Cap 06

Collections

Parte 3 – Conjuntos e Mapas

Linguagem de Programação III

Prof. Otacílio José Pereira

Objetivos e Tópicos

- Compreender as collections básicas ou mais tradicionais

Tópicos

- *Conjuntos (Interface Set)*
  - *Comparação Set e List*
  - *Implementações*
  - *Hashset*
  - *LinkedHashSet*
- *Mapas (Interface Map)*
  - *Comentários iniciais*
  - *Implementação HashMap*
  - *Exemplos*

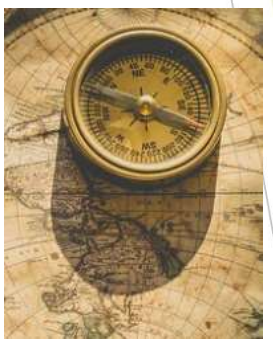
Contexto

- 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



Aula anterior: Exploração das collections básicas

- *Raciocínios importantes ao lidar com collections e estruturas de dados*
  - Collection/Estrutura de Dados: Estrutura + Operações
  - Visão de usuário (operações abstraindo codificação) e
    Visão interna (implementação da estrutura e das operações)
  - Aplicações interessantes para fixação
- Interface List e ArrayList, LinkedList e Vector
- Stack
- Queue

Foco agora:

- **Continuar exploração mas com ênfase para Sets**
- Características principais e diferença em relação a List
- Interface com suas operações
- Implementações

```
public static void main(String args[])
{
    // Variáveis e seus tipos
    Set<String> eset = new HashSet<String>();
    // Principais operações
    eset.add("Fabiana");
    eset.add("Carlos");
    eset.add("Cristina");
    eset.add("Rita");
    eset.add("Flávia");
    eset.add("José");
    // Verificando as duplicatas
    eset.add("Carlos");
    eset.add("Rita");

    System.out.println("--");
    System.out.println("-- Resultado 1 --");
    System.out.println("--");
    for (String s : eset)
    {
        System.out.print(s + "   ");
    }
```

Sets

- Características
- Interface Set

Interface Set

- É uma abstração com conceito matemático de conjunto
- O principal ponto é que um Set não aceita duplicatas
- Paralelo ou comparação entre List e Set
  - List aceita duplicatas e é fortemente baseada no conceito de acessar um elemento pela sua posição
  - Set não aceita duplicatas e visa

Interface Set

- Operações
  - As operações apresentam certa semelhança em relação às outras Collections
  - Uma diferença perceptível, sobre a questão de não ser uma estrutura para recuperação em posições, é que existe a operação contains em vez de uma operação get por exemplo.

Operação	Descrição
add	Adiciona um elemento desde que não seja duplicata
addAll	Adiciona uma collection em outra
clear	Limpa a collection
isEmpty	Verifica se está vazia, assim como em outras experiências
remove	Remove certo elemento
removeAll	Remove uma collection

Exemplo

```
public static void main(String args[])
{
    // Variáveis e seus tipos
    Set<String> eset = new HashSet<String>();
    // Principais operações
    eset.add("Fabiana");
    eset.add("Carlos");
    eset.add("Cristina");
    eset.add("Rita");
    eset.add("Flávia");
    eset.add("José");
    // Verificando as duplicatas
    eset.add("Carlos");
    eset.add("Rita");

    System.out.println("--");
    System.out.println("-- Resultado 1 --");
    System.out.println("--");
    for (String s : eset)
    {
        System.out.print(s + "   ");
    }
    System.out.println();
    System.out.println("-- Acessando elementos --");
    System.out.println("[Flávia] -> " + eset.contains("Flávia"));
    System.out.println("[Carlos] -> " + eset.contains("Carlos"));
    System.out.println("[João] -> " + eset.contains("João"));
}
```

- O que será impresso pelo programa a seguir?

