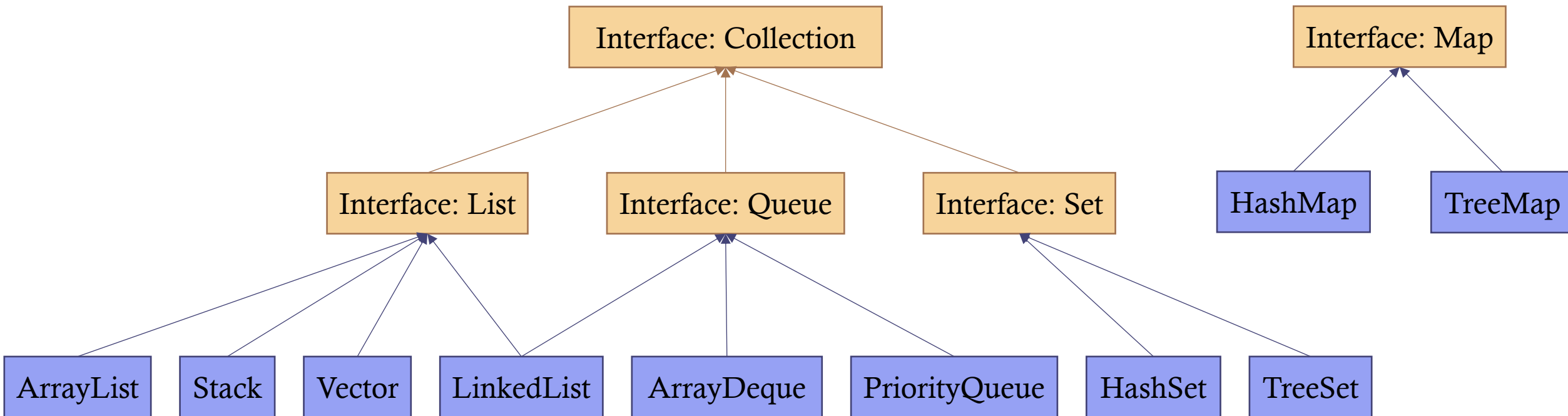

WEEK NINE

Acknowledgements: Slides created based off material provided by Dr. Michael Raymer and Dr. Travis Doom

