



Java

TDDC77

Objektorienterad Programmering

Föreläsning 11

Sahand Sadjadee
IDA, Linköpings Universitet

Outline

- **Collections(Kollektioner) (18.1, 18.2, 18.3, 18.4, 18.5, 18.7)**

Collections(Kollektioner)

Att lagra värden

- En variabel: Data av en sort
- Problem: om det blir många variabler
- En array: En rad av data av samma sort
- Problem: Måste kunna storleken i förväg, även svårt att bearbeta data i arrayen (söka, sortera osv)

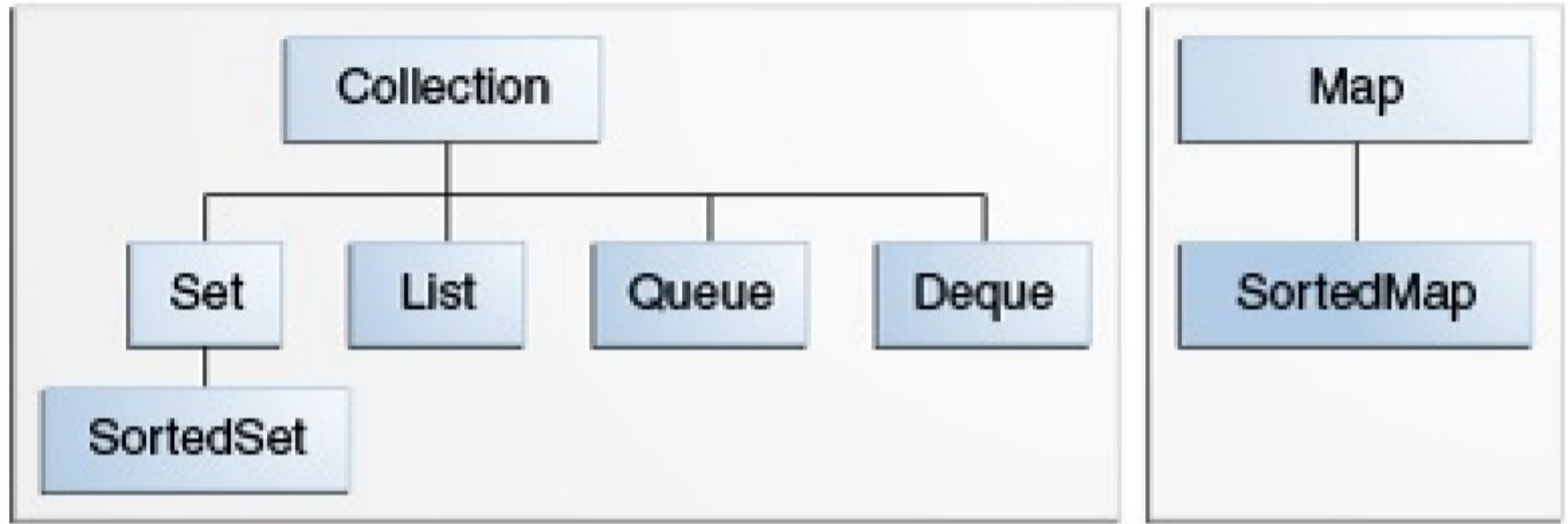
Kollektion (Collection)

- är ett objekt som grupperar flera element i en enda enhet.
- används för att lagra, hämta, manipulera, och kommunicera aggregerad data.
- bildar en naturlig grupp såsom en mapp (en samling av filer) eller en telefonkatalog (en kartläggning av namn till telefonnummer).