- Curiosidade, ao inserir em um set, no caso específico do Hashset, não há garantia quando à ordem.
- Outros sets podem ser empregados, por exemplo o TreeSet

Exemplo 2 : Curiosidade

```
// Variáveis e seus tipos
List<String> lst = new ArrayList<String>();

// Principais operações
lst.add("Rita");
lst.add("Fabiana");
lst.add("Carlos");
lst.add("Flávia");
lst.add("Cristina");
lst.add("Rita");
lst.add("Flávia");
lst.add("José");
lst.add("Carlos");
```

```
// Retirando as duplicatas com o set
Set<String> eset = new HashSet<String>(lst);

System.out.println("--");
System.out.println("-- Resultado 2 (Set) --");
System.out.println("-- (Há duplicatas?)");
for (String s : eset)
{
    System.out.print(s + "  ");
}
```

- Perceba que primeiro foi usado uma Lista
- Depois foi criada um Set com base da lista criada
- Este é um exemplo interessante pois trata especificamente da distinção entre Lista e Set

```
System.out.println();
System.out.println("-- Acessando elementos --");
System.out.println("[2] -> " + lst.get(2));
System.out.println("[4] -> " + lst.get(4));
```

```
System.out.println();
System.out.println("-- Acessando elementos --");
System.out.println("[Flávia] -> " + eset.contains("Flávia"));
System.out.println("[Carlos] -> " + eset.contains("Carlos"));
System.out.println("[João] -> " + eset.contains("João"));
```

Implementações: HashSet e HashLinkedset

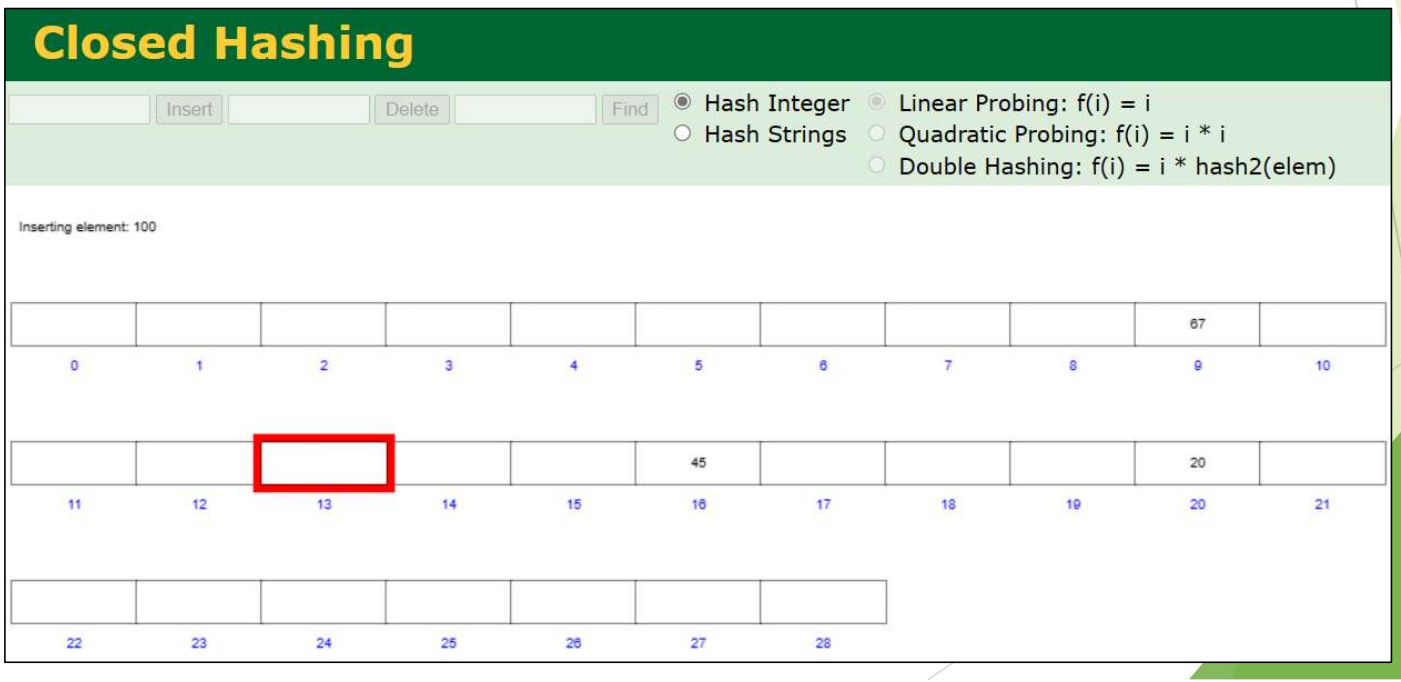
- Internamente os sets funcionam com o conceito de Tabelas de Hash (espalhamento) ou Tabelas Hash com listas encadeadas

- Explicando brevemente, no caso cada objeto que for inserido tem um valor de hash calculado que determina a posição do elemento na tabela.

