

Cap 06
Collections

Parte 1 - Introdução

Linguagem de Programação III
Prof. Otacílio José Pereira

Objetivos e Tópicos

- Compreender a framework collections
- Realizar primeiras ações com collections

Tópicos

- *Programas, Funcionalidades e Dados*
- *Correlação com Estruturas de Dados*
- *Conceito de frameworks*
- *Framework collections*
- *Um primeiro exemplo: ArrayList*





Contexto

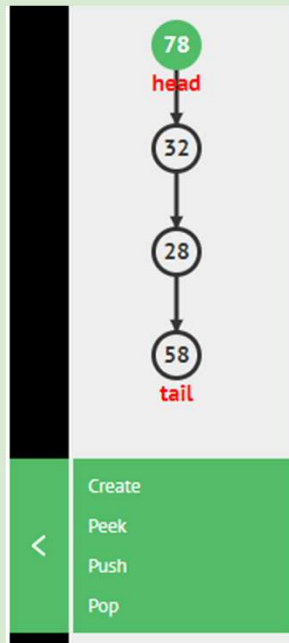
- Onde estamos?

Onde estamos?

- ▶ *Considerando nosso planejamento inicial*
- ▶ Capítulo 1 – Introdução
- ▶ Capítulo 2 - Conceitos básicos de Orientação a Objetos
 - ▶ OO, classes e objetos
 - ▶ Atributos, métodos, encapsulamento, getters, setters
 - ▶ Outros tópicos: Atributos de classe (static), tipos primitivos, tipos por referência
- ▶ Capítulo 3 – Herança e Polimorfismo
 - ▶ Parte 1 – Herança e breve introdução a Polimorfismo
 - ▶ Parte 2 – Aplicação dos conceitos em um exemplo de UI
- ▶ Capítulo 4 – Classes abstratas e interfaces
- ▶ Capítulo 5 – Collections
- ▶ Capítulo 6 – Outros tópicos

Foco





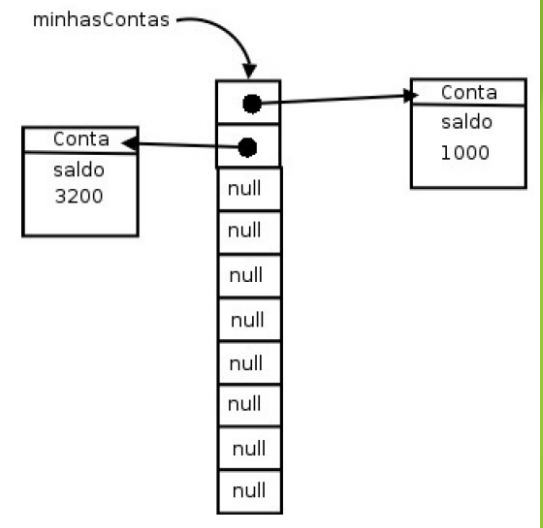
Programas e a manipulação de dados

- Dados dos programas e as estruturas
- Definição
- Estruturas típicas

Programa: Lógica + Dados

- Programas envolvem lógica e dados
 - Compreender as estruturas de dados ou collections ajudam a compreender como informações são manipuladas pelos sistemas e tecnologias
 - Qual forma os dados estavam organizados até então?
 - O ponto central era o trato com vetores
 - Ao final fizemos um experimento com Vetores + Objetos/Registros
- (Em Linguagem C, um vetor de estruturas)

