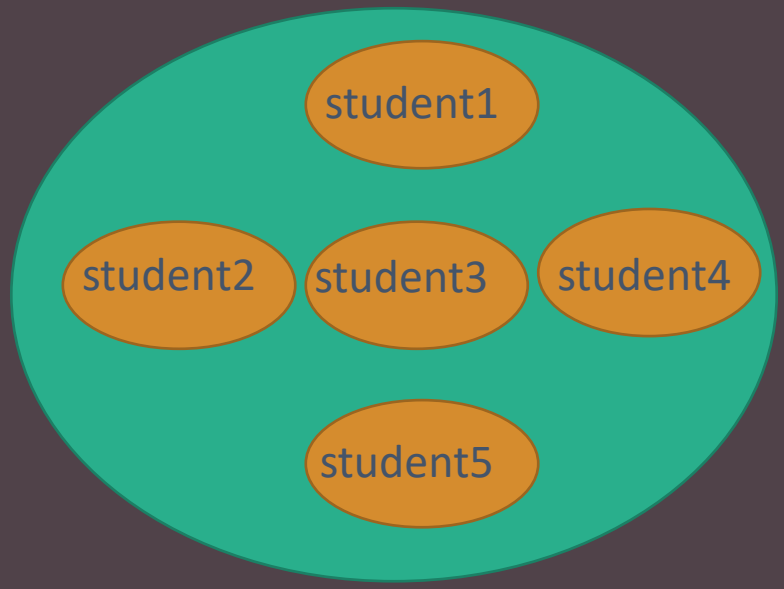
COLLECTION FRAMEWORK

What is collection?

- Collection is a group of individual objects as a single entity.



