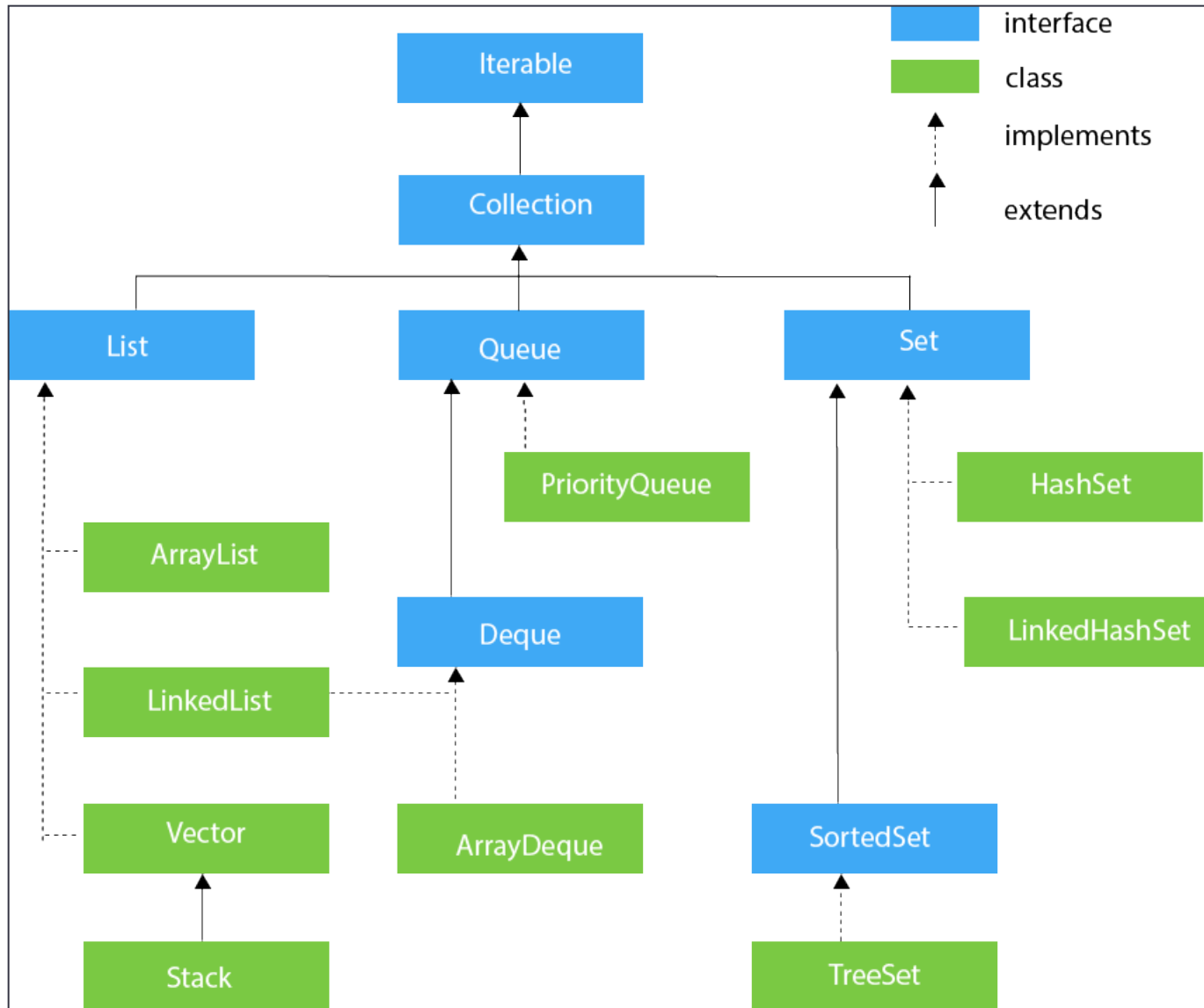


INTRODUZIONE A JAVA COLLECTIONS – INSIEMI E DIZIONARI

Angelo Di Iorio

Università di Bologna

Java Collections Core



Insiemi

- Un insieme (set, in inglese) è una **collezione** di **elementi distinti** dello stesso tipo
- Possibile interfaccia di un insieme generico di elementi:
 - `public boolean contains(T x);`
 - `public void insert(T x);`
 - `public void remove(T x);`
 - `public Set<T> union(Set<T> B);`
 - `public Set<T> intersect(Set<T> B);`
 - `public Set<T> difference(Set<T> B);`

