos)

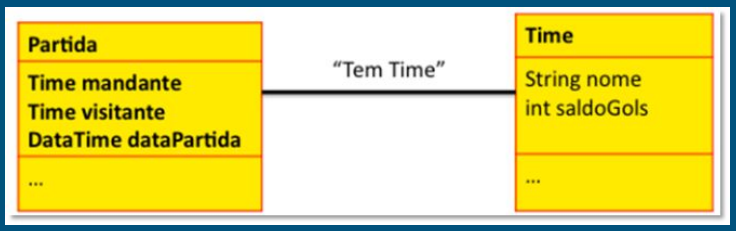
• Quando um ou mais atributos de uma Classe é uma *referência* para outra Classe temos uma **associação**.

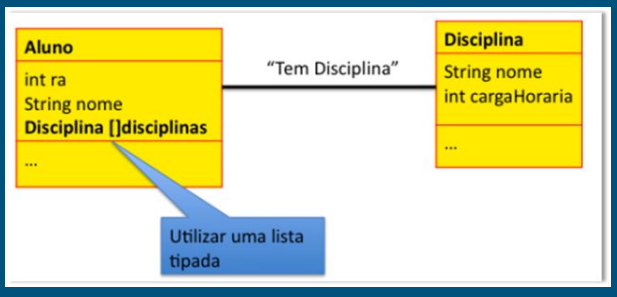
emerson@paduan.pro.br

Exemplo



Exemplo 1: N





Em Java

```
public class Pessoa{
   //atributos
   private String nome;
   private int idade;
   private char sexo;
   private Endereco end;
   ....

   public String imprimir(){
      return "Nome: " + nome +
             "\nIdade: " + idade +
             "\nSexo: " + sexo +
             "Endereço: " + end.imprimir();
   }
}
```

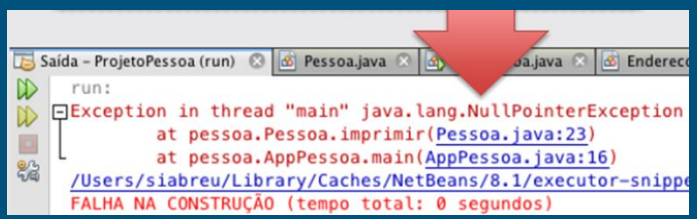
```
public class Endereco{
   //atributos
   private String logradouro;
   private String complemento;
   private int numero;
   private String cep;
   ....

   public String imprimir(){
      return "Logradouro: " + logradouro +
             "\nComplemento: " + complemento +
             "\nNúmero: " + numero +
             "CEP: " + cep;
   }
}
```

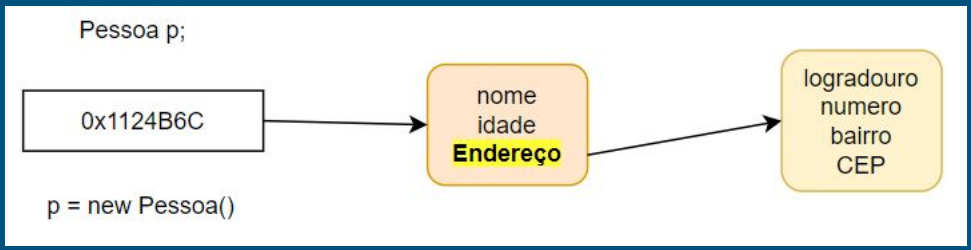

Atenção!!!

```
public class AppPessoa {
   public static void main(String[] args) {
      Pessoa objPessoa = new Pessoa();

      System.out.println(objPessoa.imprimir());
   }
}
```



Criar objeto no construtor



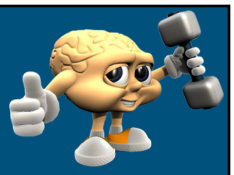
Resolvendo

```
public class Pessoa{
    //atributos
    private String nome;
    private int idade;
    private char sexo;
    private Endereco end;

    //construtor default
    public Pessoa() {
        this.end = new Endereco();
    }

    public String imprimir(){
        return "Nome: " + nome +
               "\nIdade: " + idade +
               "\nSexo: " + sexo +
               "Endereço: " + end.imprimir();
    }
}
```

Exercício



1. Criar um relacionamento de Associação entre as classes Animal e Proprietário:
   Animal “Tem UM” *Proprietario*

2. Criar a classe *Animal* com os seguintes atributos: nome, raça, ano de nascimento e proprietário
3. Criar a Classe *Proprietario* com os seguintes atributos: nome e telefone
4. Criar uma classe main

emerson@paduan.pro.br