Cybertek

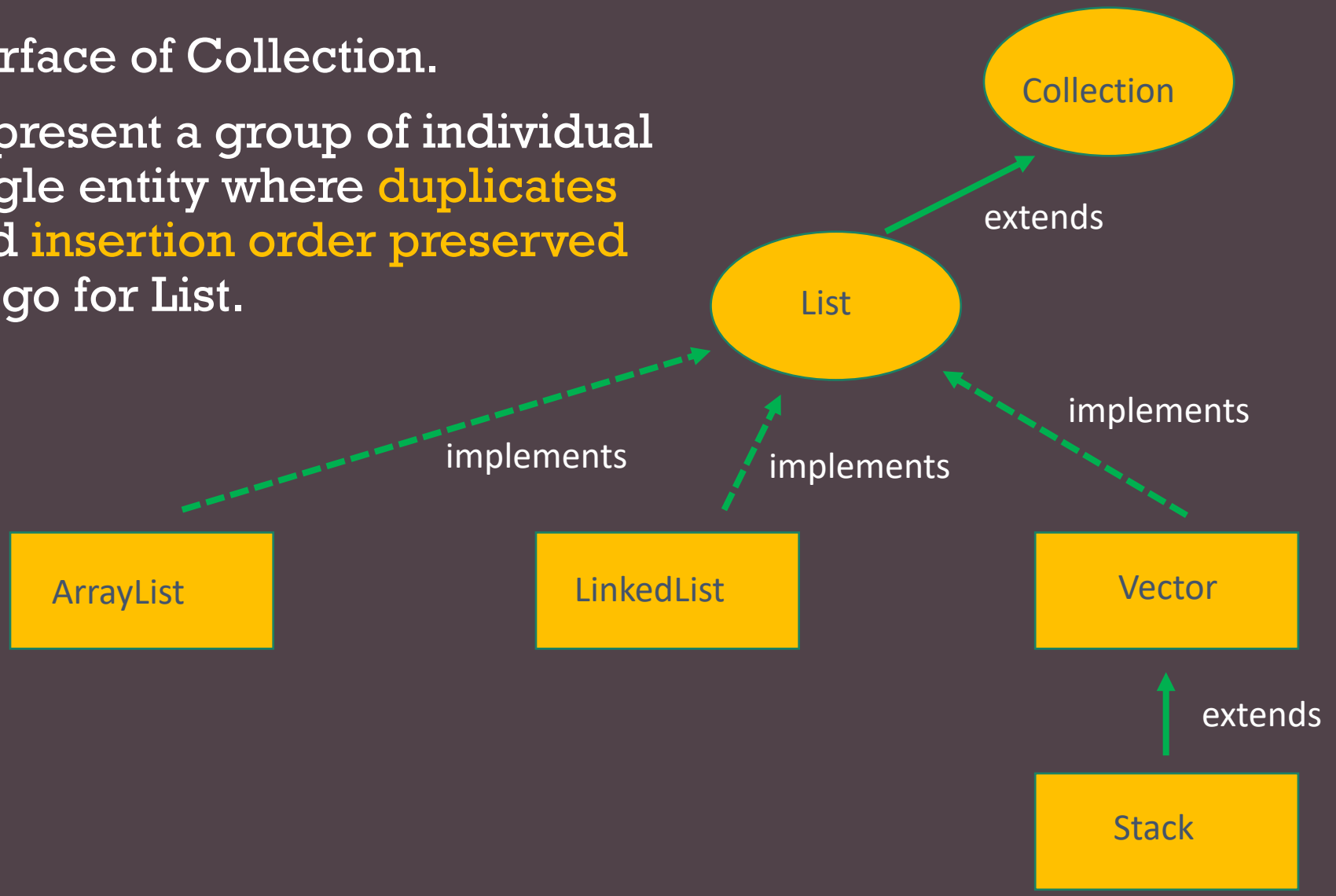
COLLECTIONS

- Growable in nature. Can increase or decrease the size.
- Can hold different data types.
- Standard data structure. There are ready methods to use.
- It defines several classes and interfaces which can be used to represent a group of objects as single entity.

LIST

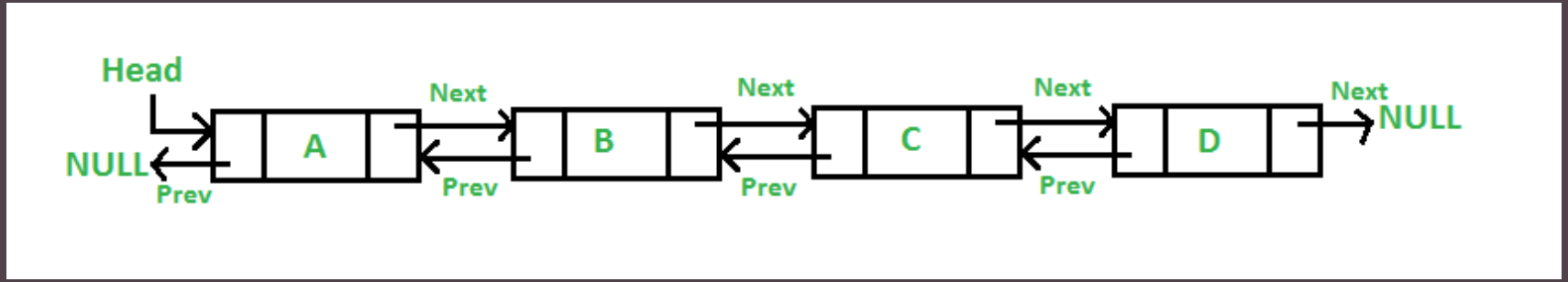
Cybertek

- List is child interface of Collection.
- If we want to represent a group of individual objects as a single entity where duplicates are allowed, and insertion order preserved then we should go for List.



Doubly Linked List

- Doubly Linked List: (DLL) contains an extra pointer, typically called *previous pointer*, together with next pointer and data which are there in singly linked list.