- Por exemplo, imagine que uma função de hash é a soma de todos os elementos divididos por um certo valor
  - ABC = 65+66+67 = 198 = 198 / 11 = Quociente 18 e resto 0
  - O registro é guardado na posição 0

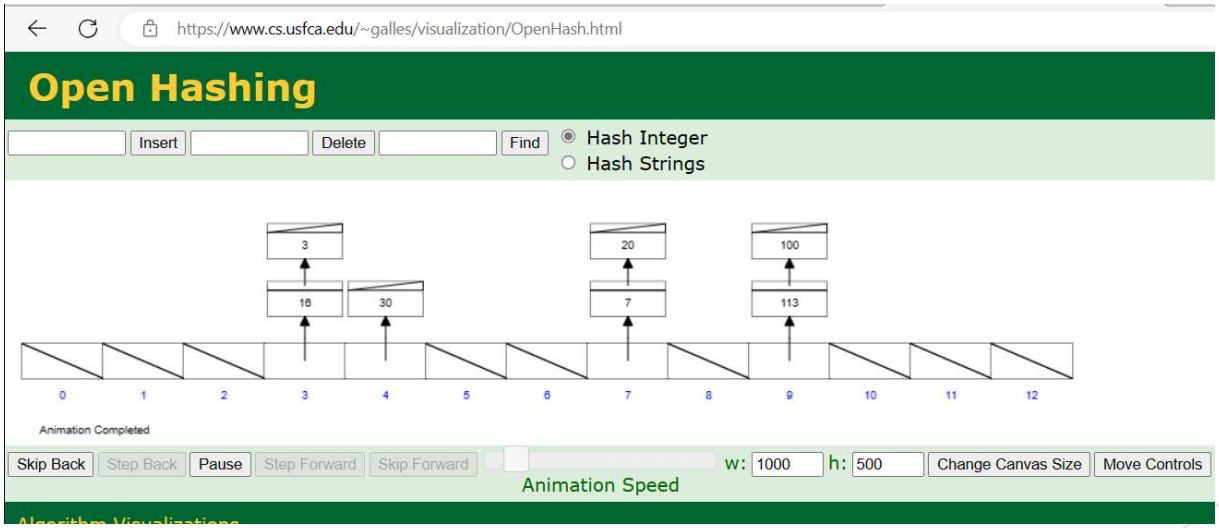
Implementações: HashSet

- Caso queira entender mais a respeito (Equivalente ao HashSet)
  - Closed Hash / Hash via tabela / Tabela de dispersão
  - https://www.cs.usfca.edu/~galles/visualization/ClosedHash.html



Implementações: HashLinkedset

- Caso queira entender mais a respeito (Equivalente ao HashLinkedSet)
  - Open Hash / Hash via tabela com lista encadeada
  - https://www.cs.usfca.edu/~galles/visualization/OpenHash.html



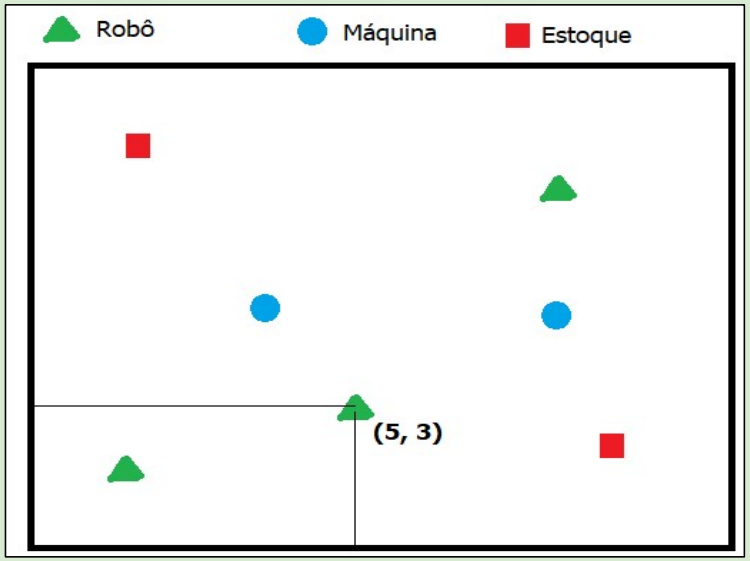
  - Outra dica de ambiente: www.visualgo.net

