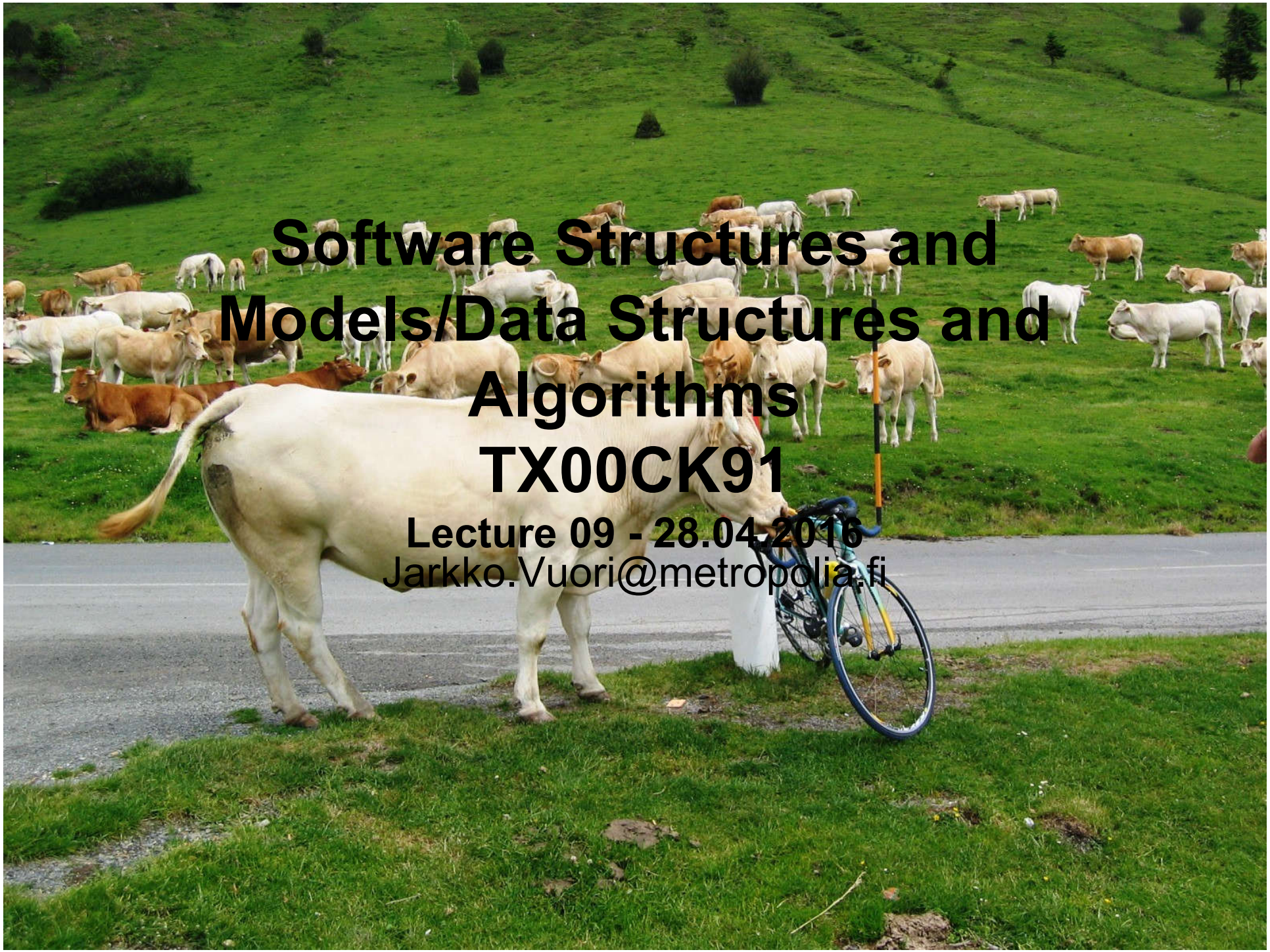


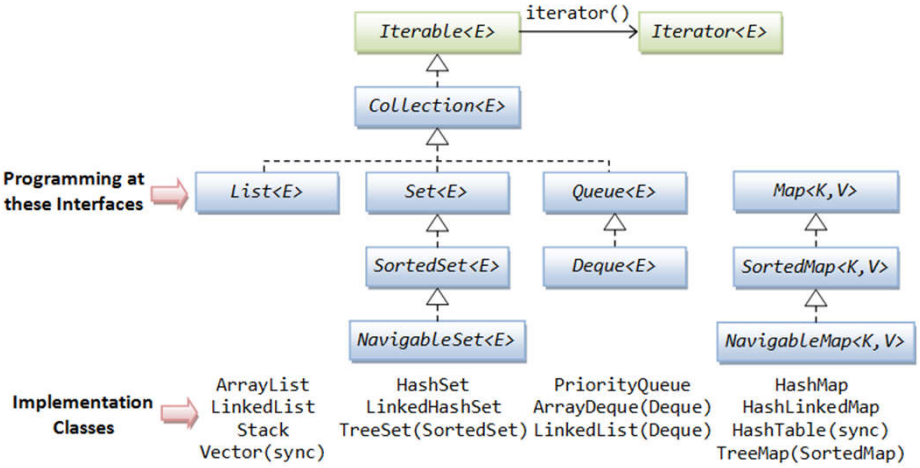
Software Structures and Models/Data Structures and Algorithms
TX00CK91

Lecture 09 - 28.04.2016

Jarkko.Vuori@metropolia.fi

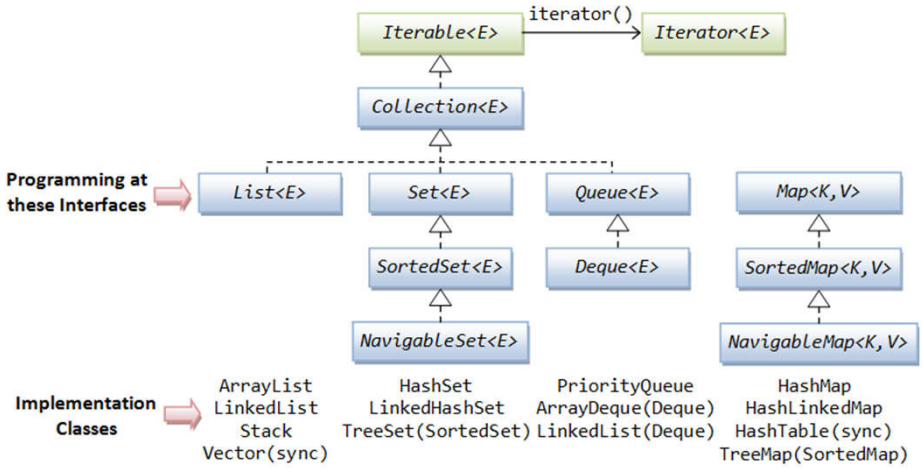
Java collection

- Java collection (java.util) abstract data structures
  - List
    - maintains order and allows duplicates
  - Set
    - does contain no elements twice
  - Map
    - Has a key to access its elements
    - Similar to linear arrays, except that an array uses an integer key to index and access its elements; whereas a map uses any arbitrary key (such as Strings or any objects)
- You as a developer are free to choose which implementation of these data structures are best suited for the kind of data you deal with (HashSet vs. TreeSet / LinkedList vs. ArrayList / etc.)
  - For example for Sets and Maps you may choose between hash-based implementations and tree-based implementations, which may or may not be suited for what you want to do
    - in most cases a hash-based implementation will be the best choice, while sometimes, when order is important, a tree may be better suited for your needs
- Note that there are no graph data structure available, but there are plenty of choices for Graph libraries, e.g. GraphT



Java Collection Tree

- Dynamic tree in Java collections is always binary search tree (Sorted Set of Map)
- Dynamic (search) tree is available at two different interfaces
  - Set
  - Map
- We'll study TreeSet here



Methods in Set

Method	Description
void add(Object o)	Adds the specified element to this set if it is not already present
void clear()	Removes all of the elements from this set
boolean contains(Object o)	Returns true if this set contains the specified element
Object first()	Returns the first (lowest) element currently in this sorted set
boolean isEmpty()	Returns true if this set contains no elements
Object last()	Returns the last (highest) element currently in this sorted set
boolean remove(Object o)	Removes the specified element from this set if it is present
int size()	Returns the number of elements in this set (its cardinality)

TreeSet Demo

- Program on the right prints to the console:

```
[A, B, C, D, E, F]
```

```
import java.util.*;

public class TreeSetDemo {

   public static void main(String args[]) {
      // Create a tree set
      TreeSet ts = new TreeSet();
      // Add elements to the tree set
      ts.add("C");
      ts.add("A");
      ts.add("B");
      ts.add("E");
      ts.add("F");
      ts.add("D");
      System.out.println(ts);
   }
}
```