Cybertek

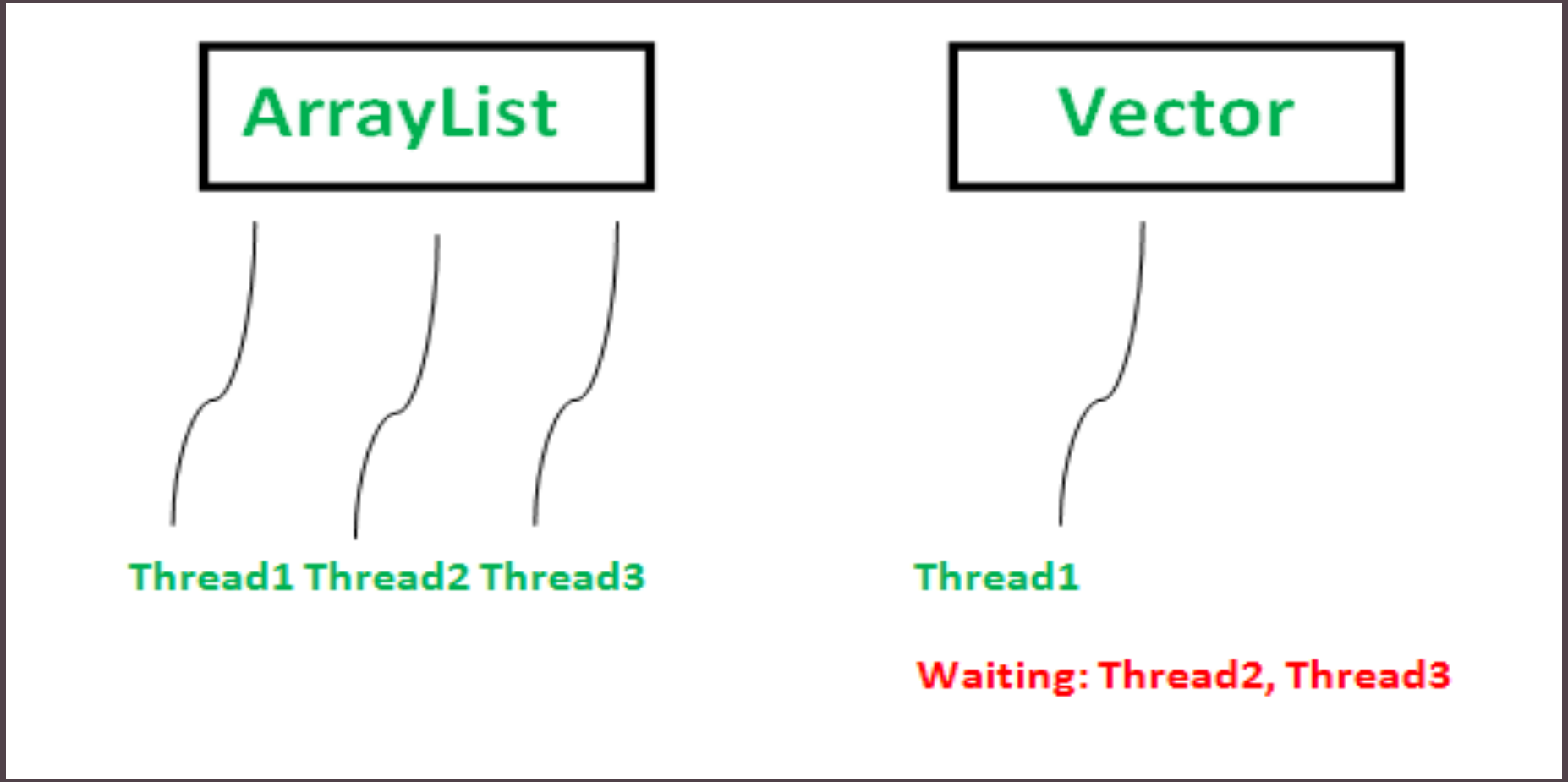
Vector

- Vector implements a dynamic array. It is similar to ArrayList, but with two differences :
  - Vector is synchronized(thread-safe)
  - Vector contains many legacy methods that are not part of the collection framework.