Implementações: HashSet e HashLinkedset

- HashSet
  - Apresenta uma excelente performance para acessar os elementos
  - Entretanto não consegue garantir a ordem dos elementos

- HashLinkedSet
  - A performance é um pouco pior que HashSet
  - Pode ser acessado na ordem em que foram inseridos

- TreeSet
  - Utiliza um esquema de árvore internamente
  - Árvore rubro-negra
  - Os elementos podem ser percorridos em ordem



Exercício sobre collections

- Posicionamento de elementos

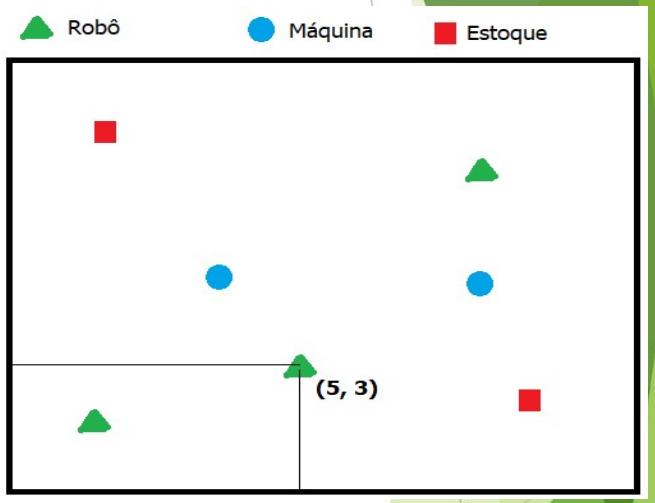
Exercício

- Faça um programa que registra os equipamentos que estão localizados em uma determinada indústria

- Os equipamentos podem ser máquinas, paredes e outros elementos em uma indústria

- A indústria é organizada em quadrantes, cada equipamento só pode ocupar uma tupla (x, y)

