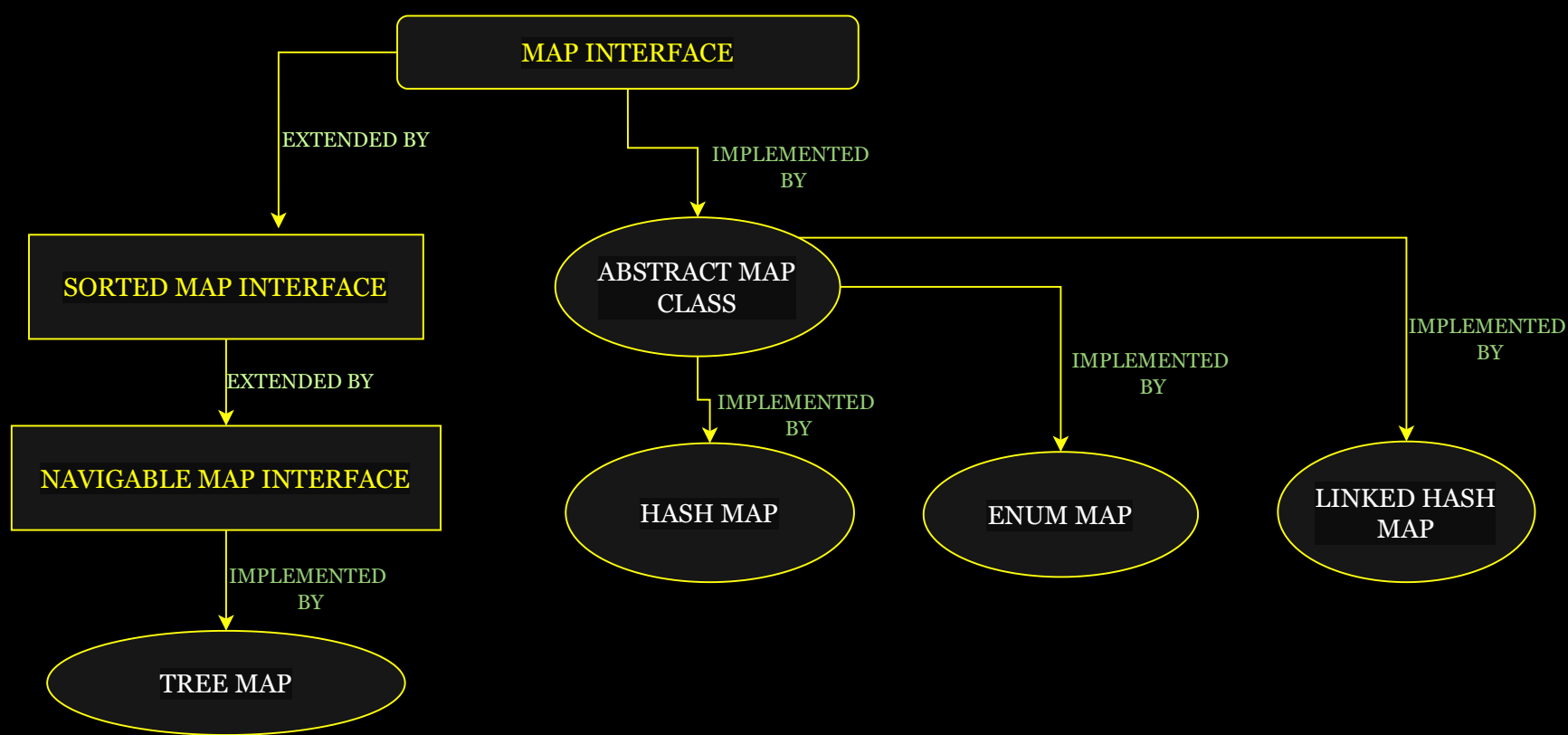