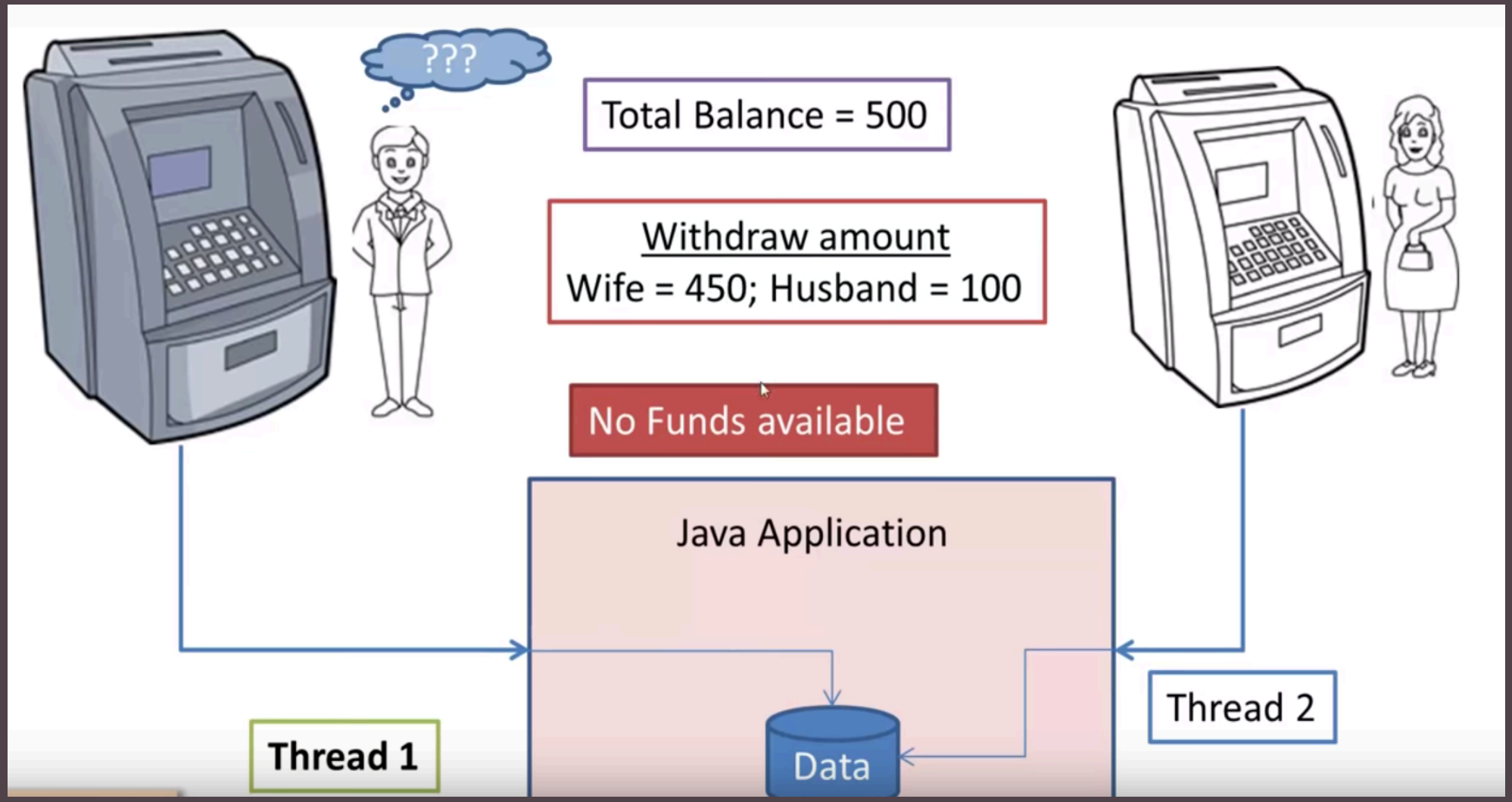
Synchronize

- Code Synchronization helps in preventing multiple threads executing a code simultaneously.
- Code Synchronization is implemented with help of Locks.
- A thread that is trying to access the code that is marked as Synchronized should acquire the lock from object.
- Locks:
  - Every object has a lock. Only one lock per object.
- Synchronize can only be applied for methods and block of code.

Synchronized (Thread-Safe)