Set Interface in Java

Method Summary

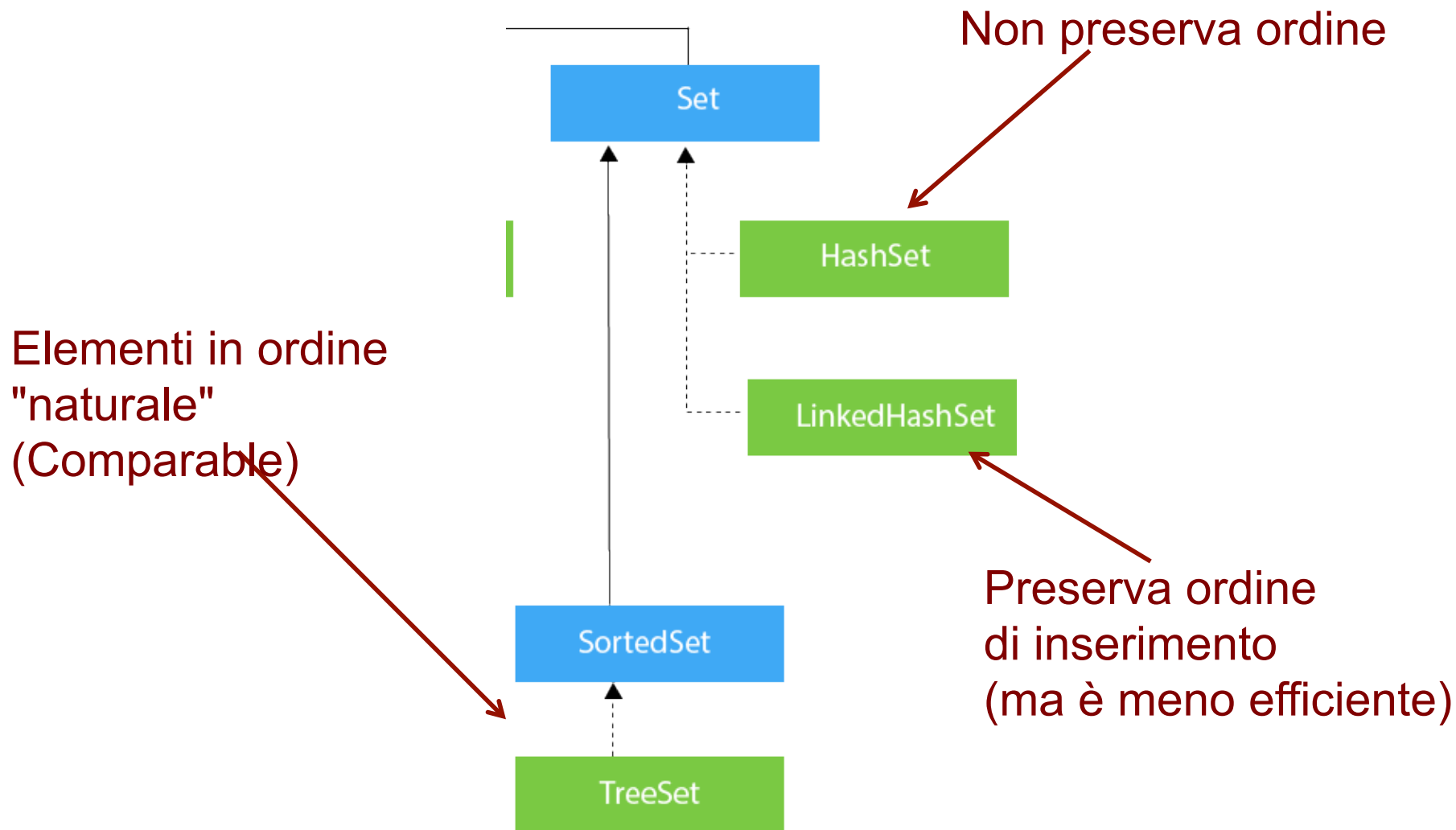
<code>boolean</code>	<code>add(Object o)</code> Adds the specified element to this set if it is not already present (optional operation).
<code>boolean</code>	<code>addAll(Collection c)</code> Adds all of the elements in the specified collection to this set if they're not already present (optional operation).
<code>void</code>	<code>clear()</code> Removes all of the elements from this set (optional operation).
<code>boolean</code>	<code>contains(Object o)</code> Returns true if this set contains the specified element.
<code>boolean</code>	<code>containsAll(Collection c)</code> Returns true if this set contains all of the elements of the specified collection.
<code>boolean</code>	<code>equals(Object o)</code> Compares the specified object with this set for equality.
<code>int</code>	<code>hashCode()</code> Returns the hash code value for this set.
<code>boolean</code>	<code>isEmpty()</code> Returns true if this set contains no elements.
<code>Iterator</code>	<code>iterator()</code> Returns an iterator over the elements in this set.

