



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



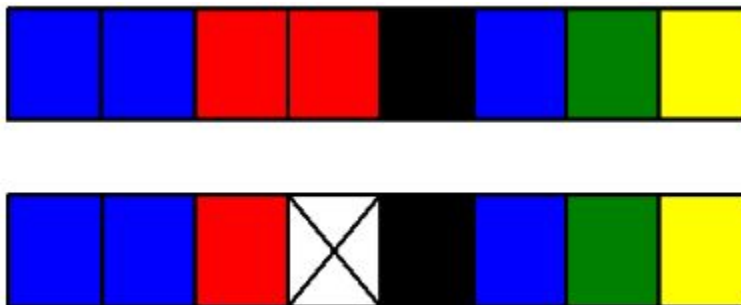
Curso Técnico em Informática

Alocação Dinâmica: Listas

Prof: Diego Corrêa

Alocação Dinâmica

- Vetores não podem ser redimensionados
- Para trabalhar com Vetores, é necessário criar um conjunto de métodos para operá-los
- Ex: De contas bancárias

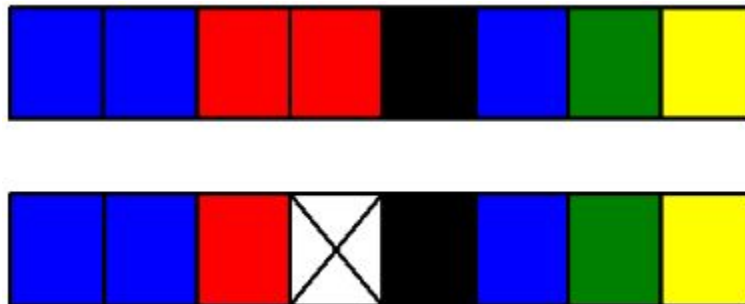


Retire a quarta Conta

`conta[3] = null;`

Alocação Dinâmica

- Neste caso, é necessários métodos para analisar se uma posição é nula
- Em outros casos, é necessário métodos para verificar se uma posição já está ocupada, antes de definir objeto



Retire a quarta Conta

`conta[3] = null;`

Alocação Dinâmica

- Devido a isso é necessário uma alocação dinâmica da memória
- Com métodos que tornem possível a operação das **Listas**

<ArrayList>

- São listas dinâmicas
- Sua grande vantagem é que seu acesso é parecido como os vetores
- Isso deixa o acesso mais rápido

<ArrayList>

- Para usá-la, é necessário importar a classe:
- **java.util.ArrayList**
- **java.util.List**
- ... existem outros tipos de Lista, como:
 - **Set**: Elementos não se repetem
 - **Map**: Mapeamento chave-valor
 - ...

<ArrayList>

- Instanciando:
 - `ArrayList lista = new ArrayList();`
- Alguns métodos da ArrayList:
 - `lista.add(objeto);`
 - `lista.add(indice,objeto);`
 - `lista.size();`
 - `lista.remove(objeto);`
 - `lista.remove(indice);`
 - `lista.forEach((objeto) -> método(objeto))`
 - ... diversos outros

<ArrayList>

Exercício

1. Criar um projeto: Alocação_Dinamica
2. Criar uma lista
3. Adicionar 3 Strings: Manoel, Joaquim, Maria
4. Imprimir
5. Adicionar 2 Strings: Ana, Carlos
6. Imprimir

<ArrayList>

```

package alocacao_dinamica;
import java.util.ArrayList;

/**
 *
 * @author lowe
 */
public class Alocacao_Dinamica {

    /**
     * @param args comandos vindos de fora do programa
     */
    public static void main(String[] args) {
        ArrayList lista = new ArrayList();
        lista.add("Manoel");
        lista.add("Joaquim");
        lista.add("Maria");
        System.out.println(lista);
        lista.add("Ana");
        lista.add("Carlos");
        System.out.println(lista);
    }
}
    
```

<ArrayList>

- É possível limitar o tipo de objeto a ser incluso numa ArrayList
- Casting toda vez que uma Object for usada é um trabalho repetitivo e cansativo
- Para limitar o tipo, devemos fazer a seguinte modificação da declaração

```
ArrayList<Tipo do Objeto> lista = new  
ArrayList<Tipo do Objeto>();
```

<ArrayList>

1. Complementando o exercício anterior
2. Crie um método que receba uma String e imprima na tela
3. Na ArrayList lista use o método `forEach()` para imprimir cada String
4. Use o método que diz o tamanho do ArrayList para imprimir quantos elementos tem
5. Adicione as Strings: Ana, Carlos
6. Remova a String: Manuel
7. Faça o 3 e 4 novamente

<ArrayList>

```
public class Alocacao_Dinamica {

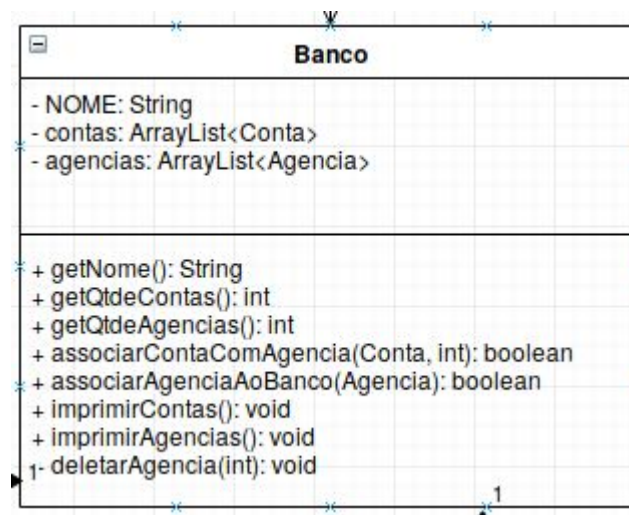
    /**
     * @param args comandos vindos de fora do programa
     */
    public static void main(String[] args) {
        ArrayList<String> lista = new ArrayList();
        lista.add("Manoel");
        lista.add("Joaquim");
        lista.add("Maria");
        lista.forEach((elemento) -> Alocacao_Dinamica.imprimirDadosFormatados(elemento));
        System.out.println("O tamanho da list é: " + lista.size());
        lista.add("Ana");
        lista.add("Carlos");
        lista.remove("Manoel");
        lista.forEach((elemento) -> Alocacao_Dinamica.imprimirDadosFormatados(elemento));
        System.out.println("O tamanho da list é: " + lista.size());
    }

    /**
     * Recebe uma String e exibe na tela
     * @param elemento uma String
     * @see
     */
    public static void imprimirDadosFormatados(String elemento) {
        System.out.println("Nome: " + elemento);
    }
}
```


<ArrayList>

UML

No UML a chamada do ArrayList é dado a seguir:



<ArrayList>

Atividade Final
Adaptação do Sistema do Banco