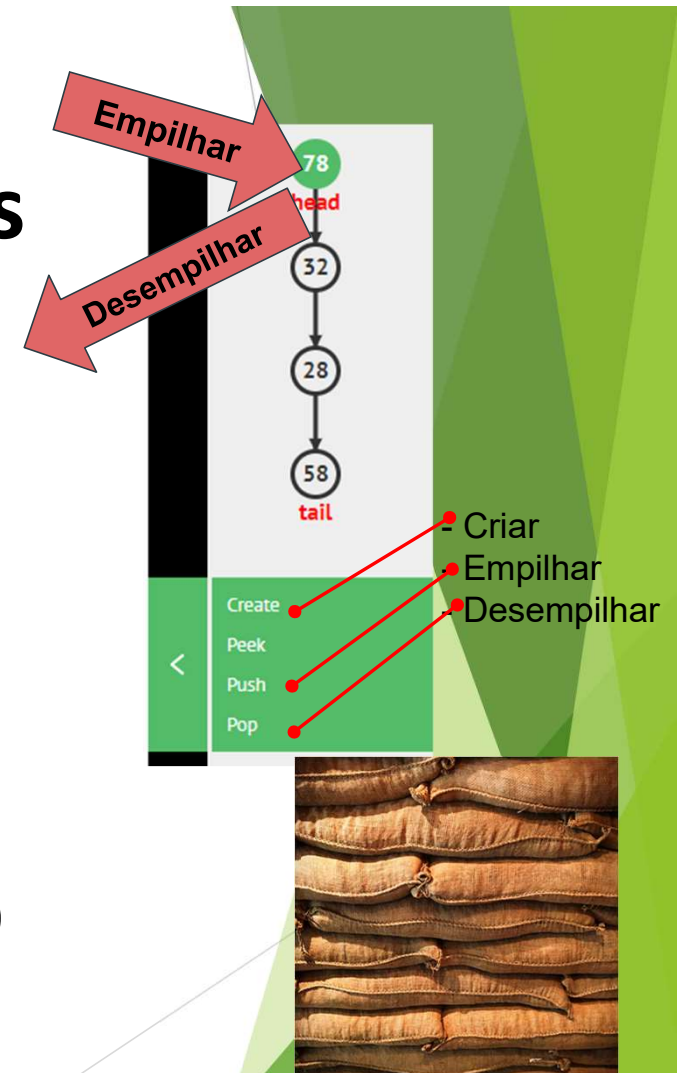
Registros



regNome	Adriano	Flávia	...			
regProduto	Bicicl.	Churr.	...			
regData	2020/10/	2020/10/	...			
regValor	10.5	15	...			

Collections e estruturas de dados

- Collections tem forte relação com a disciplina de estruturas de dados
- Ou um termo associado é Tipos Abstratos de Dados
- A ideia chave é que os dados manipulados pelos programas podem ser armazenadas (estrutura) e manipuladas (operações) de forma peculiar e genérica ou obedecendo alguns “padrões”
- Estrutura: Informações e a sua estruturação (elos)
- Operações: Como os dados são manipulados



Organização dos dados em programas

- Em um primeiro exercício já podemos nos lembrar de alguns recursos de linguagem de programação que lidam com dados