Set Interface in Java

<code>boolean</code>	<code>remove(Object o)</code> Removes the specified element from this set if it is present (optional operation).
<code>boolean</code>	<code>removeAll(Collection c)</code> Removes from this set all of its elements that are contained in the specified collection (optional operation).
<code>boolean</code>	<code>retainAll(Collection c)</code> Retains only the elements in this set that are contained in the specified collection (optional operation).
<code>int</code>	<code>size()</code> Returns the number of elements in this set (its cardinality).
<code>Object[]</code>	<code>toArray()</code> Returns an array containing all of the elements in this set.
<code>Object[]</code>	<code>toArray(Object[] a)</code> Returns an array containing all of the elements in this set whose runtime type is that of the specified array.

Le classi `TreeSet<T>` e `HashSet<T>` implementano l'interfaccia `Set<T>`

HashSet e TreeSet



```
Book b1 = new Book("Il nome della Rosa", 1980);  
Book b2 = new Book("The Da Vinci Code", 2003);  
Book b3 = new Book("Outliers", 2008);  
Book b4 = new Book("The Client", 1993);  
Book b5 = new Book("The Broker", 2005);
```

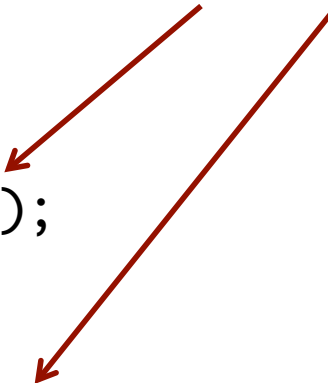
```
HashSet<Book> groupA = new HashSet<Book>();  
groupA.add(b1);  
groupA.add(b2);  
groupA.add(b3);
```

```
HashSet<Book> groupB = new HashSet<Book>();  
groupB.add(b3);  
groupB.add(b4);  
groupB.add(b5);
```

```
groupA.retainAll(groupB);  
System.out.println(groupA);
```

```
groupA.addAll(groupB);  
System.out.println(groupB);
```

Cosa stampano?



Esercizio

- Dato il precedente esempio, scrivere un metodo che prende in input un insieme di libri S e intero Y e restituisce l'insieme di libri pubblicati dopo l'anno Y (escluso)

Esercizi

- A. Scrivere un metodo che prende in input un insieme di parole (`Set<String>`) ed elimina dall'insieme le parole che iniziano per vocale
- B. Aggiungere un metodo che prende in input un insieme di stringhe e restituisce un altro insieme con le stringhe in ordine alfabetico

Esercizio

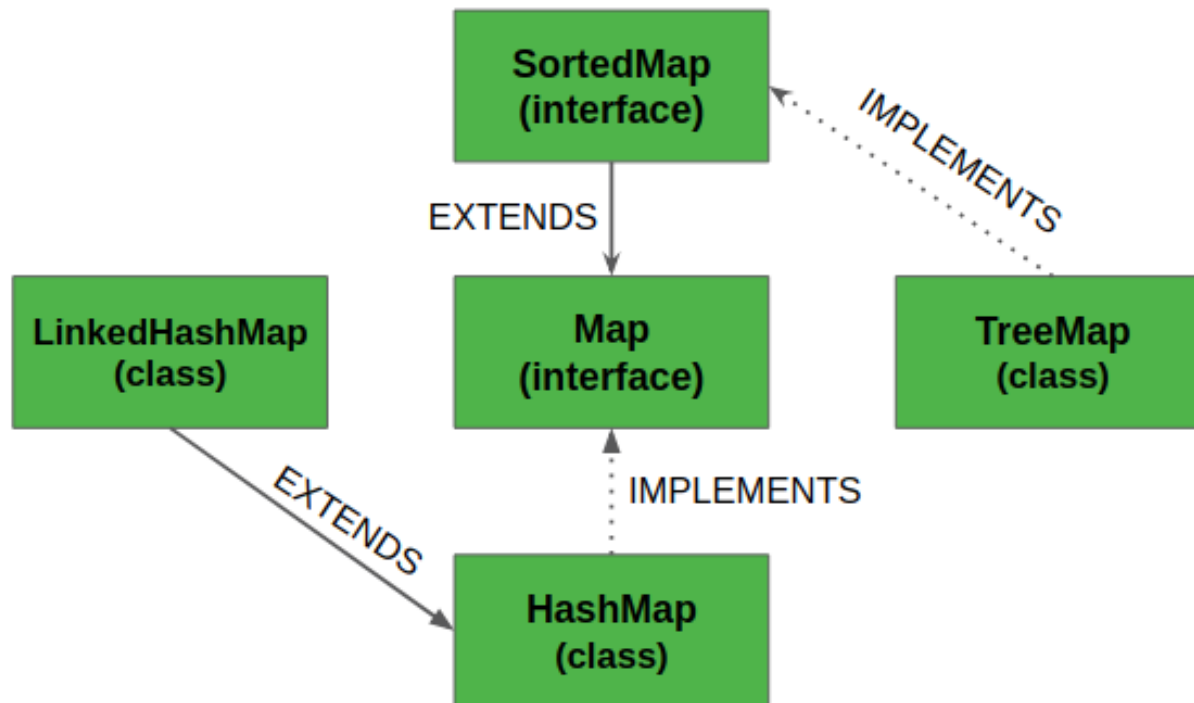
Scrivere un metodo che prende in input due insiemi di parole (`Set<String>`) e restituisce l'insieme delle lettere iniziali di queste parole (`Set<Character>`)