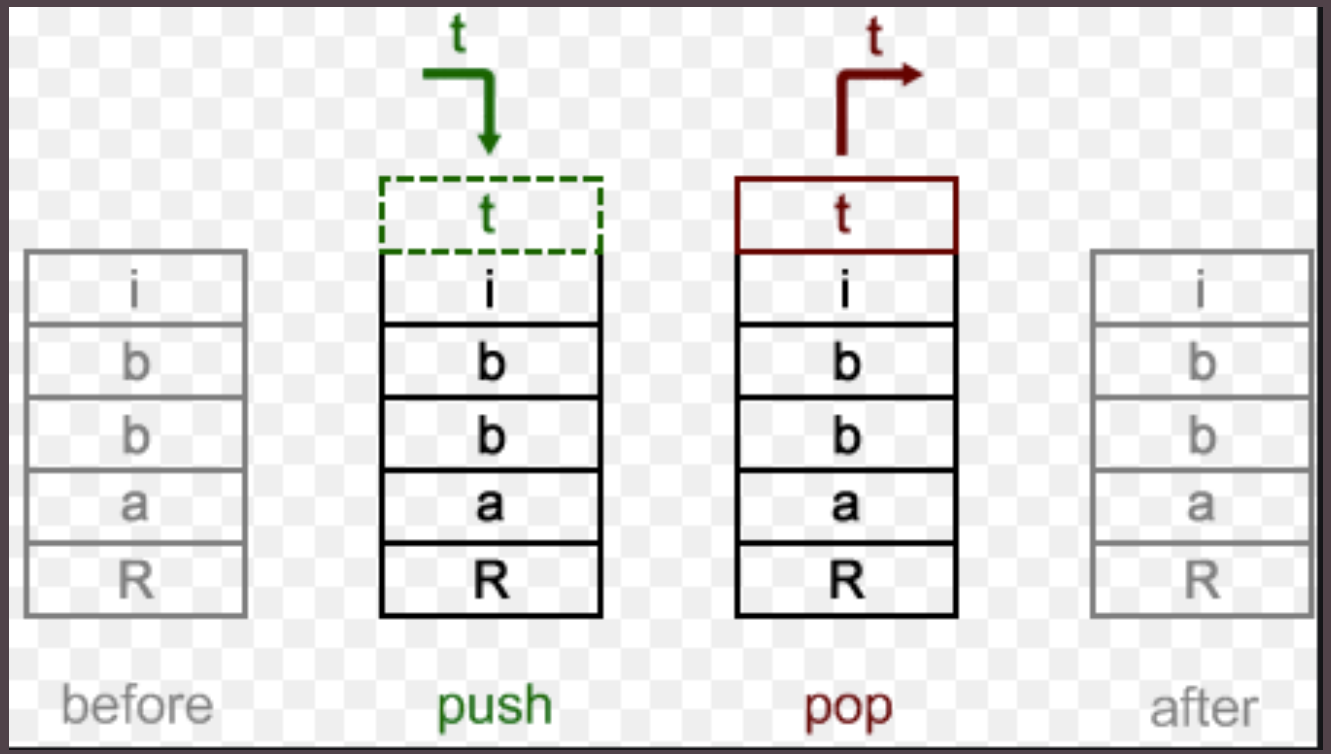


Synchronized (Thread-Safe)



Stack

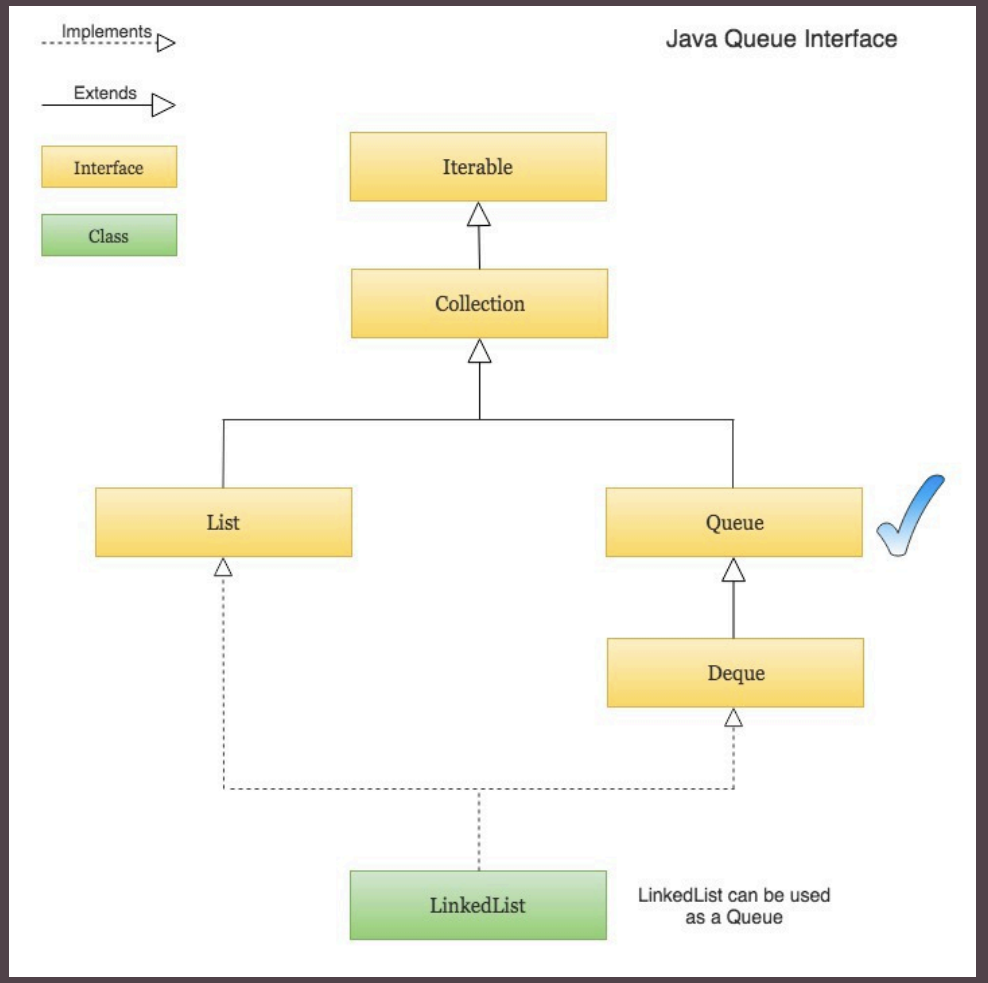
- Stack is a subclass of Vector that implements a standard last-in, first out. (LIFO)



Cybertek

Queue

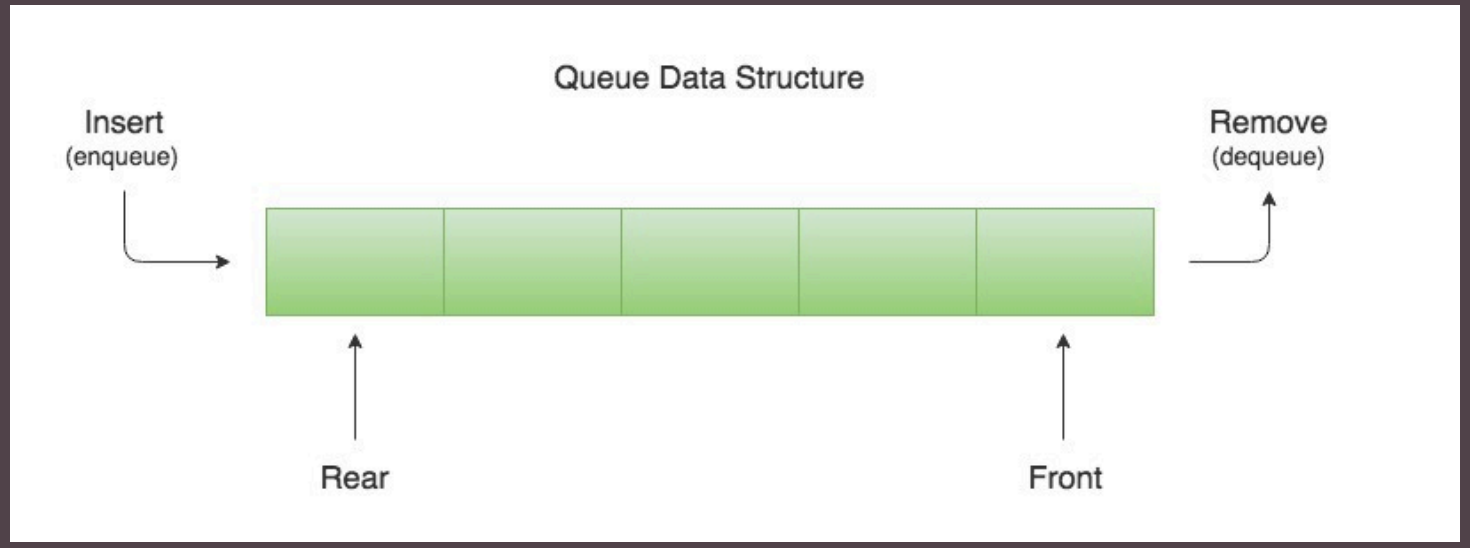
- It is child interface of Collection.
- If we want to represent a group of individual objects prior to processing then we should go for Queue.



