

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

COLEÇÕES DE OBJETOS - CONJUNTOS

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

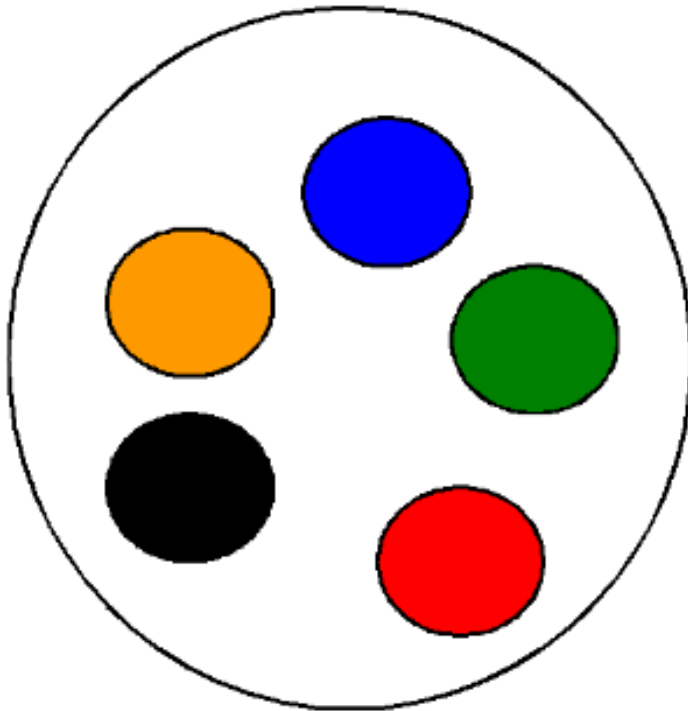
Introdução

2

- É uma estrutura que armazena um conjunto de objetos.
- E as listas também não eram a mesma coisa ?? Qual a diferença ??
- Os conjuntos (**Sets**) não contêm elementos duplicados.
- Eles também não possuem uma ordem definida, pode variar de acordo com a implementação.

Introdução

3



Possíveis ações em um conjunto:

- A camiseta Azul está no conjunto?
- Remova a camiseta Azul.
- Adicione a camiseta Vermelha.
- Limpe o conjunto.

- **Não existem elementos duplicados!**
- **Ao percorrer um conjunto, sua ordem não é conhecida!**

Conjuntos (Sets)

4

- Um conjunto possui um contrato com a interface Set. Logo, deve possuir um conjunto de comportamentos sugeridos por ela.
- A principal classe que utilizaremos para os conjuntos é a HashSet.
- Para criar um conjunto basta criar um objeto da classe citada.
 - `HashSet conj = new HashSet();`
 - OU
 - `Set conj = new HashSet();`

Conjuntos (Sets)

5

- ❑ Não possui métodos que utilizam índices como parâmetros.
- ❑ Principais métodos:
 - ❑ `add(Object o)`
 - ❑ `remove (Object o)`
 - ❑ `Contains (Object o)`
 - ❑ `size()`

Conjuntos (Sets)

6

```
Set cargos = new HashSet ();

cargos.add("Gerente");
cargos.add("Diretor");
cargos.add("Presidente");
cargos.add("Secretária");
cargos.add("Funcionário");
cargos.add("Diretor"); // repetido!

// imprime na tela todos os elementos
System.out.println(cargos);
```

Conjuntos (Sets)

7

- Lembrem do **Generics** ??
- Também podemos utilizar o **Generics** com conjuntos.

```
Set<String> cargos = new HashSet<>();
```

```
cargos.add("Gerente");
```

```
cargos.add("Diretor");
```

```
cargos.add("Presidente");
```

```
cargos.add("Secretária");
```

```
cargos.add("Funcionário");
```

```
cargos.add("Diretor"); // repetido!
```

```
// imprime na tela todos os elementos
```

```
System.out.println(cargos);
```

Conjuntos (Sets)

8

- E para percorrer um conjunto...

```
for (String s : l) {  
    System.out.println("Elemento: " + s);  
}
```


Exercício 01

9

1. Crie uma classe `TesteHashSetCPF` que possui um método **main**.
2. Dentro do `main` crie um `HashSet` de `String`.
3. Adicione 10 CPF (`Strings`) informados pelo usuário.
4. Digite alguns CPF repetidos e verifique o conteúdo do conjunto criado (`HashSet`) utilizando um *for* para percorrê-lo.