- Senza duplicati, in maiuscolo e in ordine alfabetico

Dizionari

- Un dizionario (*Map* in inglese) è un insieme di associazioni *chiave-valore*
- Su un dizionario è possibile:
 - ottenere il valore associato ad una particolare chiave (se presente) o `null` altrimenti
 - inserire una nuova associazione chiave-valore, cancellando eventuali associazioni precedenti
 - rimuovere un'associazione chiave-valore esistente
- L'interfaccia `Map<K, V>` in Java definisce le operazioni possibili su un dizionario ed è implementata dalle classi `TreeMap<K, V>` e `HashMap<K, V>`

Map Interface in Java



MAP Hierarchy in Java

Method Summary

void	<u>clear</u> () Removes all mappings from this map (optional operation).
boolean	<u>containsKey</u> (Object key) Returns true if this map contains a mapping for the specified key.
boolean	<u>containsValue</u> (Object value) Returns true if this map maps one or more keys to this value.
abstract Set	<u>entrySet</u> () Returns a set view of the mappings contained in this map.
boolean	<u>equals</u> (Object o) Compares the specified object with this map for equality.
Object	<u>get</u> (Object key) Returns the value to which this map maps the specified key.
int	<u>hashCode</u> () Returns the hash code value for this map.
boolean	<u>isEmpty</u> () Returns true if this map contains no key-value mappings.
Set	<u>keySet</u> () Returns a Set view of the keys contained in this map.
Object	<u>put</u> (Object key, Object value) Associates the specified value with the specified key in this map (optional operation).
void	<u>putAll</u> (Map t) Copies all of the mappings from the specified map to this map (optional operation).
Object	<u>remove</u> (Object key) Removes the mapping for this key from this map if present (optional operation).
int	<u>size</u> () Returns the number of key-value mappings in this map.
String	<u>toString</u> () Returns a string representation of this map.
Collection	<u>values</u> () Returns a collection view of the values contained in this map.

Esempio HashMap

```
Book b1 = new Book("Il nome della Rosa");  
Book b2 = new Book("The Da Vinci Code");  
Book b3 = new Book("Outliers");
```

```
HashMap<Book, Integer> copie = new HashMap<Book, Integer>();
```

```
copie.put(b1, 1300);  
copie.put(b2, 2000);  
copie.put(b3, 5000);
```

```
copie.put(b1, copie.get(b1) + 3000);  
copie.put(b2, copie.get(b2) + 2000);
```

```
for (Book b: copie.keySet()) {  
    System.out.println(b + ":" + copie.get(b));  
}
```

Cosa stampa
questo ciclo?



Java Collection Framework

- General-purpose implementations:

Interface	Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Deque		ArrayDeque		LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

Esercizio

- Scrivere un programma che prende in input una stringa da linea di comando (`Scanner`) e conta il numero di parole per ognuna delle possibili lunghezze
 - Ammesso usare metodi ausiliari
- Una parola è una sequenza di lettere maiuscolo o minuscole, esclusi spazi e punteggiatura
 - `scanner.useDelimiter("[^a-zA-Z]+");`
- **Input:** "Ciao, usiamo un dizionario in Java"
- **Output:**

Parole di 2 lettere: 2

Parole di 4 lettere: 2

Parole di 6 lettere: 1

Parole di 10 lettere: 1

Esercizio

- Aggiungere un metodo che elenca anche l'insieme di parole per ogni lunghezza
- **Input:** "Ciao, usiamo un dizionario in Java"
- **Output:**

Parole di 2 lettere: 2 - [un, in]

Parole di 4 lettere: 2 - [Java, Ciao]

Parole di 6 lettere: 1 - [usiamo]

Parole di 10 lettere: 1 - [dizionario]