Activity 14: Collections

Please give both your first and last name and the time you have class above. Throughout as you encounter a "Time" spot – put in the walk clock time.

Decide who will be what role for today and record first and last names below. _ We will rotate roles throughout the course. If you have three people, one may have two roles. If you have five people, two may share the same role. Please make sure you are not in the same role as last week.

Manager:	Name
Presenter:	Name
Recorder:	Name
Reflector:	Name

As you work on this activity and make notes on each other's process skills, please refer to one another on this document by your roles only. That way completed activities can be posted for the whole class and students remain anonymous. *All critiques and responses should be respectful and constructive at all times*.

This activity was written by Chris Mayfield, modified by Dee A.B. Weikle and shared under a Creative Commons License.

Collections

Arrays and lists are straightforward for storing a collection of objects. In this activity, you'll gain experience with two other kinds of collections. Sets and maps are quite useful for implementing a wide variety of algorithms.

Content Learning Objectives

After completing this activity, students should be able to:

- Summarize methods in the `Set` interface.
- Summarize methods in the `Map` interface
- Explain differences between sets and maps.

Process Skill Goals

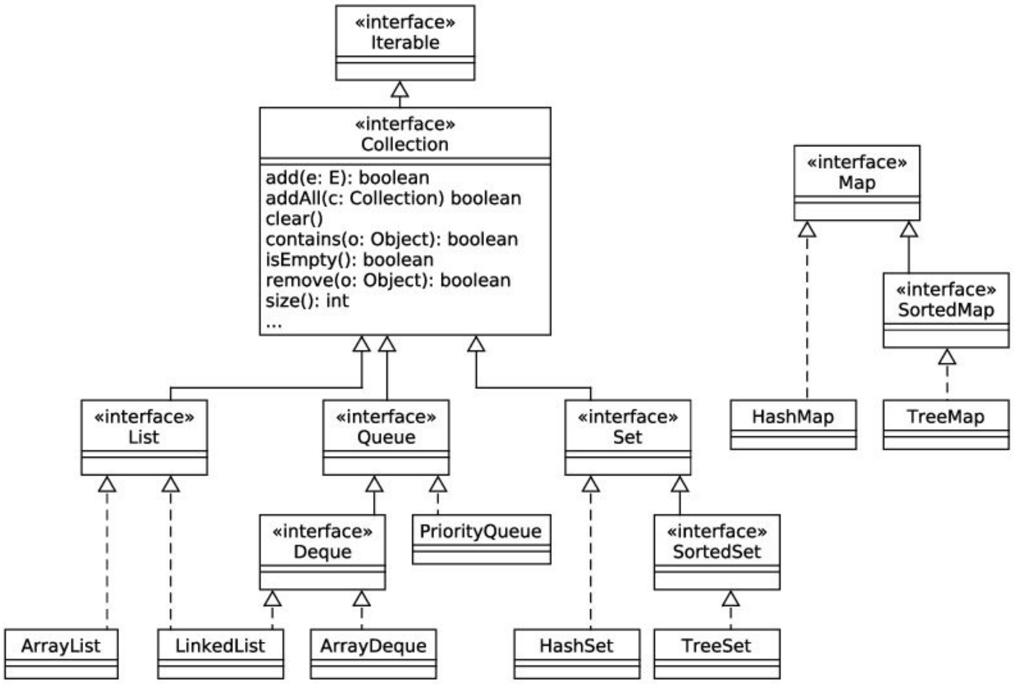
During the activity, students should make progress toward:

- Being able to explain the relationship of different classes within the Collections Hierarchy.

  (Oral Communication)

Model 1 Collections Hierarchy

The UML in the diagram below shows some of the classes and interfaces in the Java Collections Framework.



Questions (8 min)

Start time: Time

1. Fill in each blank with IS-A, or IMPLEMENTS:

a) ArrayList Answer List

b) HashMap Answer Map

c) SortedMap Answer Map

d) HashSet Answer Set

e) TreeSet Answer SortedSet

f) TreeSet Answer Set

2. List all of the classes in the UML above.

Answer

3. Based on the Diagram above, does HashMap implement Collection? Answer

4. The Deque interface is designed to support adding and removing at either end of a sequence of elements. Based on what we have learned about ArrayList, why do you think ArrayList does ***not*** implement Deque?

Answer

Model 2 Set of Strings

Each line below on the left is matched by the output from JShell on the right. Review each line and then answer each questions below as if this code was executed sequentially.