Cybertek

Queue

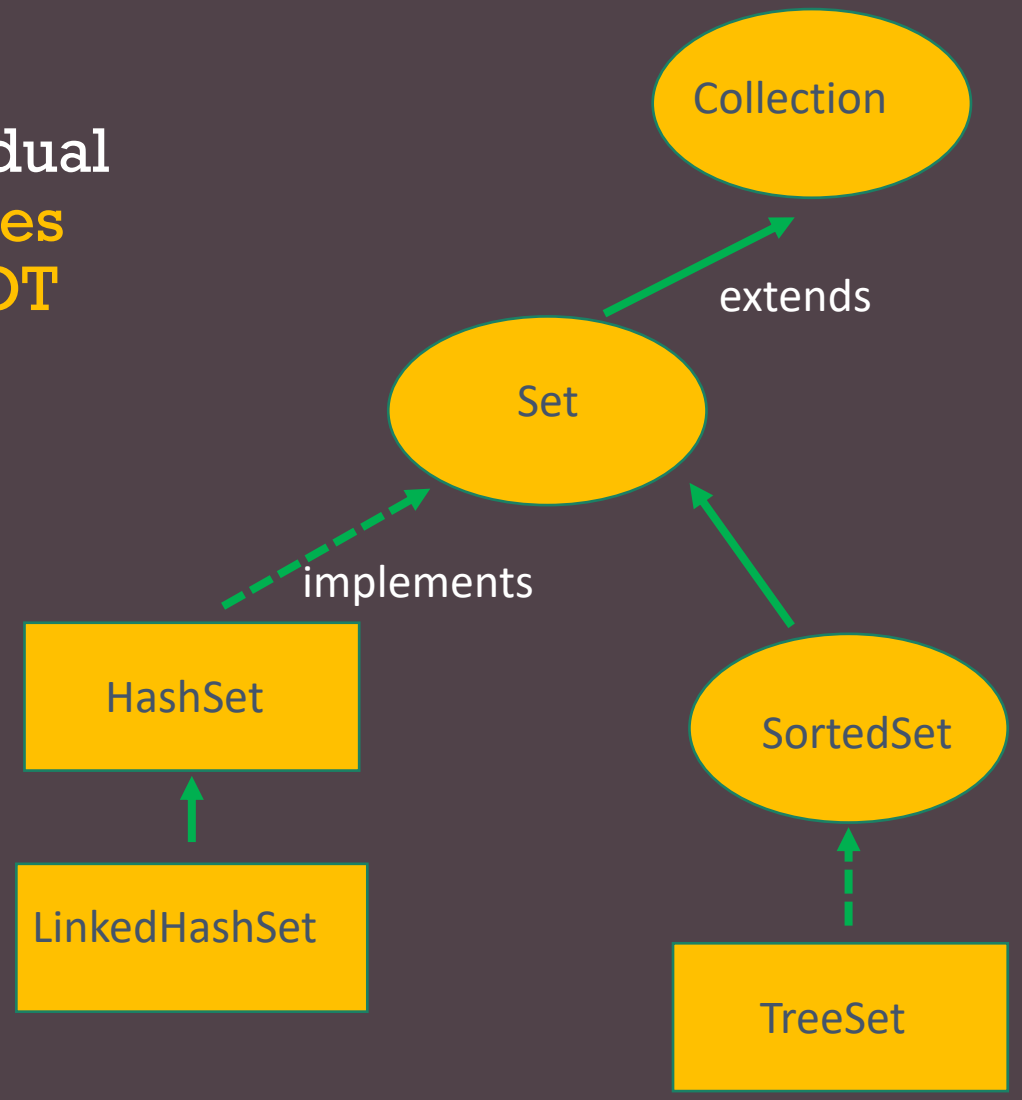
- A Queue is a First In First Out (FIFO) data structure.



Cybertek

SET

- Set is child interface of Collection.
- If we want to represent a group of individual objects as a single entity where duplicates are NOT allowed, and insertion order NOT preserved then we should go for Set.



Cybertek

DIFFERENCES BETWEEN LIST AND SET

LIST	SET
Duplicates are allowed	Duplicates are not allowed
Insertion order preserved	Insertion order not preserved

