Data Structures and Algorithms

XMUT-COMP 103 - 2020 T1

Lecture by : Dr Karsten Lundqvist

School of Engineering and Computer Science

Victoria University of Wellington

Course Structure - Lecture 02

Topics:

- Programming with unstructured and linear collections
- Different Kinds of collections:
  - Lists, Sets, Bags, Maps, Stacks, Queues, Priority Queues
- Algorithms using collections.
- Complexity
- Recursion
- Programming with Linked collections
  - Building, traversing tree structured collections
  - ? Building, traversing graph/network structured collections
- More complex algorithms

Recurring Themes

- Good Design:
  - Which choices should you make?
  - Choosing appropriate implementations for collections
  - Making the right choice the first time

- Efficiency:
  - How fast is it?
  - How much memory does it take?
  - By analysis, and by benchmarking

- Testing:
  - Does it work right?

Programming with Libraries

- Modern programs (especially GUI and network) are too big to build from scratch.

  ⇒ Have to reuse code written by other people

- Libraries are collections of code designed for reuse.
  - Java has a huge collection of standard libraries….
  - Learning to use libraries is essential

Libraries for COMP 103

- ecs100 UI, etc
- java.io Classes for dealing with files
- java.util Collection classes (you've used ArrayList, now …….)
  Other utility classes

- We will use these libraries in almost every program
- We will use other libraries also, where appropriate.

Collections of information

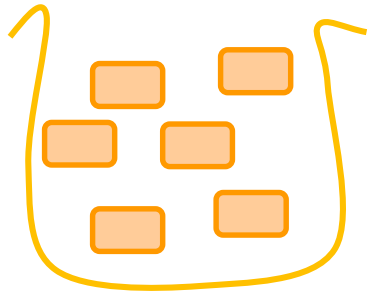
- Lots of information that computers deal with comes in **Collections**
  - Need to be able to store information in a way that reflects its structure
  - Need to be able to manipulate the information in lots of different ways

- Information about
  - earthquakes,
  - courses,
  - phone contacts,
  - documents
  - images, movies
  - people,
  - windows,
  - files,
  - network connections
  - …

Collections in Java

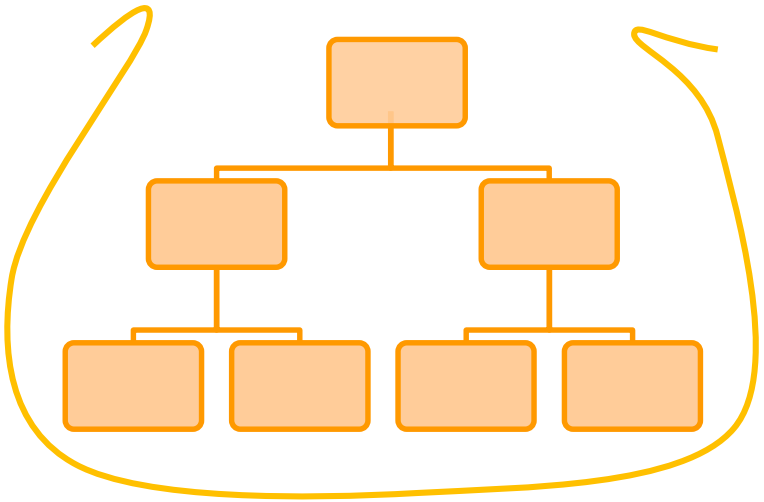
What is a Collection in Java?