- **Variáveis e tipos simples**

- Nosso exemplo de cálculo de média

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <?php
5
6 // Converte graus Celsius em graus Fahrenheit
7 function Media($Nota1, $Nota2)
8 {
9     $Media = ($Nota1 + $Nota2) / 2;
10
11     echo "<p>A média entre " . $Nota1 . " e " . $Nota2 . " <br>";
12     echo "Média = " . $Media . "</p>";
13 }
14
15 Media(2.5, 8.7);
16 ?>
17
18 </body>
19 </html>
```

Nota 1: float

Nota 2: float

Média: float

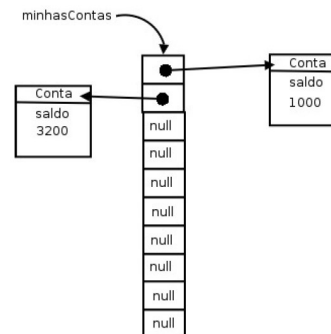
- **Vetores e Matrizes**

- Já surge o conceito de conjunto de dados
 - Nosso exemplo de pesquisa e ordenação

regNome	Adriano	Flávia	...			
regProduto	Bicicl.	Churr.	...			
regData	2020/10/	2020/10/	...			
regValor	10.5	15	...			

- **Classes e objetos (Estruturas)**

- Agregado de dados
 - Combinação com Vetores e Objetos

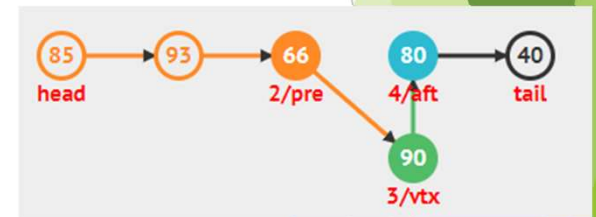
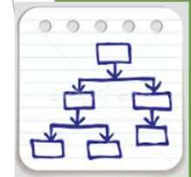


Registros

Nome	Adriano	Nome	Flávia	...	Nome	
Produto	Bicicleta	Produto	Churras	...	Produto	
Data	2020/10/09	Data	2020/10/19	...	Data	
Valor	10.5	Valor	15	...	Valor	

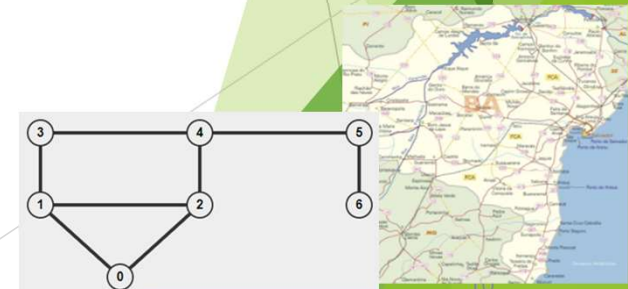
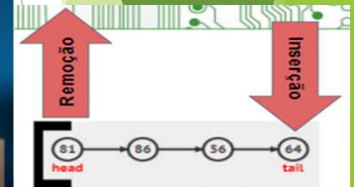
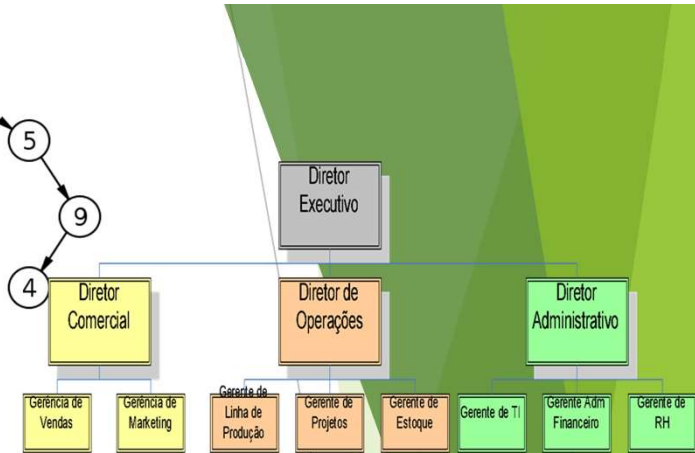
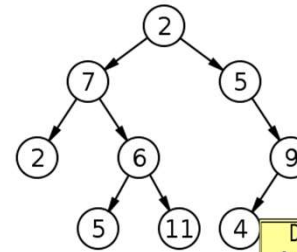
Collections (Estruturas de Dados)

- As **estruturas de dados ou collections** permitem sofisticar a organização dos dados em soluções computacionais
- Em uma definição formal,
 - Visa organizar dados e operações (algoritmos) que podem ser aplicadas como forma de apoio a solução de problemas (complexos)



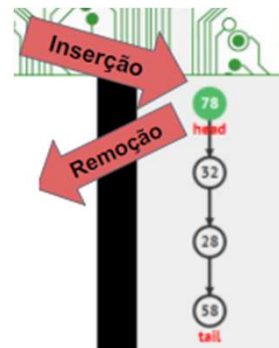
Exemplos e aplicações

- Imagine as seguintes soluções e como o conjunto de dados se comportam?