- Faça um programa que registra estes obstáculos

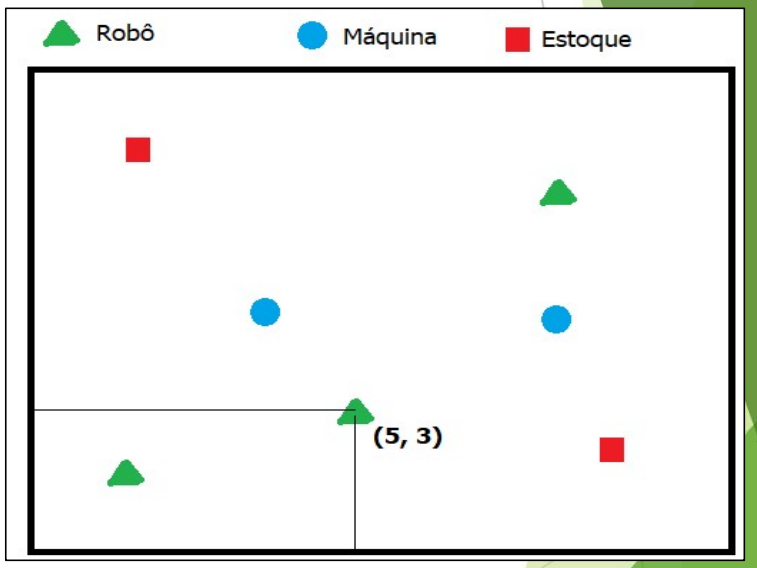


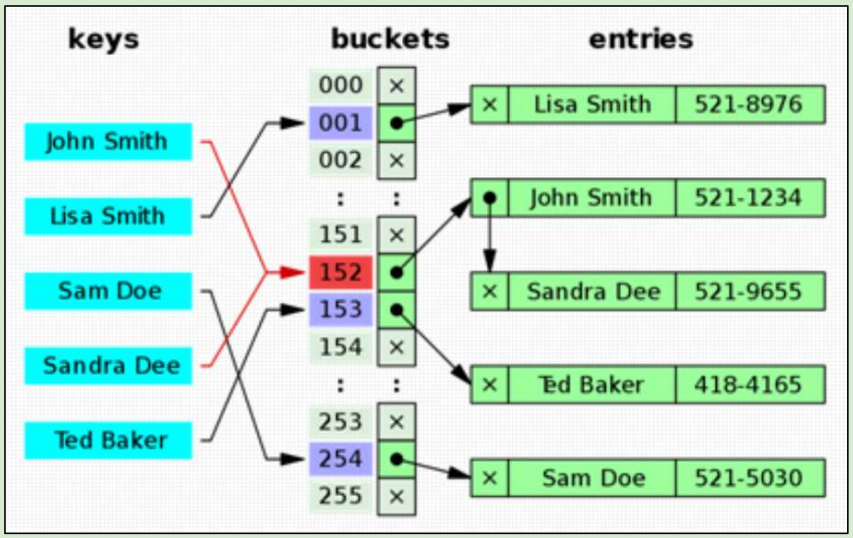
Sugestão de passos

- Criar uma classe Posição
  - X, Y, tipo

- Crie o programa que registre o conjunto de posições

- Qual a diferença de usar ArrayList e HashSet?



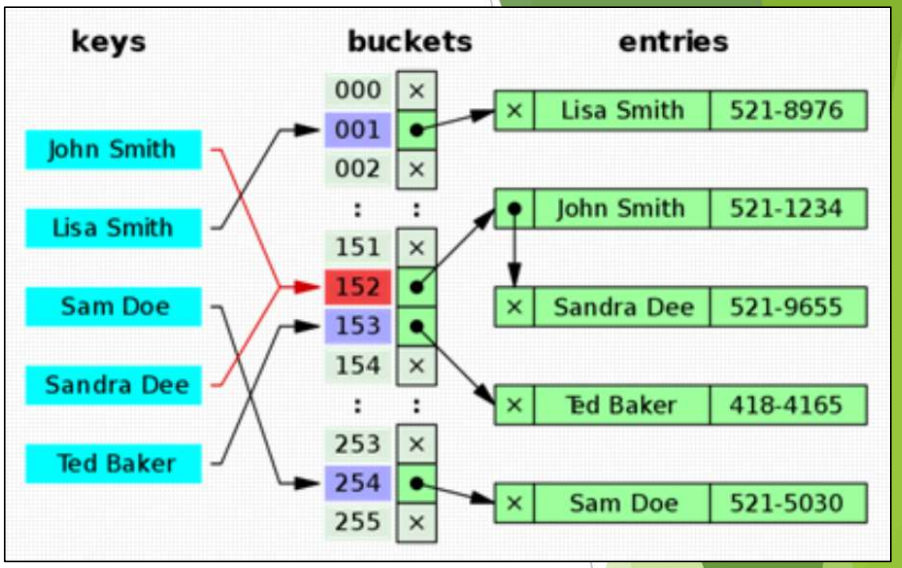


Mapas

- Associação Chave-Valor
- Implementação com Hash
- Exemplos

Mapas

- Um mapa permite criar uma associação Chave – Valor.
- As chaves funcionam como uma entrada em um índice. Isso pode melhorar a performance na recuperação de dados

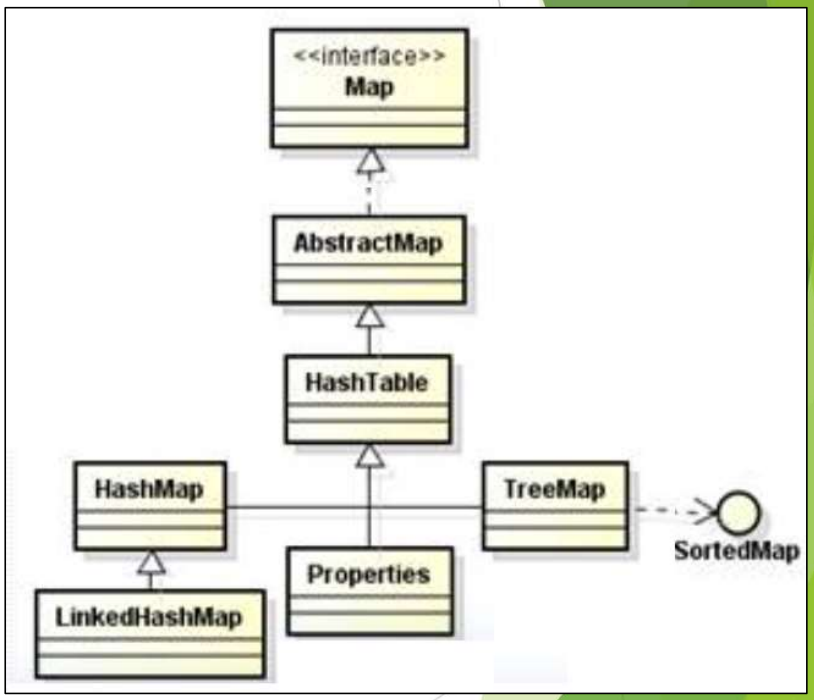


- Veja o exemplo ao lado
  - A chave para a busca por pessoas na base de dados ocorre por Nome (key)
  - Conforme a chave e um cálculo de hash, encontra-se a posição do registro e daí tem-se o registro completo(valor)

  - Veja artigo : https://www.devmedia.com.br/conhecendo-a-interface-map-do-java/37463
  - A figura foi retirada de lá e apresenta outras informações interessantes sobre mapas

Mapas

- Com a exceção da questão da tupla Chave-Valor, as implementações de Mapas tem certa equivalência com as implementações de Sets
    - HashMap
    - LinkedHashMap
    - TreeMap



Mapas (Map)

- Algumas operações

Operação	Descrição
put	Adiciona um par Chave, Valor no mapa Exemplo: emap.put(ID, Nome);
get	Recupera o valor equivalente à chave consultada Exemplo: emap.get(key);
keySet	Recupera o conjunto de chaves no mapa
containsKey	Verifica se existe uma chave no mapa

Exemplo

```
public static void main(String args[])
{     // Variáveis e seus tipos
      Map<Integer, String> emap = new HashMap<Integer, String>();

      int ID;
      String Nome;

      ID = 10;
      Nome = "João";

      if (emap.containsKey(ID))
      { System.out.print("Erro: CPF já cadastrado.");        }
      else
      { emap.put(ID, Nome);  }

      // Apenas para outros exemplos
      ID = 21;  Nome = "Maria";  emap.put(ID, Nome);
      ID = 30;  Nome = "Samuel"; emap.put(ID, Nome);
```

- Pontos de atenção
  - Key : CPF - Valor: String
  - containKey
  - Put e get
  - Uso do Iterator em keySet

```
Set<Integer> keys = emap.keySet();
Iterator<Integer> it = keys.iterator();
int key; String valor;
while (it.hasNext())
{
    key = it.next();
    valor = emap.get(key);
    System.out.println(" Key: " + key + " - Valor: " + valor);
}
```

Exercício

- No livro do Deitel, na parte de mapas (Tópico 16.11) existe uma implementação bem interessante de um contador de repetições em texto usando Mapas e Conjuntos e com ordenação (TreeSet)

- Refaça aquela implementação discutindo o uso das collections

- Ao lado está uma parte da implementação

```
// exibe conteúdo do mapa
private static void displayMap(Map<String, Integer> map)
{
   Set<String> keys = map.keySet(); // obtém chaves

   // classifica as chaves
   TreeSet<String> sortedKeys = new TreeSet<>(keys);

   System.out.printf("%nMap contains:%nKey\t\tValue%n");

   // gera saída de cada chave no mapa
   for (String key : sortedKeys)
      System.out.printf("%-10s%10s%n", key, map.get(key));

   System.out.printf(
      "%nsize: %d%nisEmpty: %b%n", map.size(),map.isEmpty());
}
```

Conclusões

Quais eram os objetivos?

- Compreender as collections básicas ou mais tradicionais

Tópicos

- *Conjuntos (Interface Set)*
  - *Comparação Set e List*
  - *Implementações*
  - *Hashset*
  - *LinkedHashSet*
- *Mapas (Interface Map)*
  - *Comentários iniciais*
  - *Implementação HashMap*
  - *Exemplos*