Lösningen heter Collections

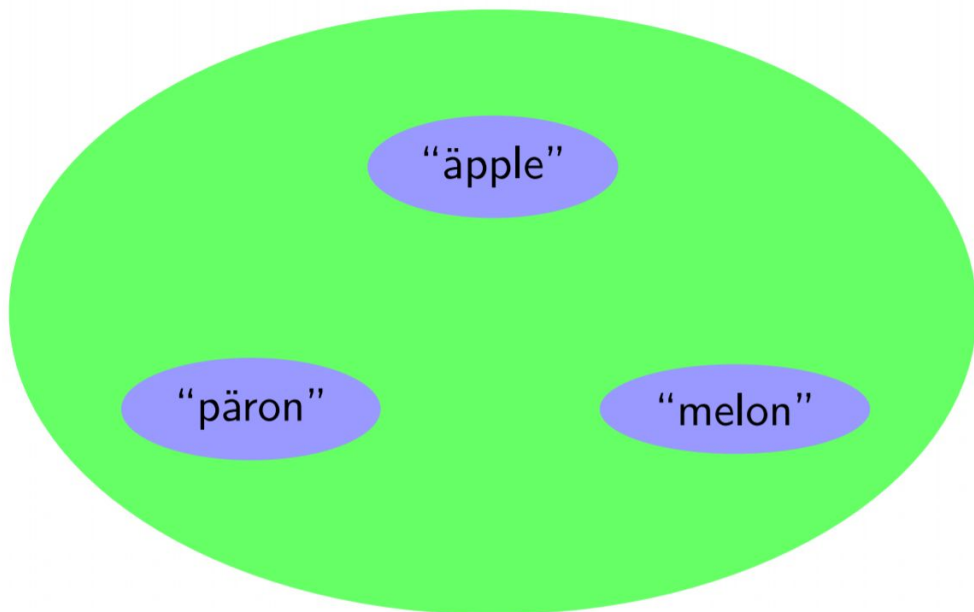
- Samling av klasser för effektiv och avancerad lagring av data
- Finns i paketet `java.util`
- Är uppdelade i interface och klasser
- Sök efter “Collection” i API:et
- <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>
- Finns även bra tutorials, ex <http://docs.oracle.com/javase/tutorial/collections>

Lösningen heter Collection



Set - Mängd

- osorterad mängd av data utan kopior
- Interfacet heter `Set`
- Implementationer: exempelvis `HashSet`, `SortedSet`



Set

```
import java.util.*;

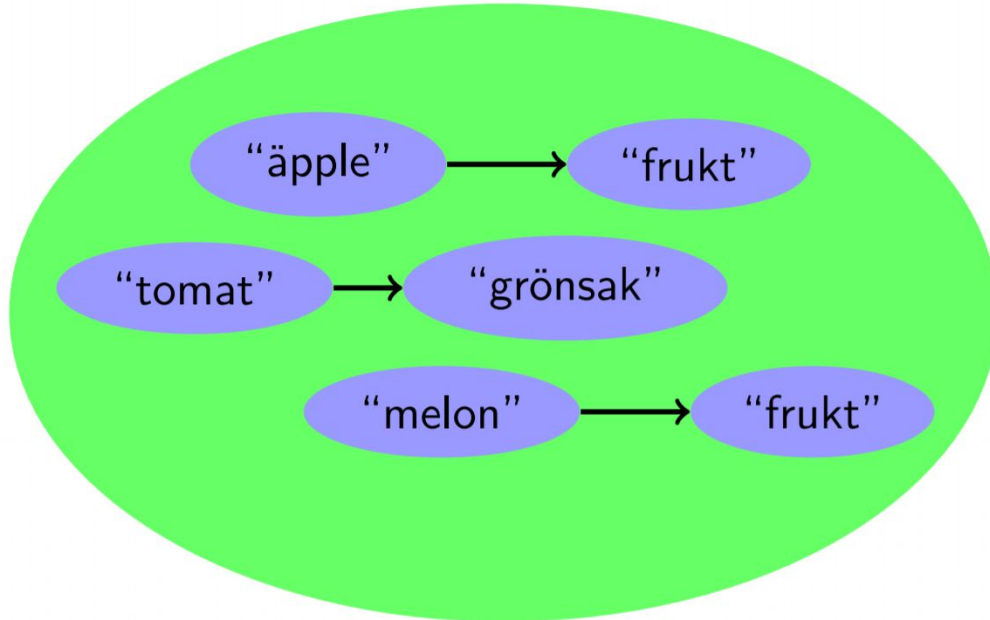
/* The program FindDuplicates prints messages for each duplicated
 * argument it is passed.
 * example: java FindDuplicates i came i saw i left
 */
class FindDuplicates {
    public static void main(String[] args) {
        Set<String> minSet = new HashSet<String>();
        for (int i=0; i<args.length; i++)
            if (minSet.add(args[i]) == false)
                System.out.println("Duplicate detected: " + args[i]);

        System.out.println(minSet.size() + " distinct words: " + minSet);
    }
}

// for(String str: args)
```

