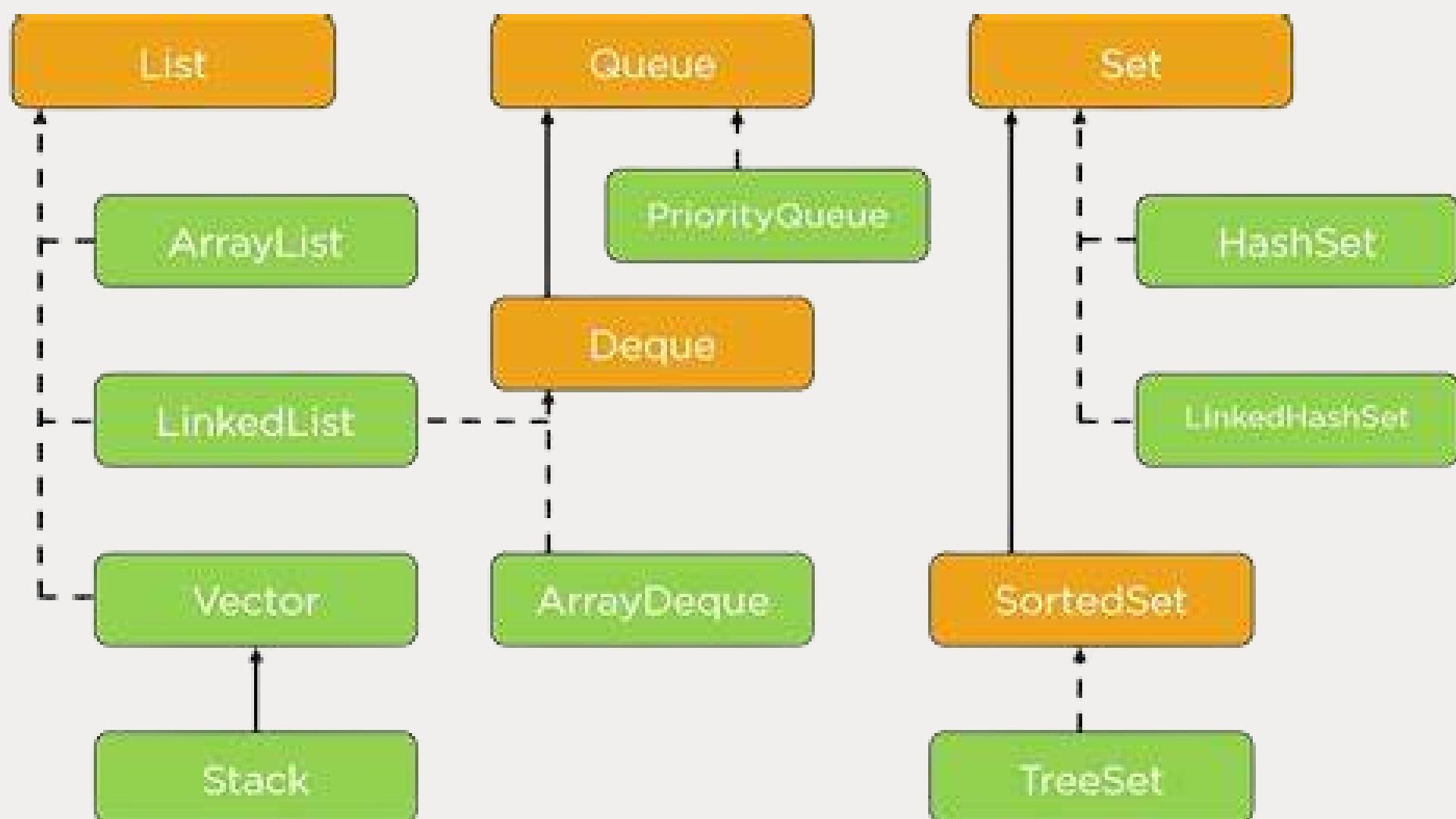
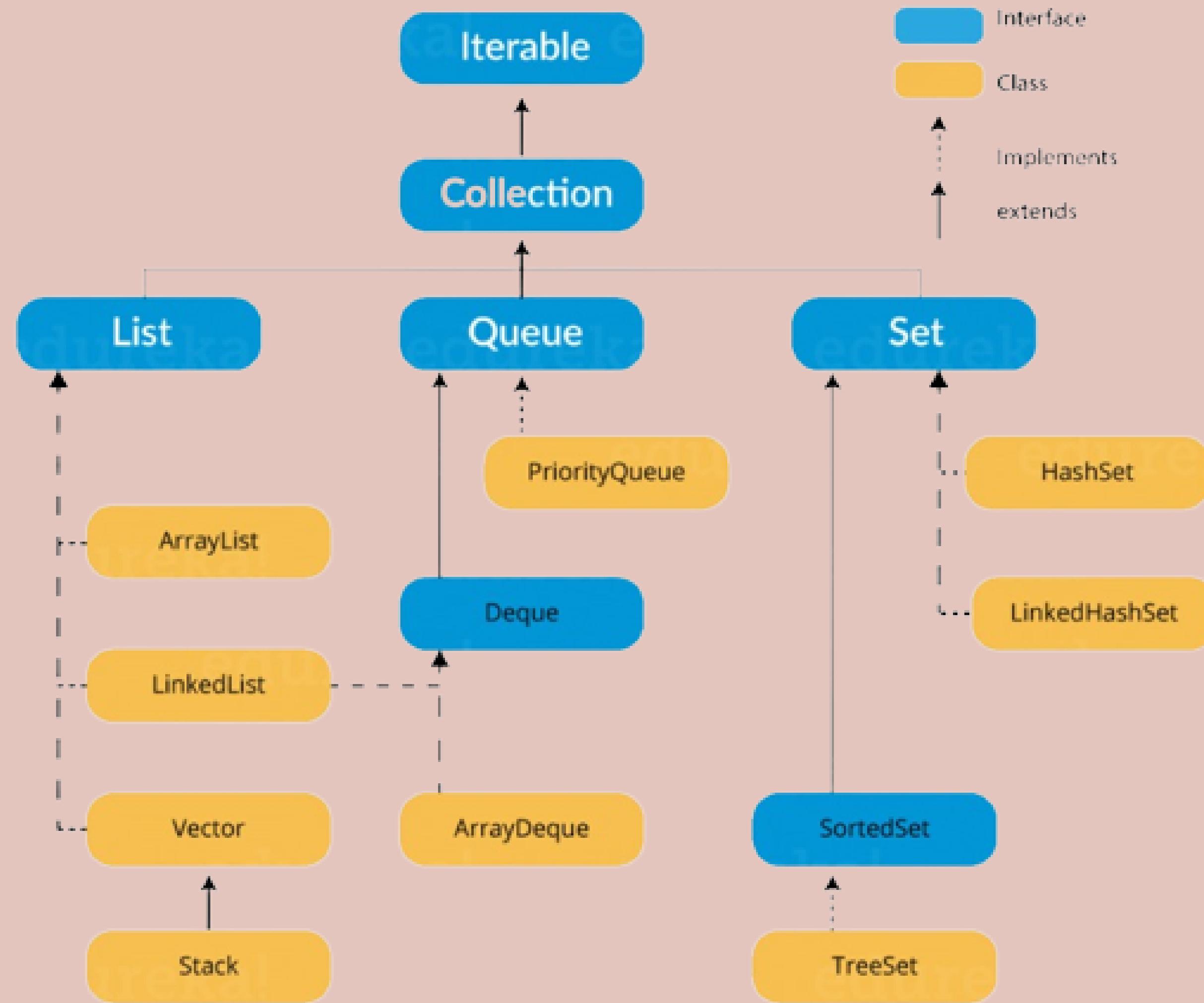