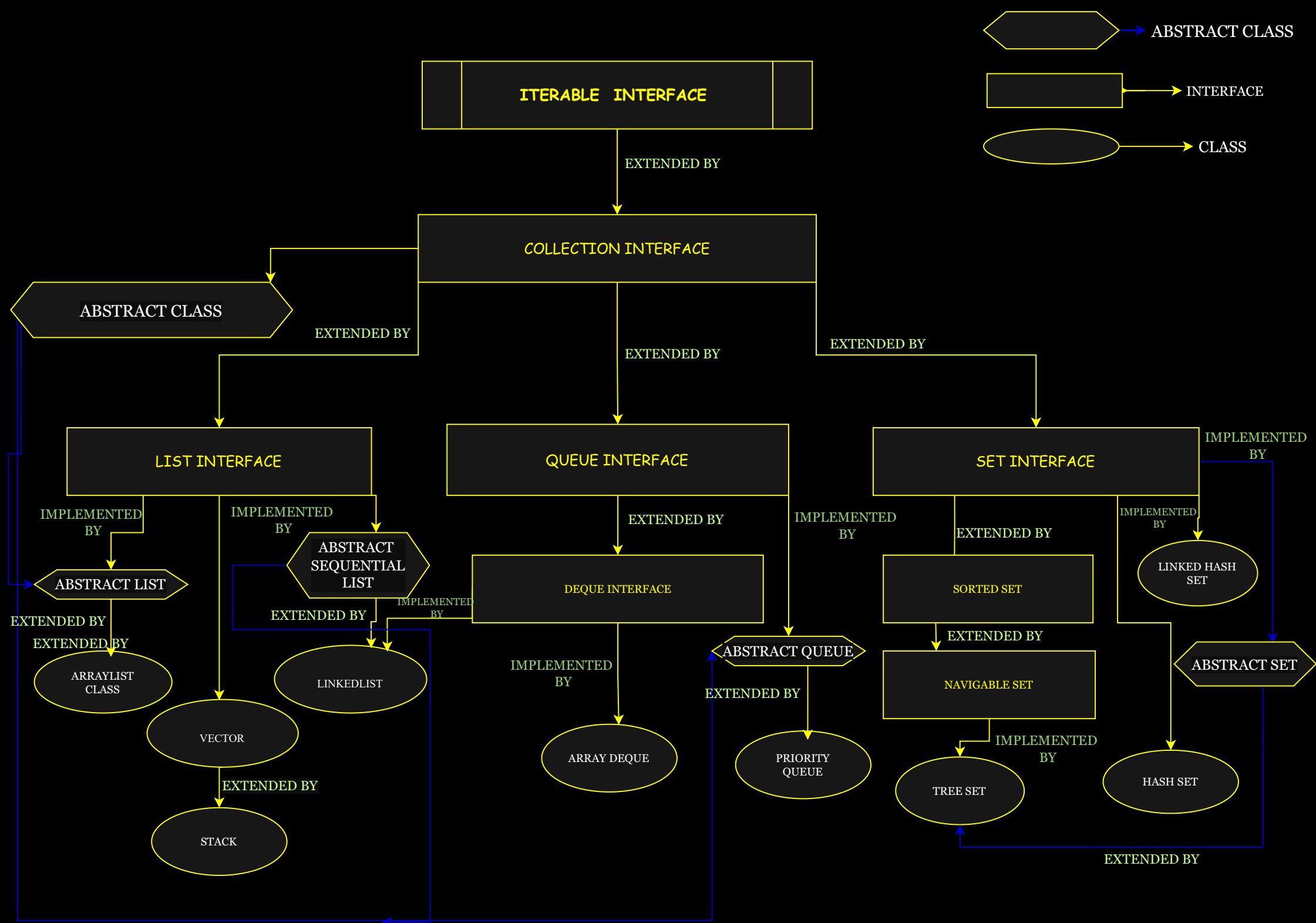


# Hierarchy of the Collection Framework



## FUNCTIONS AND METHODS IN COLLECTION

## COLLECTION INTERFACE

<u>add()</u>	<u>size()</u>
<u>addAll(Collection c)</u>	toArray()
<u>clear()</u>	toArray(T[] a)
<u>contains(Object o)</u>	retainAll(Collection<?> c)
<u>containsAll(Collection c)</u>	<u>toArray(generator)</u>
<u>equals(Object o)</u>	<u>stream()</u>
<u>hashCode()</u>	spliterator()
<u>isEmpty()</u>	Sort()
<u>iterator()</u>	
<u>remove(Object o)</u>	
<u>removeAll(Collection c)</u>	
<u>removeIf(Predicate)</u>	
<u>forEach( action)</u>	

List
set()
get()
indexOf(element)
lastIndexOf(element)
sort(Comparator comp)

MAP
containsValue(Object)
containsKey(Object)
entrySet()
get(Object)
keySet()
putAll(Map)
values()
putIfAbsent(K key, V value)
put(Object, Object)

ITERATOR & ENUMERATION
hasNext()
next()
hasMoreElements()
nextElement()
hasPrevious(),previous()
previousIndex()

## ARRAY LIST METHODS

- clone()
- ensureCapacity?(int minCapacity)
- listIterator?()
- listIterator?(int index)
- removeRange?(int fromIndex, int toIndex)
- subList?(int fromIndex, int toIndex)
- trimToSize()

## VECTOR METHODS

- elementAt(int index)
- elements()
- clone()
- capacity()
- copyInto(Object[] anArray)
- firstElement()
- lastElement()
- lastIndexOf(Object o, int index)
- removeElement(Object obj)
- removeElementAt(int index)
- removeIf(filter)
- removeRange(int fromIndex,int toIndex)
- setSize(int newSize)
- subList(int fromIndex, int toIndex)
- toString()
- trimToSize()
- toArray()

```
ENUM AND HASH MAP METHODS
```

```
clone()  
equals(Object o)  
hashCode()  
keySet()  
values()  
Size()  
toString()  
isEmpty()
```

## STACK METHODS

```
empty()
peek()
pop()
search(Object element)
push(Object element)
```

## ENUM SET METHODS

allOf(Class<E> elementType)
clone()
complementOf(EnumSet<E> s)
copyOf(Collection<E> c)
copyOf(EnumSet<E> s)
noneOf(Class<E> elementType)
of(E e)
of(E first, E... rest)
of(E e1, E e2, E e3)
of(E e1, E e2, E e3, E e4)
range(E from, E to)

## HASH TREE METHODS

ceiling?(E e)
clear()
Comparator comparator()
contains(Object o)
descendingIterator?()
descendingSet?()
first()
floor?(E e)
headSet(Object toElement)
higher?(E e)
isEmpty()
Iterator iterator()
last()
lower?(E e)
pollFirst?()
pollLast?()
remove(Object o)
size() T
spliterator()
subSet(fromElement, toElement)
tailSet(Object fromElement)

## LINKED\_LIST METHODS

```
addFirst(E e)
addLast(E e)
descendingIterator()
element()
get(int index)
getFirst()
getLast()
indexOf(Object o)
lastIndexOf(Object o)
offer(E e)
offerLast(E e)
peekLast()
peekFirst()
poll(),pollFIRST(),POLLLAST()
pop()
push(E e)
removeFirstOccurrence(Object o)
toArray()
toString()
toArray(T[] a)
```

## HASH SET METHODS

```
add(E e)
clear()
contains(Object o)
remove(Object o)
isEmpty()
clone()
equals()
hashCode()
removeAll(collection)
toString()
```

# Real world examples where collections can be used?

## Application of ArrayLists:

- Use anywhere no removal and insertions in the middle
- Collecting database records into ArrayList
- Returning List of transactions for Credit Card

## Application of Linked Lists:

- image viewer software
- Web pages can be accessed using the previous and the next URL links and in music players
- symbol table management in a designing compiler
- Train coaches are connected to one another

## Circular linked List:

- Escalators
- To keep the track of turns in a multi-player gam
- switching between applications and programs in pc

## Application of Stack:

- Converting infix to postfix expressions.
- Wearing/Removing Bangles, Pile of Dinner Plates, Stacked chairs
- Undo/Redo button/operation in word processors
- Syntaxes in languages are parsed using stacks
- History of visited websites
- Message logs and all messages you get are arranged in a stack
- Call logs, E-mails, Google photos’ any gallery, YouTube downloads, Notifications ( latest appears first )
- Recursion and parentheses matching

## Application of Queue:

- job scheduling in OS.
- Data packets in communication are arranged in queue format
- Sending an e-mail, it will be queued.
- Most internet requests and processes use queue.
- Server while responding to request
- Printer spooler

## Priority Queue:

- Process scheduling in the kernel.
- Priority queues are used in file downloading operations in a browser

## Application of Tree:

- Databases also use tree data structures for indexing
- File explorer/my computer of mobile/any computer
- Posting questions on websites like Quora, the comments are a child of questions
- Evaluate an expression
- Used by JVM (Java Virtual Machine) to store Java objects
- To store the possible moves in a chess game

## Application of Hash Tables:

- Data stored in databases is generally of the key-value format which is done through hash tables
- Every time we type something to be searched in google chrome or other browsers,
- it generates the desired output based on the principle of hashing
- In order to make a connection between the filename to its corresponding file path hash tables are used
- Social network “feeds”
- To store a set of fixed keywords that are referenced very frequently
- Used for fast data lookup-symbol table for compilers, database indexing, caches