Java code	Shell output
`Set<String> names = new Set<>();`	java.util.Set is abstract; cannot be instantiated
`Set<String> names = new HashSet<>();`	[]
`names.add("WAS")`	true
`names.add("BAL")`	true
`names.add("PHI")`	true
`names`	[PHI, WAS, BAL]
`names.contains("DEN")`	false
`names.add("DEN")`	true
`names.contains("DEN")`	true
`names.contains("den")`	false
`names.add("DEN")`	false
`names.add(123)`	int cannot be converted to java.lang.String
`names.size()`	4
`names`	[PHI, WAS, DEN, BAL]
`names.remove("WAS")`	true
`names.remove("IND")`	false
`names`	[PHI, DEN, BAL]
`names.isEmpty()`	false
`names.clear()`	
`names.size()`	0
`names.isEmpty()`	true

Questions (15 min)

Start time: Time

5. For the collection above:

a) What is the interface name? Answer

c) What is the variable name? Answer

b) What is the class name? Answer

d) What is the element type? Answer

6. Based on the shell output, describe what the following methods return:

a) `add` Answer

b) `remove` Answer

7. Consider the contents of `names` just before `"WAS"` was removed.

a) What was the size of `names` at this point? Answer

b) How many times was the `add` method called? Answer

c) Explain why these two numbers are different.

Answer

8. Continuing the previous question:

a) In what order were the strings added to the set? Answer

b) In what order were they displayed in the output' Answer

c) Why do you think the two orders are different?

Answer

9. In your own words, summarize what a `Set` is in Java. Give an example from everyday life.

Answer

In discrete mathematics, sets have three basic operations:

- Union ($S \cup T$) : all elements in S or T (or both)
- Intersection ($S \cap T$) : elements in both S and T
- Difference ($S - T$) : elements in S but not in T

10. Based on the documentation for `java.util.Set`, which methods implement these operations?

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Set.html

Answer

Model 3 Map of Team Names

The following abbreviations are for National Football League (NFL) teams:

ATL	Atlanta Falcons
DEN	Denver Broncos
IND	Indianapolis Colts
MIA	Miami Dolphins
SEA	Seattle Seahawks

Review the table below the same way you did for Model 2.

Java code	Shell output
`Map<String, String> teams;`	null
`teams = new Map<>();`	java.util.Map is abstract; cannot be instantiated
`teams = new HashMap<>();`	{}
`teams.isEmpty()`	true
`teams.put("MIA", "Miami Dolphins")`	null
`teams.put("MIA", "Miami")`	"Miami Dolphins"
`teams.size()`	1
`teams`	{MIA=Miami}
`teams.put("ATL", "Atlanta")`	null
`teams.put("SEA", "Seattle")`	null
`teams`	{MIA=Miami, ATL=Atlanta, SEA=Seattle}
`teams.containsKey("ATL")`	true
`teams.containsKey("DEN")`	false
`teams.containsValue("Miami")`	true
`teams.containsValue("Dolphins")`	false
`teams.get("SEA")`	"Seattle"
`teams.get("IND")`	null
`teams.get(0)`	null
`teams.remove("MIA")`	"Miami"
`teams.remove("MIA")`	null
`teams`	{ATL=Atlanta, SEA=Seattle}
`teams.keySet()`	[ATL, SEA]
`teams.values()`	[Atlanta, Seattle]

Questions (20min) **Start time:** Time

11. For the collection above:

a) What is the interface? Answer

b) What is the class? Answer

c) What type of keys? Answer

d) What type of values? Answer

12. Based on the shell output, describe what the following methods return:

a) `put` Answer

b) `get` Answer

13. What type of object does the `keySet` method return? Describe its contents.

Answer

14. What type of object does the `values` method return? Describe its contents.

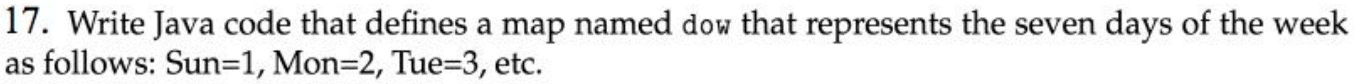
Answer

15. In your own words, summarize what a `Map` is in Java. Give an example from everyday life.

Answer

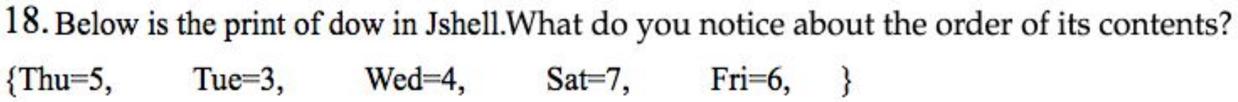
16. Why did `teams.get(0)` return null, even though there were values in the map?

Answer



17. Write Java code that defines a map named `dow` that represents the seven days of the week as follows: Sun=1, Mon=2, Tue=3, etc.

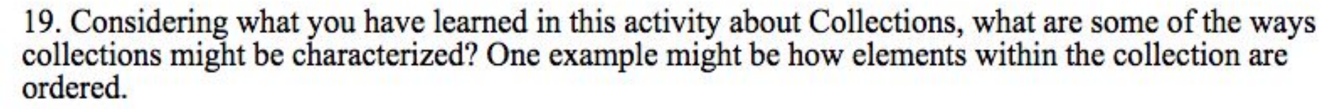
Answer



18. Below is the print of dow in Jshell.What do you notice about the order of its contents?

```
{Thu=5,     Tue=3,     Wed=4,     Sat=7,     Fri=6,   }
```

Answer



19. Considering what you have learned in this activity about Collections, what are some of the ways collections might be characterized? One example might be how elements within the collection are ordered.

Answer