 - Pensando o WhatsApp, quais as Informações e como elas são organizadas para visualização?
 - E o conjunto de contatos, quais informações e operações?
 - Uma imobiliária que precisa de organizar os interessados em alugar um apartamento, como ela organiza estes interessados? Quem tem prioridade para o aluguel?
- Percebem como as informações de cenários reais possuem suas entidades e a estruturação entre elas!



Estruturas de Dados Típicas ou Clássicas

- Quando estamos estudando estruturas de dados clássicas, existem algumas bem típicas
 - Pilha
 - Fila
 - Lista
 - Árvores
 - Grafos
- Perceba que independente do cenário, a figura acima ilustra por exemplo uma lista (de contatos, ou de mensagens, ou de interessados em aluguéis)



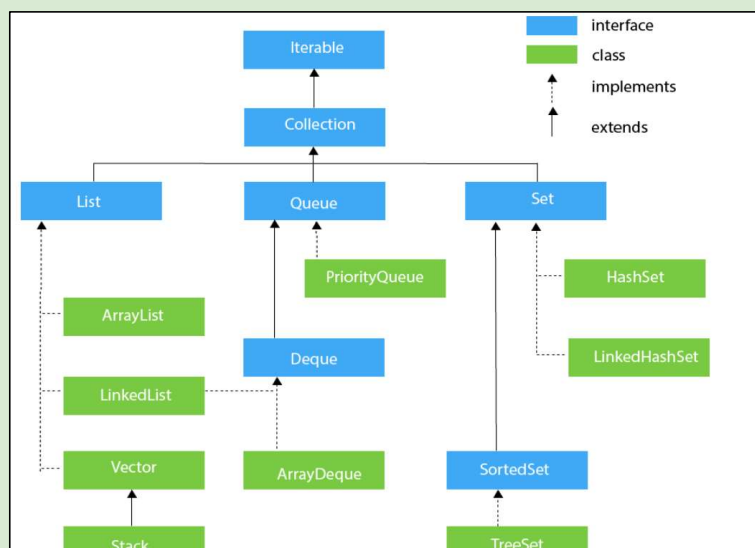
Reflita

- Analise um dia seu do início ao fim, que estruturas você costuma perceber no seu cotidiano! Como os dados se comportam na sua manipulação?



Síntese

- Programa/Software : Lógica-Processamento + Dados
- E precisam refletir um cenário/problema do mundo real
- Dados podem ser tratados usando diferentes recursos
 - Variáveis simples
 - Variáveis compostas
 - Estruturas de dados (ou Collections)
- Estruturas de dados são formados
 - Pelos nós/entidades/"objetos"
 - Pela estruturação/elos/correlações
- **Questão chave: Qual cenário real?**
 - > **Qual estrutura de dados/Collections?**



Collections

- Primeiro exemplo
- Introdução
- Collections framework

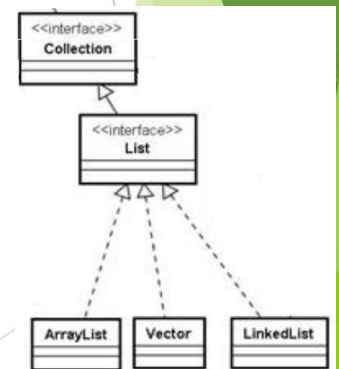
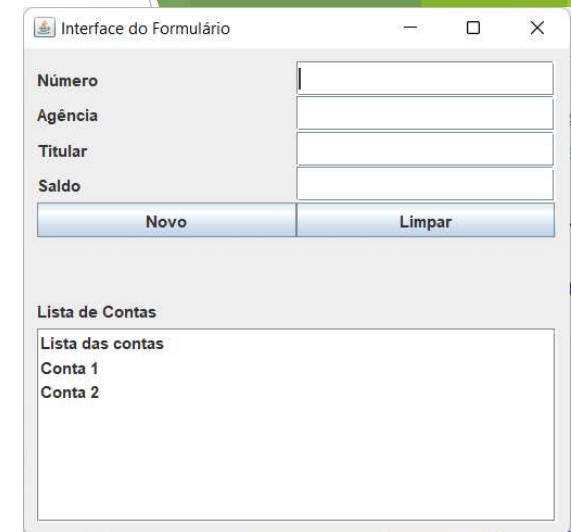
Primeiro exemplo

- Observe o programa ao lado
- Qual collection está sendo usada?
- Como os dados são inseridos, em que ordem?
- O que será impresso?
- Sabemos como internamente os dados são organizados?

```
public static void main(String args[])
{
    ArrayList<String> l = new ArrayList();
    l.add("Fabiana");
    l.add("Marcelo");
    l.add("Cristina");
    for (int i=0; i<l.size(); i++)
    {
        String aux = l.get(i);
        System.out.println(" Nome : " + aux);
    }
}
```

Collections Framework

- Framework
 - Em geral é um termo usado para um pacote ou conjunto de pacotes para fornecer componentes úteis para alguma finalidade
 - Exemplos:
 - Swing: usamos para interface gráfica
 - Acesso a dados
 - Desenho de figuras
- **Collections framework**
 - Fornece um conjunto destes componentes para lidar com estruturas de dados avançadas
 - Por isso foi dada uma base de estruturas de dados antes



Collections Framework

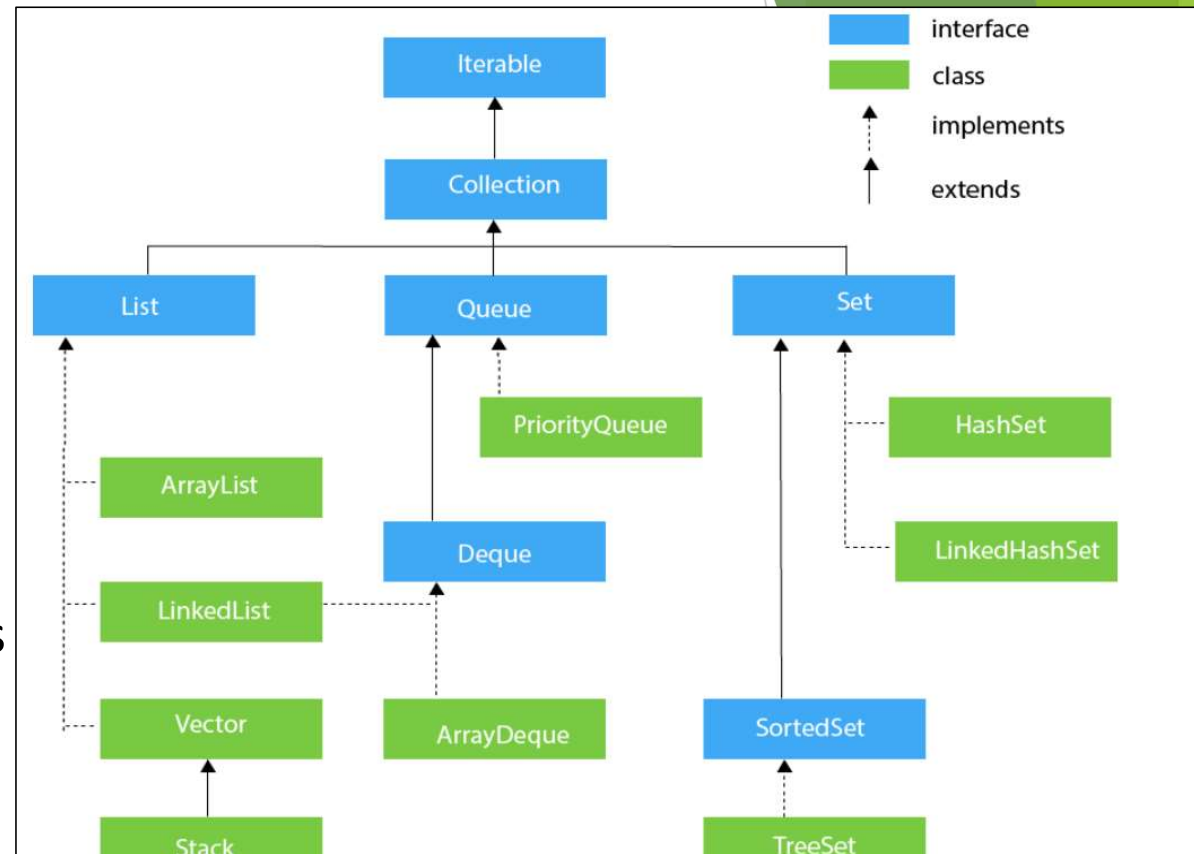
- Por que usar?
 - Lembra quando criamos nossa interface gráfica?
 - Como implementamos a inserção de novos elementos
 - Havia um vetor
 - Havia um inteiro para adicionar
 - Precisávamos o tempo inteiro gerenciar estes elementos!
- No nosso exemplo anterior, tivemos estas preocupações?

