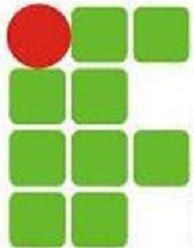
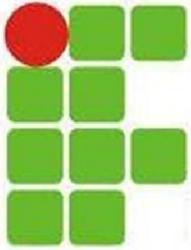

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 16:
Collections**

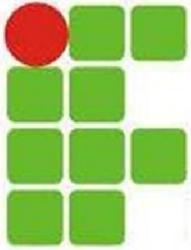




Collections

- Coleções são objetos do Java que podem agrupar vários outros objetos
- As Coleções possuem métodos para facilitar a programação
- Estão no pacote ***java.util*** por isso!
- Exemplos de Coleções:
 - Lists
 - Sets
 - Maps

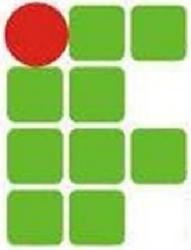




Collections

- Aprendemos a utilizar Arrays, então porque aprender Coleções?
- Coleções são mais fáceis de usar e já possuem métodos prontos de inserção, remoção, busca e outros
- Além disso, Coleções possuem tamanho dinâmico, enquanto Arrays não podem ter seu tamanho alterado, só isso já facilita muito a vida do programador!

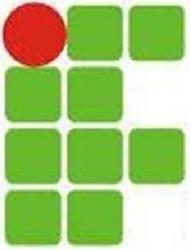




List

- Uma Lista é uma Interface de Coleção que permite elementos duplicados
- Listas mantêm uma ordenação específica dos elementos
- Já possui todos os métodos padrão implementados: busca, remoção, inserção...
- Sua principal implementação é o `ArrayList<tipo>`, onde “tipo” é o tipo de objeto que esta coleção vai conter





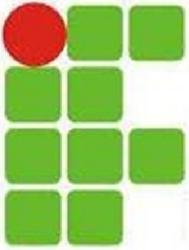
ArrayList

- Vejamos como é fácil criar um ArrayList de Strings, inserir nomes e imprimí-los:

```
public static void main(String[] args) {  
    ArrayList<String> nomes = new ArrayList<>();  
  
    nomes.add("Diego");  
    nomes.add("Adorilson");  
  
    for (String nome : nomes) {  
        System.out.println(nome);  
    }  
}
```

```
Output - POO-Exercicio16 (run) ✖  
run:  
Diego  
Adorilson
```





ArrayList

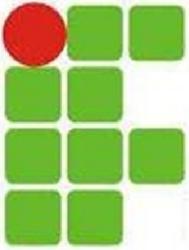
- Agora vejamos como remover um nome ou buscá-lo em uma ArrayList:

```
public static void main(String[] args) {  
    ArrayList<String> nomes = new ArrayList<>();  
    nomes.add("Diego");  
    nomes.add("Adorilson");  
    nomes.add("Nickerson");  
    nomes.add("Igor");  
    System.out.println(nomes.contains("Diego"));  
    nomes.remove("Diego");  
    System.out.println(nomes.contains("Diego"));  
}
```

Output - POO-Exercicio16 (run) ✖

```
run:  
true  
false
```

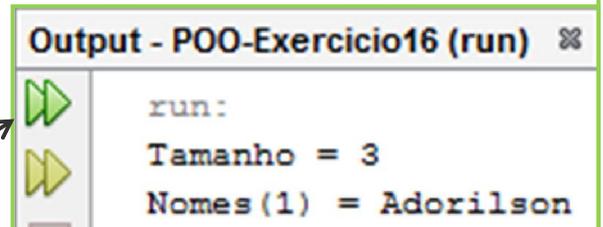


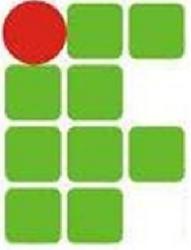


ArrayList

- Para saber o tamanho de um ArrayList utilizamos seu método `size()` e para resgatar um elemento do ArrayList em uma posição específica, utilizamos o método `get()`:

```
public static void main(String[] args) {  
    ArrayList<String> nomes = new ArrayList<>();  
    nomes.add("Diego");  
    nomes.add("Adorilson");  
    nomes.add("Nickerson");  
    System.out.println("Tamanho = " + nomes.size());  
    System.out.println("Nomes(1) = " + nomes.get(1));  
}
```