- An object that contains other objects

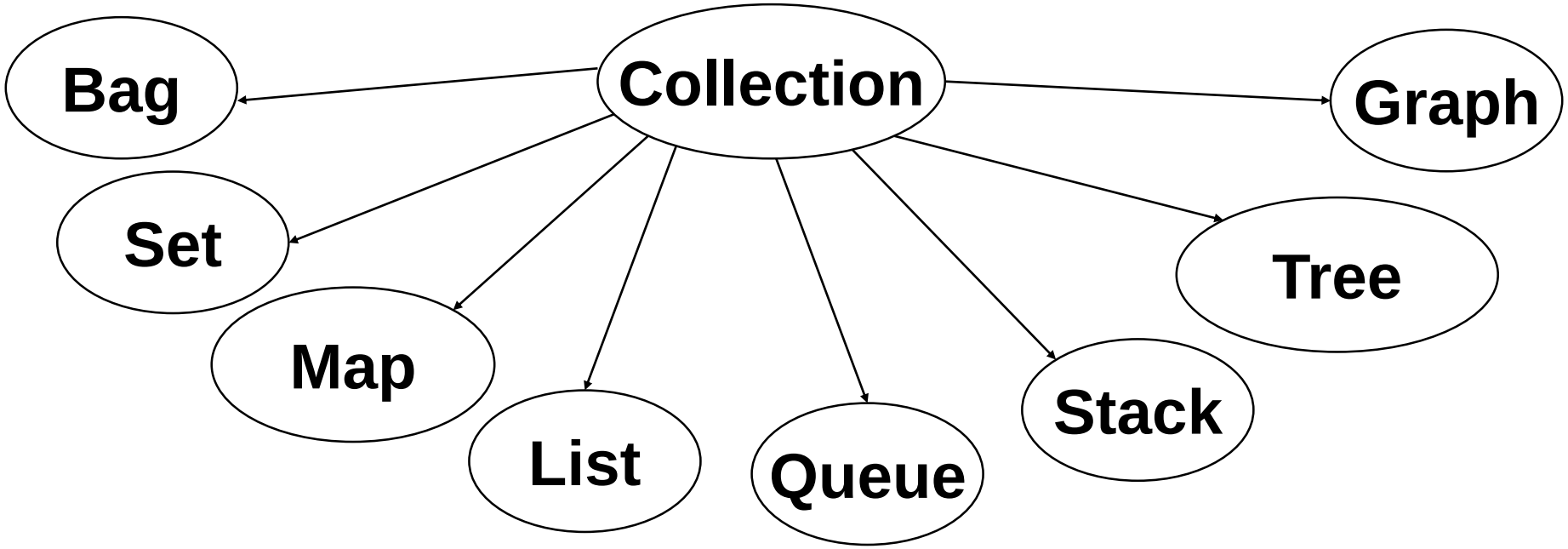


- What defines the collection type?
  - What you can do to it
  - What structure it imposes on its contents
  - What properties it ensures

- Mostly we won't care much it works inside:
  - As long as it has the right behaviour, the inside doesn't matter?
  - But we need to know what operations are expensive (and how expensive they are)
  - For trees, we will need to make our own!



“Standard” Collections



What's the difference?

Structure

Constraints

Collections: What's the difference

- Different structures
  - No structure – just a collection of values
  - Linear structure of values – the order matters
  - Set of key-value pairs
  - Hierarchical structures
  - Grid/table
  - ….
- Different constraints
  - duplicates allowed/not allowed
  - get, put, remove anywhere
  - get, put, remove only at the ends, or only at the top, or …
  - get, put, remove by position, or by value, or by key, or …
  - ….

Essential operations:

- All Collections:
  - size, clear
- Sets and Bags:
  - add, remove, contains
- Lists:
  - get(index), add(index, item), remove(index), set(index, item), indexOf(item)
- Stacks:
  - push(item), pop, peek
- Queues:
  - enqueue(item), dequeue, peek
- Maps:
  - get(key) put(key, item),

Abstract Data Types

Set, Bag, Queue, List, Stack, Map, etc are

Abstract Data Types

- an ADT is a type of data, described at an abstract level:
  - Specifies the **operations** that can be done to an object of this type
  - Specifies how it will **behave**.

- Doesn't specify how it is implemented underneath – "black box"
  - though we will always need some concrete implementation of it.

Eg: Set ADT

(simple version)

- Operations:
  - **add(***value* **)**,
  - **remove(***value* **)**,
  - **contains(***value* **)**→*boolean*

- Behaviour:
  - A new set contains no values.
  - A set will contain a value *iff* the value has been added to the set and it has not been removed since adding it.
  - A set will not contain a value *iff* the value has never been added to the set, or it has been removed from the set and has not been added since it was removed.

Java Collections Library (in java.util)

Defines interfaces ⇒ Abstract Data Types and classes: