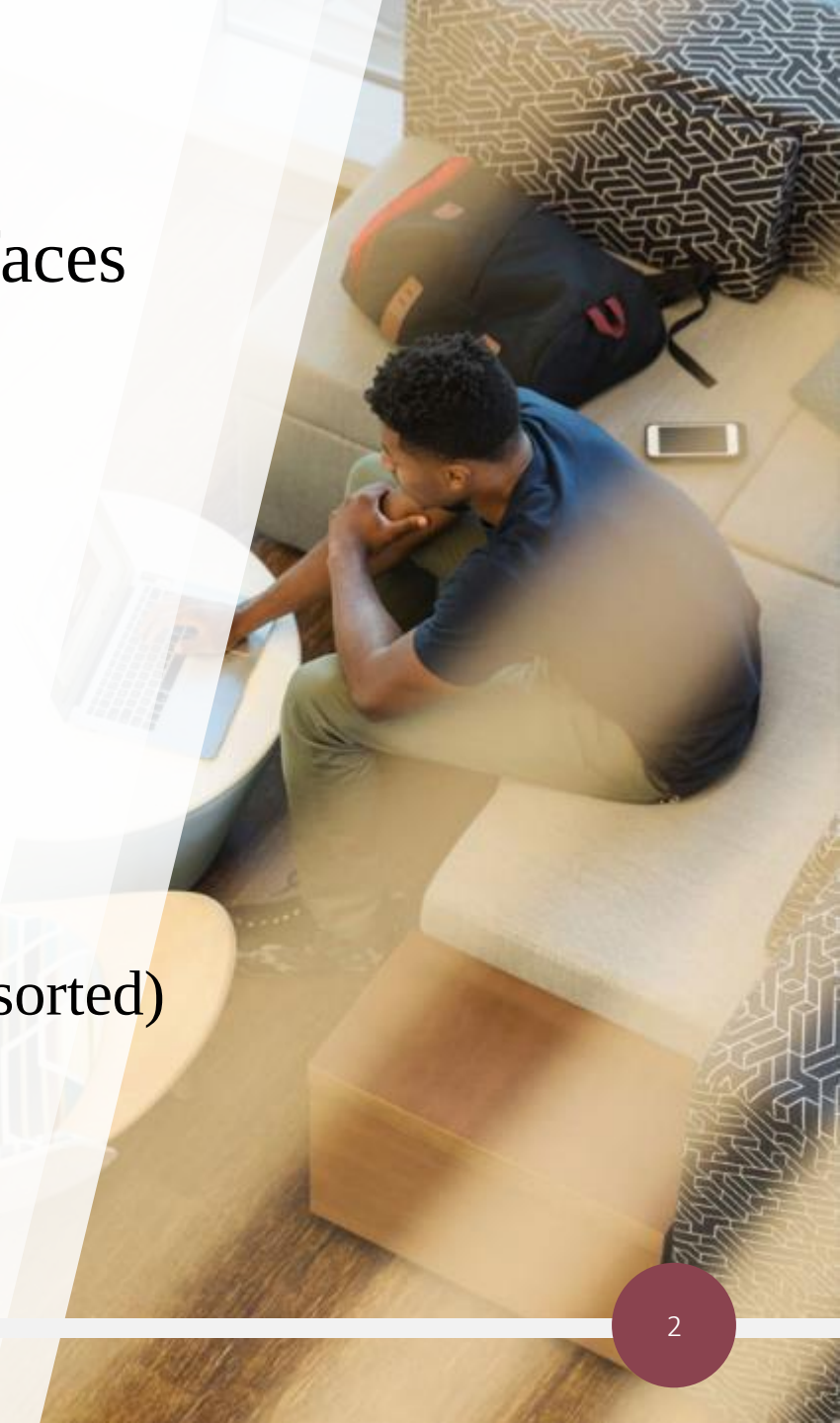
Java 21

Sequenced Collections

Sequenced Collections

- Sequenced Collections (JEP 431) establish new interfaces for collections with a defined encounter order.

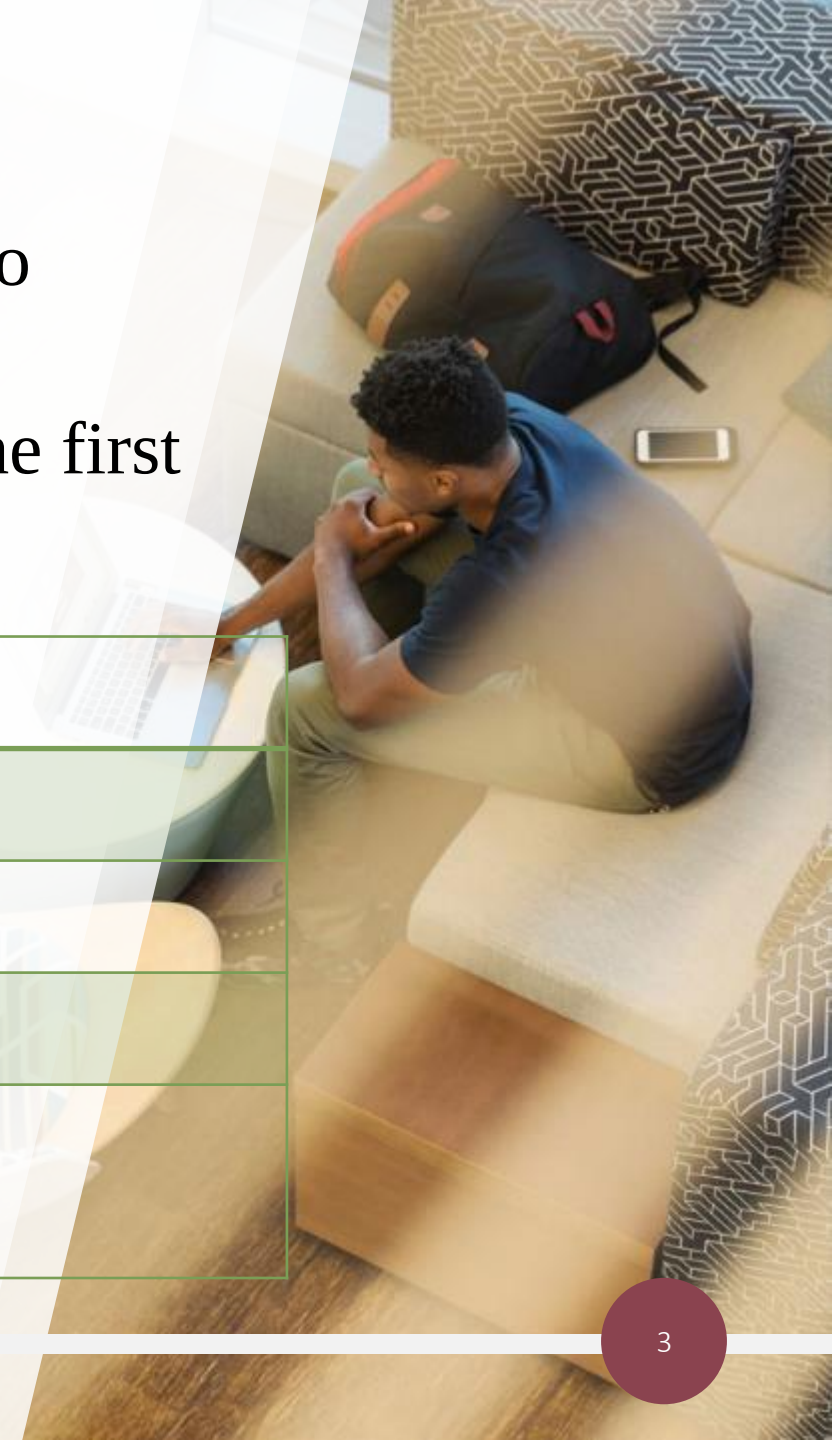
- Motivation:
  - regarding defined encounter order, Java's collections framework is neither straightforward or consistent.
  - for example, *List* and *Deque* define an encounter order but their common supertype, *Collection*, does not.
  - *Set* does not define an encounter order, and neither does its subtype *HastSet*; but other subtypes do, such as *SortedSet* (sorted) and *LinkedHashSet* (insertion-order).



Sequenced Collections

- Without interfaces to define them, operations related to encounter order are implemented inconsistently.
- Even though many implementations support getting the first or last element, each collection defines its own way.

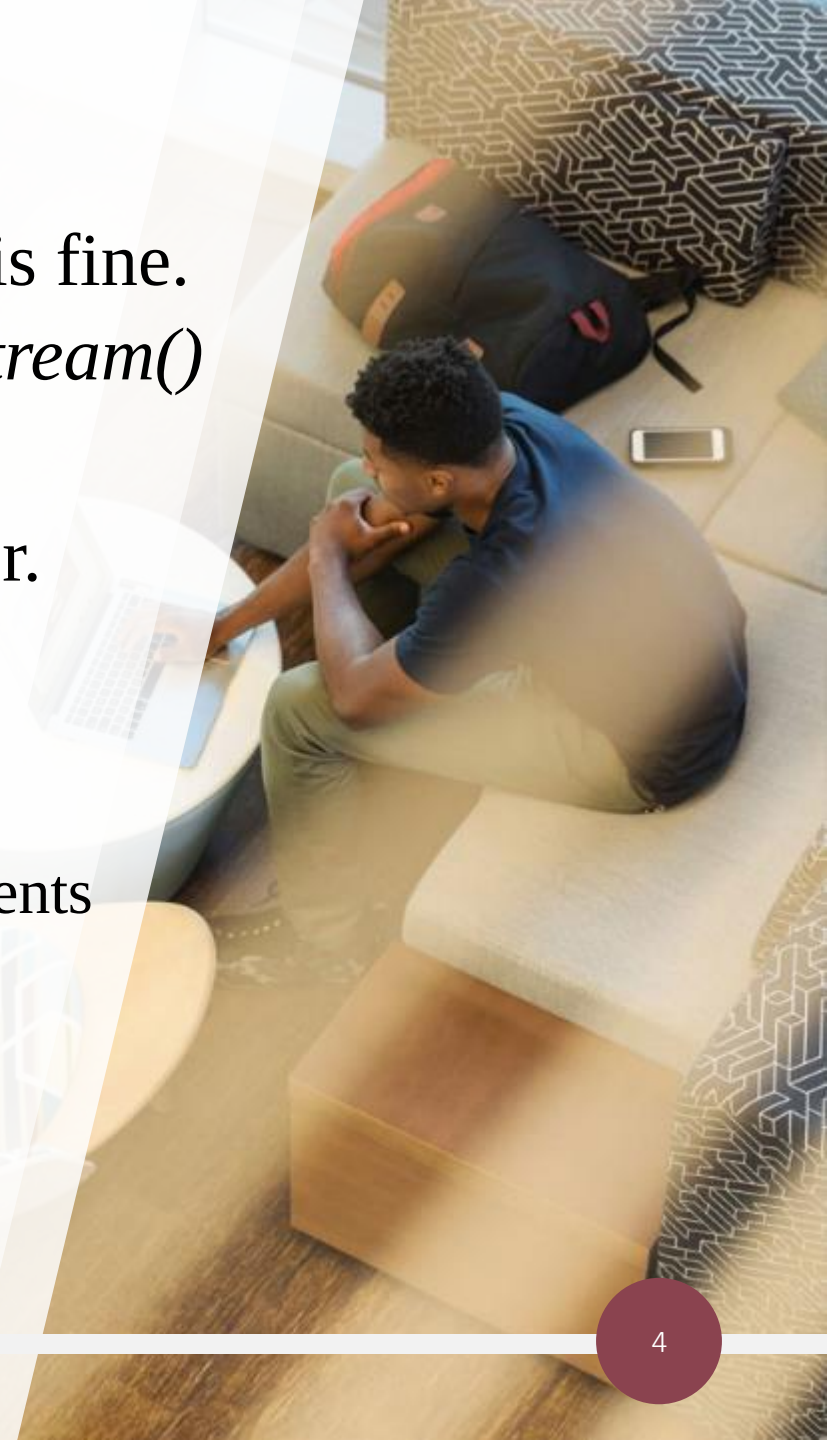
	First element	Last element
List	list.get(0)	list.get(list.size() -1)
Deque	deque.getFirst()	deque.getLast()
SortedSet	sortedSet.first()	sortedSet.last()
LinkedHashSet	linkedHashSet.iterator().next()	// missing



Sequenced Collections

- Iterating the elements in a collection from first to last is fine.
  - for example: enhanced-*for* loop, use an iterator or *stream()*

- However, iterating in reverse order is a different matter.
  - *NavigableSet* provides a *descendingSet()*
  - *Deque* provides *descendingIterator()*
  - *List* provides a *ListIterator*
  - *LinkedHashSet* provides no support (one must copy its elements into another collection).



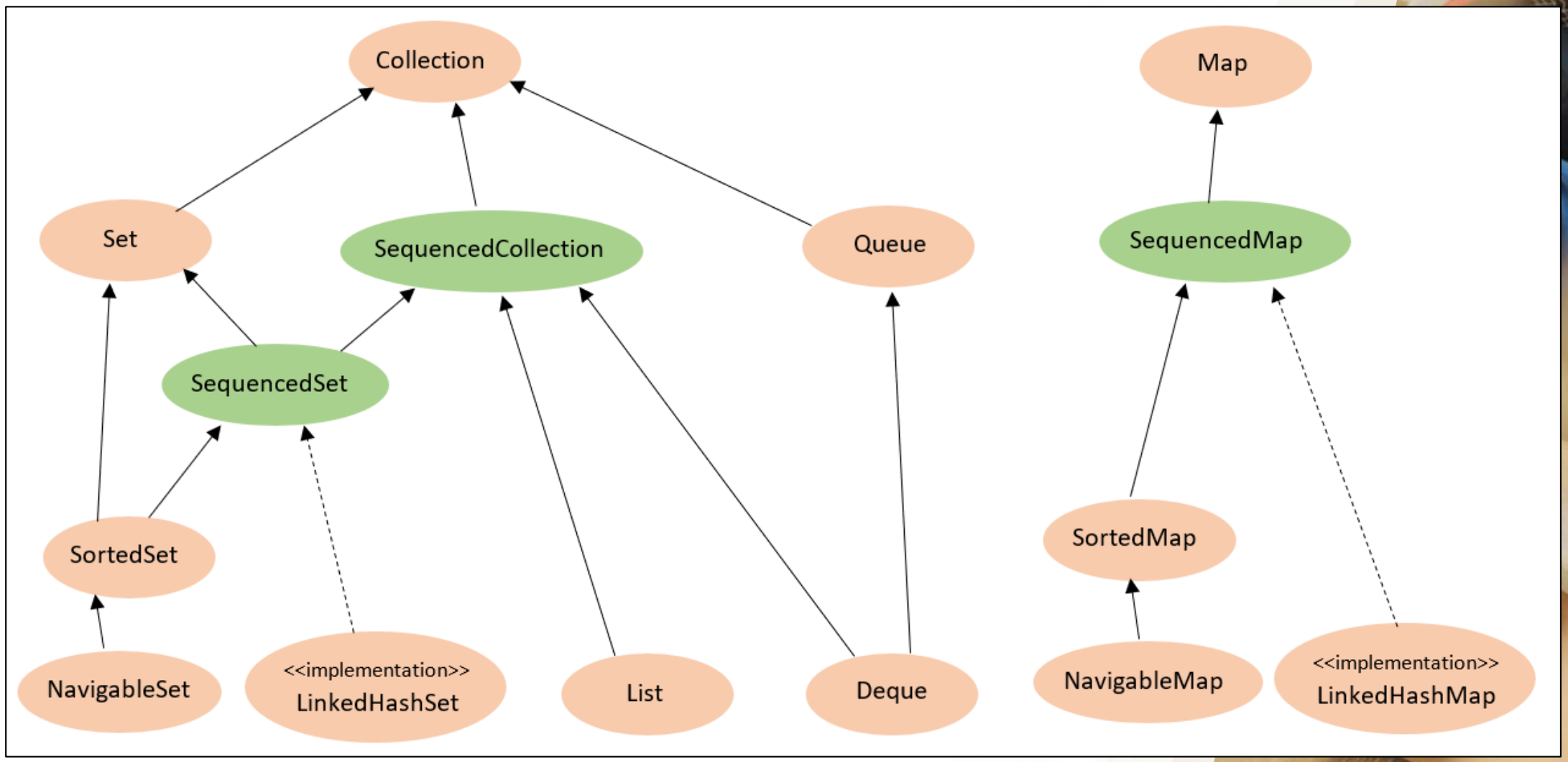
Sequenced Collections

- In JEP 431, new interfaces are defined for sequenced collections, sequenced sets and sequenced maps.

- APIs are defined for accessing first/last elements and also for processing elements in reverse order.

- All of the methods in the interfaces have default implementations.

Sequenced Collections - API



Sequenced Collections - API

```
interface SequencedCollection<E> extends Collection<E>{
      // new method
      SequencedCollection<E> reversed();
      // methods promoted from Deque
      void addFirst(E);
      void addLast(E);
      E getFirst();
      E getLast();
      E removeFirst();
      E removeLast();
}

interface SequencedSet<E> extends Set<E>, SequencedCollection <E>{
      // new method
      SequencedSet<E> reversed();
}
```

Sequenced Collections - API

```
interface SequencedMap<K,V> extends Map<K,V>{
      // new methods
      SequencedMap<K,V> reversed();
      SequencedSet<K> sequencedKeySet();
      SequencedCollection<V> sequencedValues();
      SequencedSet<Entry<K,V>> sequencedEntrySet();
      V putFirst(K, V);
      V putLast(K, V);
      // methods promoted from NavigableMap
      Entry<K,V> firstEntry();
      Entry<K,V> lastEntry();
      Entry<K,V> pollFirstEntry();
      Entry<K,V> pollLastEntry();
}
```