

UFPI - CCN - DC  
Ciência da Computação  
Estrutura de Dados

## Sets, Maps e Bags



Prof. Raimundo Moura  
[rsm@ufpi.edu.br](mailto:rsm@ufpi.edu.br)

60

### Conjunto (= set)

- ▶ É uma *estrutura de dados abstrata* que pode armazenar certos valores, sem qualquer *ordem* particular e sem valores repetidos.
- ▶ É uma implementação computacional do conceito *matemático* de um *conjunto finito*.
- ▶ Diferentemente da maioria dos outros tipos de coleções, em vez de recuperar um elemento específico de um conjunto, normalmente testa-se um valor para verificação se é membro do conjunto.
  - Queremos saber se:  $x \in \text{Conjunto}$

61

## Mapeamento (= map)

- ▶ É uma *estrutura de dados abstrata* capaz de associar uma chave a um valor.
- ▶ Os *maps* armazenam seus valores de forma *não ordenada*.
- ▶ Estrutura de um map:
  - $\text{map}(\text{chave}) \rightarrow \text{valor}$

62

## Saco(= bag) e sua API

- ▶ Um *saco(= bag)* é um ADT que consiste em uma coleção de coisas munida de duas operações: *add*, que insere uma coisa na coleção, e *iterate*, que percorre as coisas da coleção (**ou seja, examina as coisas uma a uma**).
- ▶ A ordem em que o iterador percorre as coisas não é especificada e está fora do controle do cliente.
- ▶ Como de的习惯, as coisas de que um saco é feito serão chamadas *itens*.
- ▶ Diferentemente de um conjunto matemático, os itens de um saco não precisam ser distintos dois a dois.

63

## Saco: API (= Interface)

```
public class Bag<Item> implements Iterable<Item>
```

<code>Bag()</code>	construtor: cria um saco de Itens vazio
<code>void add(Item item)</code>	coloca item neste saco
<code>Iterator&lt;Item&gt; iterator()</code>	um iterador que percorre os Itens do saco
<code>boolean isEmpty()</code>	este saco está vazio?
<code>int size()</code>	número de Itens neste saco

64

## Saco: API (= Interface)

- O método `iterator()` implementa a operação `iterate`: para processar, um a um, todos os itens de uma instância de saco (Bag), um cliente deve dizer:

```
Iterator<Item> it = saco.iterator();
while (it.hasNext()) {
    Item x = it.next();
    ... // processamento de x
}
```

- Felizmente, o cliente pode abreviar esse bloco de código por uma instrução muito simples conhecida como `foreach`:

```
for (Item x : saco) {
    ... // processamento de x
}
```

OBS: Leia para cada x no saco faça ....

65

## Saco: Implementação

- ▶ Exemplo de cliente de saco:
  - ▶ Uma sacola de nomes
  - ▶ Calcular a *média* e o *desvio padrão* de números dados na entrada padrão.
- ▶ Saco implementado em lista ligada
  - ▶ O saco bag armazena os dados, permitindo que eles sejam examinados mais de uma vez. Sem isso, o problema seria mais difícil de resolver.