Loop Through Collection

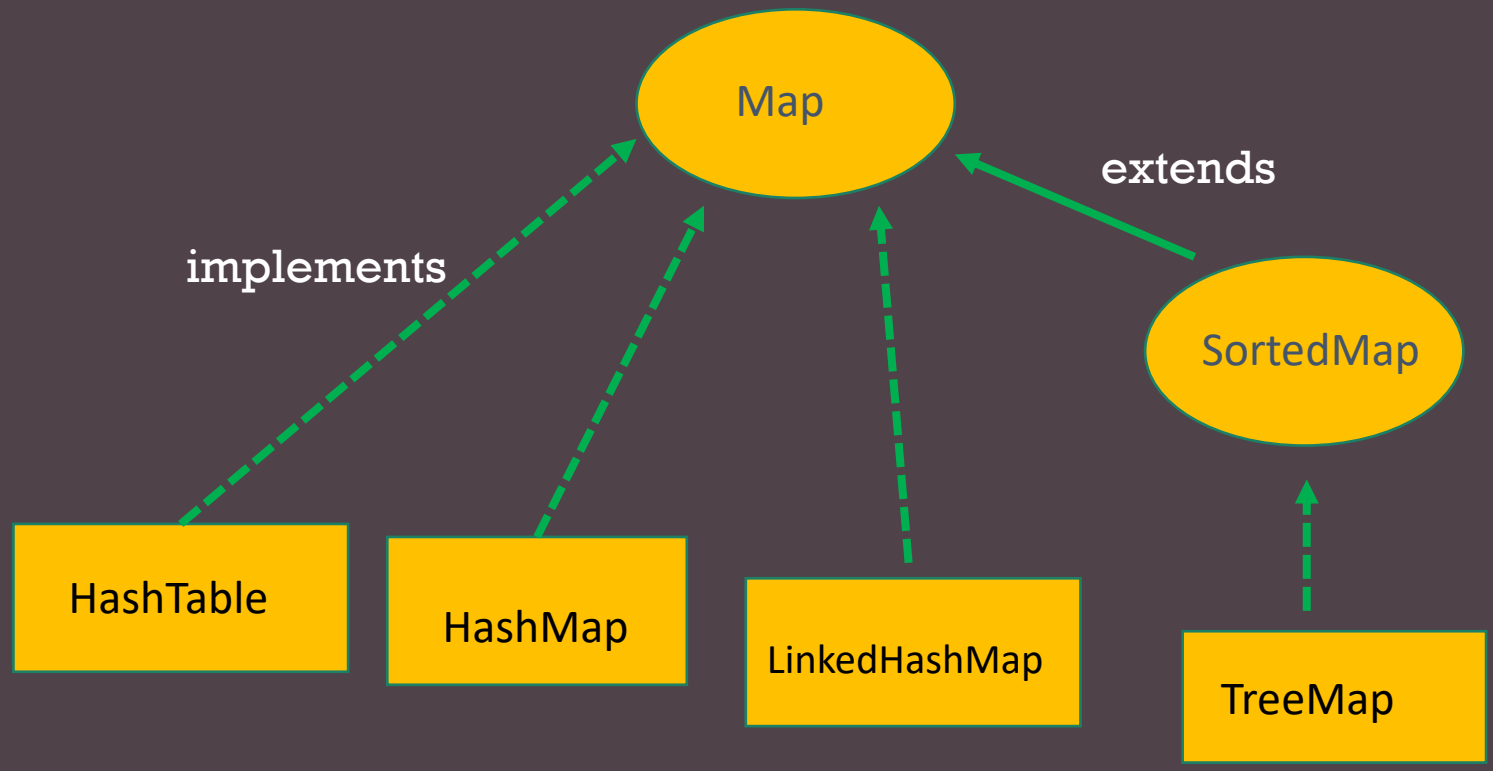
1. For each loop
2. Any other loop(for, while, do while) by using get(index) method
3. Iterator
4. forEach method that came with java 8 (lambda expression)

What is the difference between Iterator and For Each Loop?

- When using iterator object, we can remove values while looping
- When using for each loop, we can not remove values from the collection
- We need to create iterator object to able to use it
- For each loop works with a temporary variable

Collection of Pairs : Map

- Data structure based on key + value pairs
- Map interface does not extend Collection interface



Difference between HashMap and HashTable

HashMap	HashTable
Every method in HashMap are not synchronized	Every method in HashTable are synchronized
HashMap is fast	HashTable is slow
HashMap allows one null key and multiple null values	HashTable does not allow any null key or value
HashMap implements Map interface	HashMap implements Map interface

Difference between HashMap and TreeMap

HashMap	TreeMap
HashMap does not maintain any sorting order	TreeMap elements are sorted according to natural sorting order
Internally it used hash table	Internally is uses Red Black Tree
Contains one null key and many null values	Can not contain null keys but may contains many null values
HashMap implements Map interface	It implements SortedMap interface