Java COLLECTIONS

Java Collections



son un conjunto de interfaces y clases que proporcionan estructuras de datos para almacenar, organizar y manipular grupos de objetos. Estas colecciones hacen que el manejo de datos sea más sencillo y eficiente, ya que ofrecen diferentes implementaciones optimizadas para diferentes tipos de operaciones.



List, Set, Queue

- List: Una colección ordenada que permite duplicados. Los elementos se pueden acceder mediante un índice. Algunas implementaciones conocidas son ArrayList y LinkedList.
- Set: Una colección que no permite duplicados. No se garantiza un orden específico para los elementos. Algunas implementaciones populares son HashSet y TreeSet.
- Queue: Una colección que representa una cola, donde los elementos se insertan al final y se eliminan del principio. Algunas implementaciones son PriorityQueue y LinkedList.

Ejemplo ArrayList

```
import java.util.ArrayList;

public class EjemploArrayList {
    public static void main(String[] args) {
        // Crear un ArrayList de enteros
        ArrayList<Integer> miArrayList = new ArrayList<>();

        // Agregar elementos al ArrayList
        miArrayList.add(10);
        miArrayList.add(20);
        miArrayList.add(30);

        // Acceder a elementos del ArrayList
        int primerElemento = miArrayList.get(0); // 10

        // Iterar sobre el ArrayList
        for (int numero : miArrayList) {
            System.out.println(numero);
        }
    }
}
```

Ejemplo LinkedList

```
import java.util.LinkedList;

public class EjemploLinkedList {
    public static void main(String[] args) {
        // Crear una LinkedList de enteros
        LinkedList<Integer> miLinkedList = new LinkedList<>();

        // Agregar elementos a la LinkedList
        miLinkedList.add(10);
        miLinkedList.add(20);
        miLinkedList.add(30);

        // Acceder a elementos de la LinkedList
        int primerElemento = miLinkedList.get(0); // 10

        // Iterar sobre la LinkedList
        for (int numero : miLinkedList) {
            System.out.println(numero);
        }
    }
}
```

CUADRO COMPARATIVO

	ArrayList	LinkedList
Estructura interna	Utiliza un array para almacenar los elementos.	Utiliza una lista doblemente enlazada de nodos.
Tiempo de acceso	Acceso rápido mediante índices ($O(1)$).	Acceso más lento mediante índices ($O(n)$).
Uso de memoria	Menor consumo de memoria.	Mayor consumo de memoria debido a las referencias adicionales.
Operaciones recomendadas	Lectura y búsqueda de elementos.	Inserciones y eliminaciones frecuentes en posiciones intermedias.