ArrayList

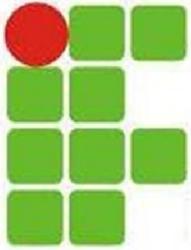
- Ordenando um ArrayList de nomes:

```
ArrayList<String> nomes = new ArrayList<>();  
nomes.add("Diego");  
nomes.add("Adorilson");  
nomes.add("Nickerson");  
nomes.add("Igor");  
nomes.add("Marcus");  
Collections.sort(nomes);  
  
for (String nome : nomes) {  
    System.out.println(nome);  
}
```

Output - POO-Exercicio16 (run)

```
run:  
Adorilson  
Diego  
Igor  
Marcus  
Nickerson
```



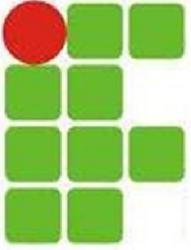


Set

- A a Interface Set representa um conjunto que não permite elementos duplicados, similar aos conjuntos da matemática
- O Set não garante que os elementos vão ser resgatados na ordem em que foram inseridos
- Suas principais implementações são:

- HashSet
- LinkedHashSet
- TreeSet

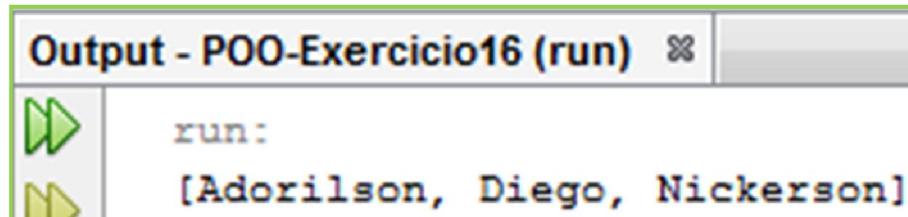




HashSet

- Exemplo de adição de elementos repetidos em um Set:

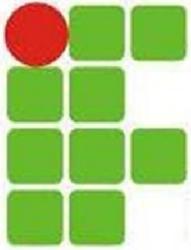
```
HashSet<String> nomes = new HashSet<>();  
nomes.add("Diego");  
nomes.add("Adorilson");  
nomes.add("Nickerson");  
nomes.add("Adorilson");  
nomes.add("Diego");  
System.out.println(nomes);
```



Output - POO-Exercicio16 (run) ✖

```
run:  
[Adorilson, Diego, Nickerson]
```



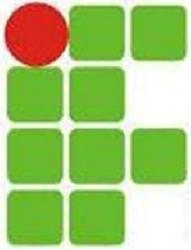


Map

- Um Map é composto por um conjunto de objetos e um valor chave para cada objeto
- Por exemplo um conjunto de objetos carro é representado pelas suas placas, assim fica mais fácil encontrar cada carro específico
- Diferente das outras Coleções, o método para inserir é o put(chave, objeto) e para resgatar é o get(chave)



- Suas principais implementações são o HashMap, TreeMap e HashTable



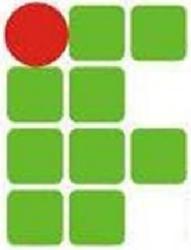
HashMap

- Criando um HashMap, adicionando elementos, resgatando e imprimindo:

```
HashMap<String, String> siglas = new HashMap<>();  
siglas.put("GNU", "GNU is Not Unix");  
siglas.put("GPL", "General Public License");  
siglas.put("FTP", "File Tranfer Protocol");  
  
System.out.println(siglas.get("FTP"));  
System.out.println(siglas.get("GPL"));  
System.out.println(siglas.get("GNU"));
```

```
Output - POO-Exercicio16 (run) ✖  
run:  
File Tranfer Protocol  
General Public License  
GNU is Not Unix
```



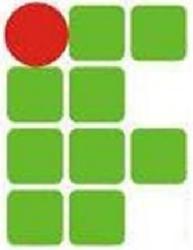


Exercício

- Desenvolva um sistema de cadastro de Aluno utilizando Coleções
 - O sistema deverá possuir as funções de:
 - Cadastrar
 - Remover
 - Atualizar
 - Pesquisar
- Aluno



- **Vale 1,0 ponto extra até a próxima aula!**



Perguntas?