Map

- par av data, key -> value.
- Interfacet heter `Map`
- Implementationer: exempelvis `HashMap`



Map

```
import java.util.*;

public class Freq {
    public static void main(String[] args) {
        Map<String, Integer> m = new HashMap<String, Integer>();

        // Initialize frequency table from command line
        for (String a : args) {
            Integer freq = m.get(a);
            m.put(a, (freq == null) ? 1 : freq + 1);
        }

        System.out.println(m.size() + " distinct words:");
        System.out.println(m);
    }
}

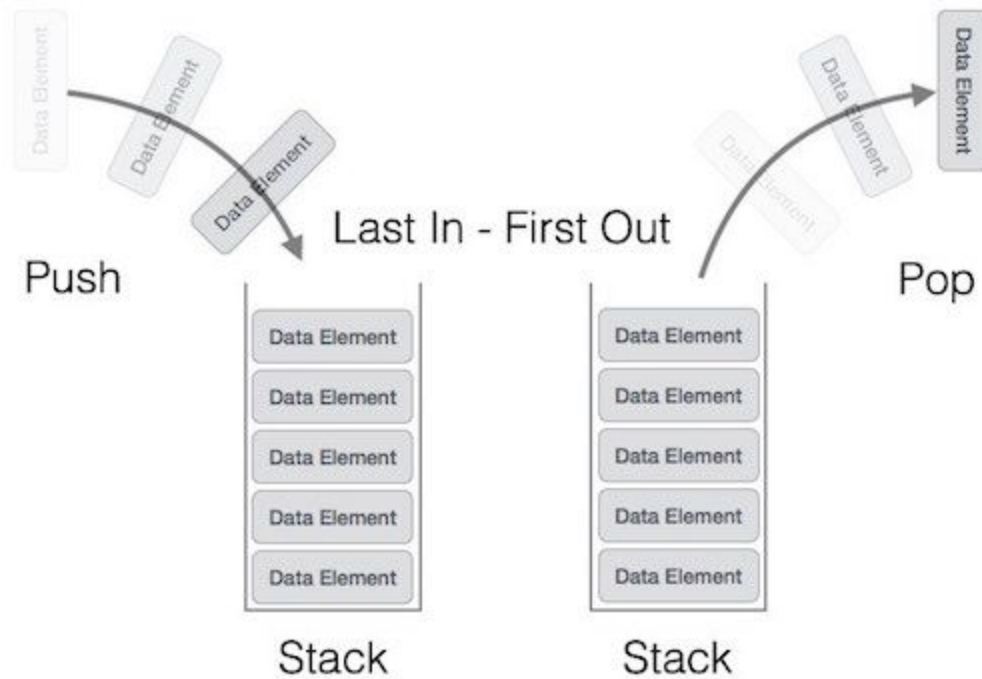
//java Freq if it is to be it is up to me to delegate
```



Klassen Collections

- Klassen Collections har statiska metoder för att manipulera listor, mängder osv.
 - `frequency()`
 - `min()`, `max()`
 - `reverse()`
 - `sort()`, `shuffle()`

Stack



Stack Operations:

- push(): Insert a new element into the stack i.e just insert a new element at the beginning of the linked list.
- pop(): Return the top element of the Stack i.e simply delete the first element from the linked list.
- peek(): Return the top element.

Klassen *LinkedList* innehåller alla metoder som behövs för att kunna använda den som en stack. Klassen *LinkedList* kan också användas som en vanlig lista.

<https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>

Queue



Queue

Queue Operations:

- `add()` adds an item to the queue.
- `remove()` removes and returns an item from the queue.
- `peek()` gets the element at the front of the queue without removing it.
- `isfull()` – Checks if the queue is full.
- `isempty()` – Checks if the queue is empty.

Klassen ***LinkedList*** innehåller alla metoder som behövs för att kunna använda den som en queue.

<https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>



Thanks for listening!