```
// Adicionando o objeto no vetor  
ultimaConta++;  
contas[ultimaConta] = c;
```

```
public static void main(String args[])  
{  
    ArrayList<String> l = new ArrayList();  
    l.add("Fabiana");  
    l.add("Marcelo");  
    l.add("Cristina");  
    for (int i=0; i<l.size(); i++)  
    {  
        String aux = l.get(i);  
        System.out.println(" Nome : " + aux);  
    }  
}
```

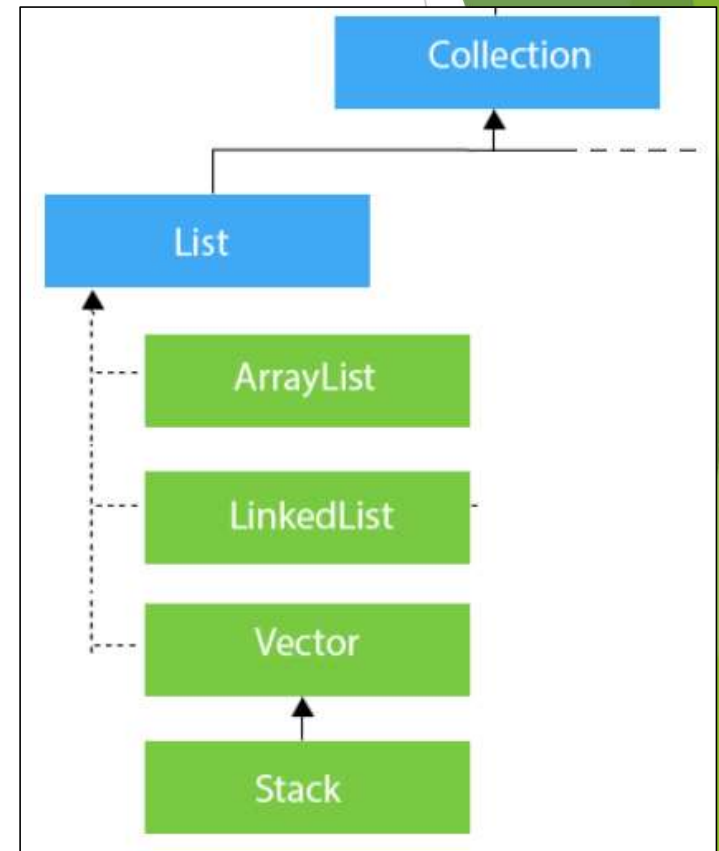
Visão geral

- A Collections framework envolve um conjunto de interfaces e classes representando os diversos tipos de estruturas de dados que podem ser usadas
- **Set**
 - Não aceita objs duplicados
- **List**
 - Aceita duplicados
 - Maior controle da posição
- **Queue**
 - Controle de ordem de chegada



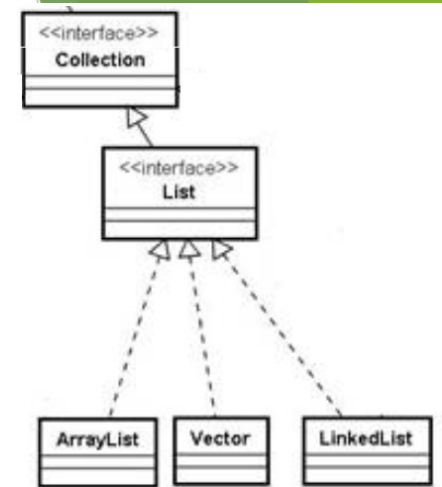
Curiosidade: Uso de Interfaces

- Vamos aproveitar o contexto para reforçar o conceito de interfaces!
- Reflita
 - É possível instanciar um objeto diretamente da interface List
`List l = new List();`
 - Existe código nas interfaces List e Collection?
 - Qual o papel delas na hierarquia?
 - Caso você crie o seu tipo Lista, como “acoplar” à hierarquia para fazer algo assim!
`List l = new MinhaLista();`



Proposta de Exercício

- Pesquise quais métodos tratam as operações a seguir da interface List e por consequência da classe ArrayList
- Complemente o código com alguns destes métodos
 - Adicionar elemento ao final
 - Adicionar elemento em certa posição
 - Remover item do final
 - Remover item do início
 - Recuperar item em certa posição
 - Ordenar a lista
 - Experimente outras formas de percorrer a lista
 - Avalie outros que interessar



```
public static void main(String args[])
{
    ArrayList<String> l = new ArrayList();
    l.add("Fabiana");
    l.add("Marcelo");
    l.add("Cristina");
    for (int i=0; i<l.size(); i++)
    {
        String aux = l.get(i);
        System.out.println(" Nome : " + aux);
    }
}
```