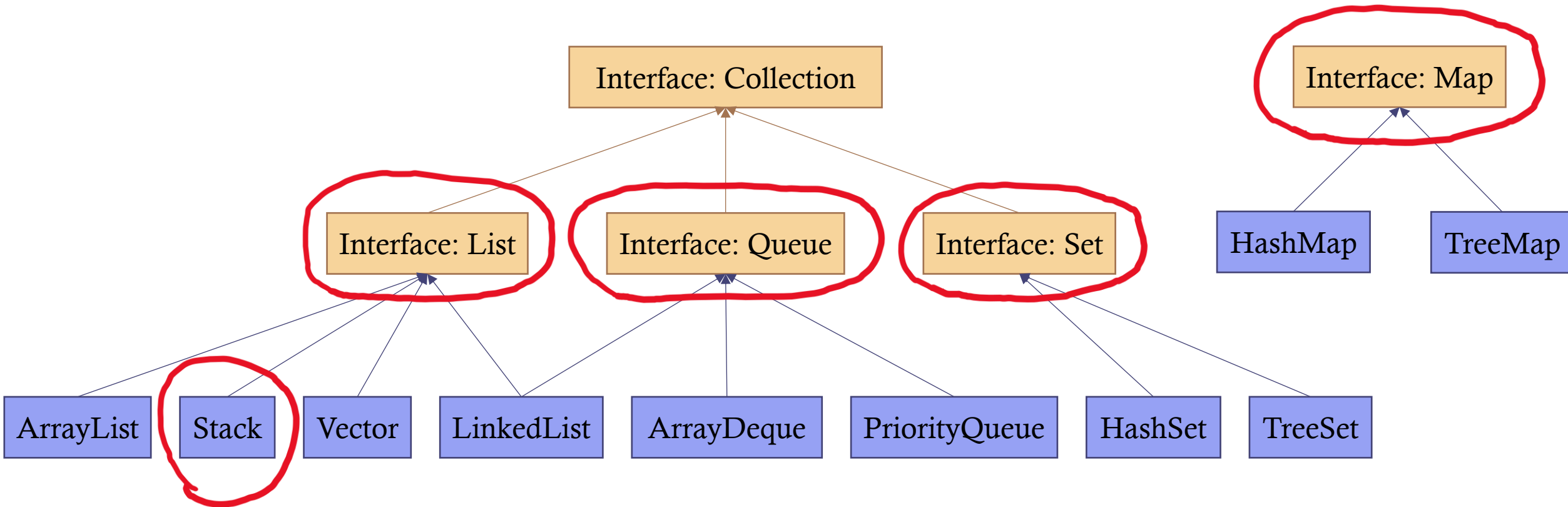
JAVA COLLECTION INTERFACE

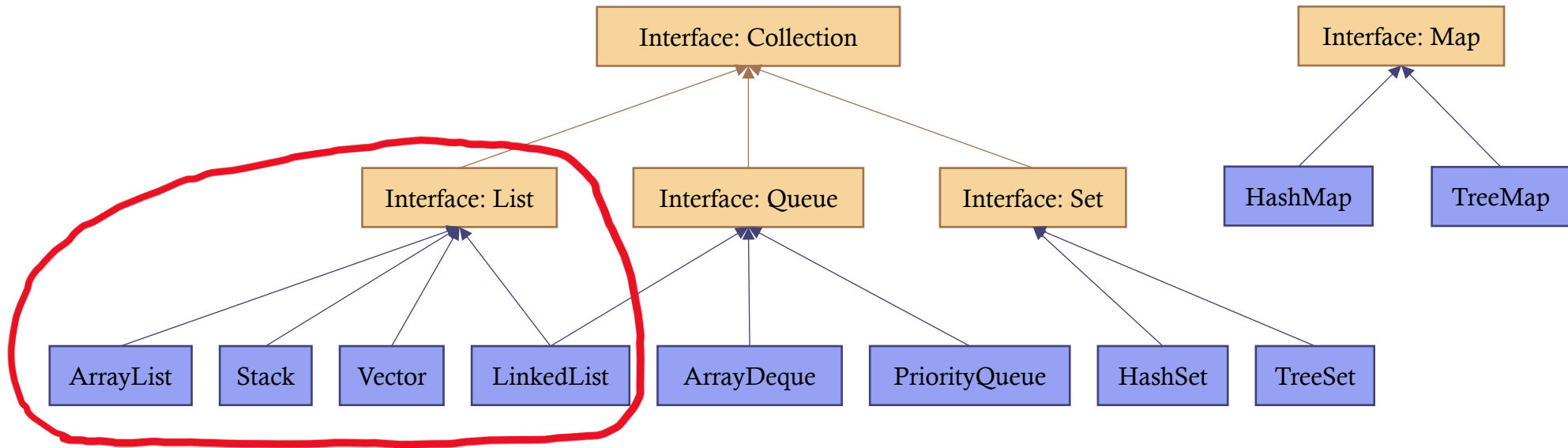


ABSTRACT DATA TYPES (ADTS)

- Abstract container for data with some loose operations
- An ADT is a general idea for data collection, details are hidden
- Describes how we want to store/interact data at a high-level
- An ADT **does not**:
 - Specify/restrict the type of data to be stored (generics)
 - Specify the implementation of operations (method bodies don't need to be specified)
 - Dictate how the data is actually stored/accessed from memory
- In Java, ADTs are often (but not always) implemented via interfaces

INTERESTING ADTS IN JAVA





LISTS

- ArrayLists
- LinkedLists

LIST INTERFACE

- An ordered, indexed collection
- Users can precisely insert/access elements via an index
- Duplicates typically allowed
- Key methods:
 - add()
 - get()
 - isEmpty()
 - remove()

ARRAYLIST

- Element references stored contiguously in memory
- Start
- Size

LINKEDLIST

- Element references don't need to be stored contiguously
- Each element holds a *link* or reference to the next item in the list
- Head