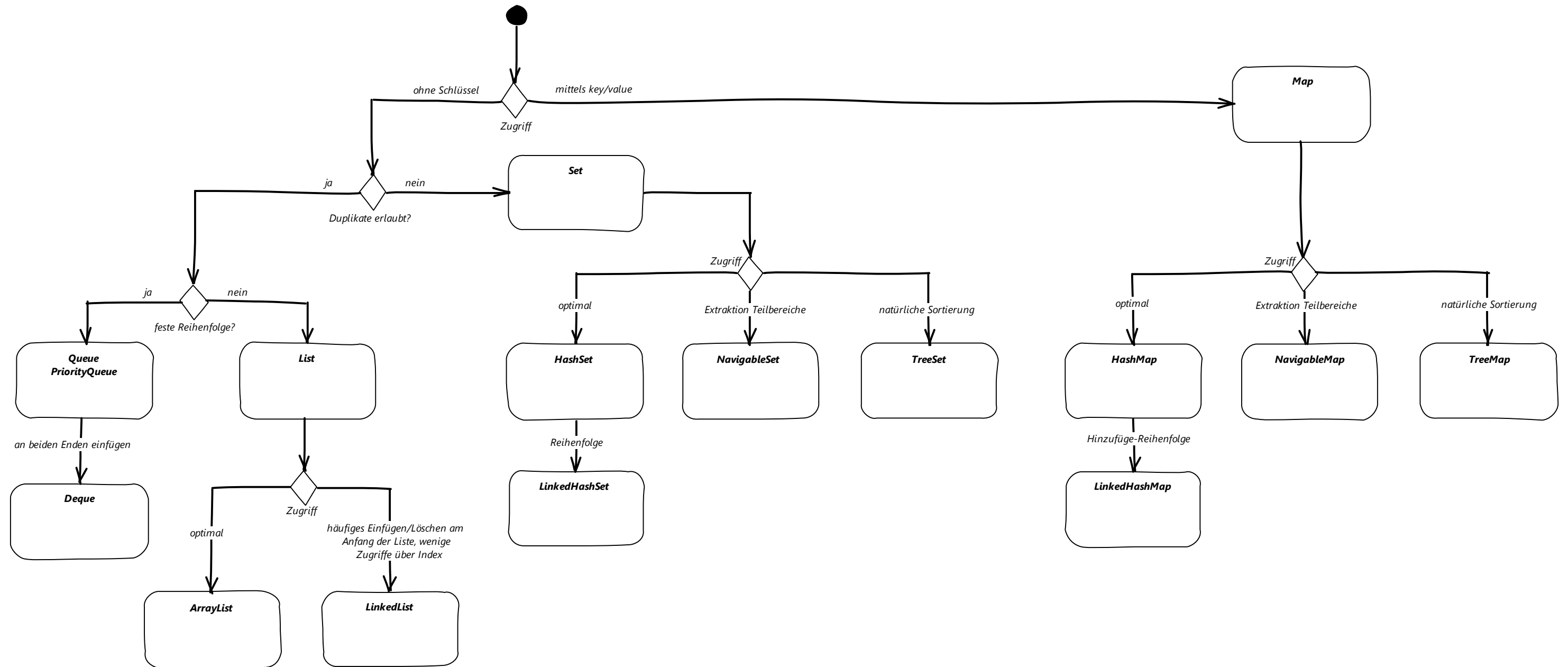


Collection

Wie entscheidet man sich für welche Collection? Hier ein mögliches Entscheidungsdiagramm zur Auswahl der passenden Collection.

Die Legacy-Collections (*Vector*, *Stack*, *HashTable*) wurden weggelassen.



Speicherreihenfolge

Eine wichtige Eigenschaft ist die Speicherreihenfolge:

- Eine List speichert in der Eingabereihenfolge; die Ausgabereihenfolge entspricht der Eingabereihenfolge (List ⇒ Einfügereihenfolge).
- HashSet speichert in zufälliger Reihenfolge. Die Ausgabereihenfolge ist also zufällig (Hash ⇒ zufällige Reihenfolge).
- LinkedHashMap und LinkedHashMap oder LinkedList sind geordnet in der Reihenfolge des Hinzufügens und in der natürlichen Reihenfolge, falls Comparable implementiert wurde (Linked ⇒ Einfügereihenfolge).
- TreeSet, TreeMap und PriorityQueue speichern in der natürlichen Reihenfolge, also lexikographisch (nach Alphabet) oder numerisch. Die Ausgabe erfolgt also sortiert nach Buchstaben oder nach Zahlen (Tree ⇒ natürliche Reihenfolge).

Eigenschaften der Collection-Typen

Interface	Klasse	Geordnet	Sortiert	Legacy (veraltet)
List	ArrayList	durch Index	nein	nein
	LinkedList	durch Index	nein	nein
	Vector	durch Index	nein	ja
	Stack	durch Index	nein	Ja
Set	HashSet	nein	nein	nein
	LinkedHashSet	Einfügereihenfolge	nein	nein
	TreeSet	ja (default Alphabet)	Durch die natürliche Reihenfolge oder durch selbst definierte Regeln	nein
	NavigableSet	ja (default Alphabet)	Durch die natürliche Reihenfolge oder durch selbst definierte Regeln	nein
Map	HashMap	nein	nein	nein
	LinkedHashMap	Einfügereihenfolge	nein	nein
	TreeMap	ja (default Alphabet)	Durch die natürliche Reihenfolge oder durch selbst definierte Regeln	nein
	HashTable	nein	nein	ja
	NavigableMap	ja (default Alphabet)	Durch die natürliche Reihenfolge oder durch selbst definierte Regeln	nein
Queue	PriorityQueue	ja	Nach vorgegebener Reihenfolge	nein
	Deque	ja	Nach vorgegebener Reihenfolge	nein

Unter Legacy-Collections versteht man Collections, die aus Kompatibilitätsgründen weiter unterstützt werden, aber in neuen Programmen nicht genutzt werden sollten. Dazu gehören *Vector*, *Stack* und *HashTable*.

Weiterführende oder ergänzende Informationen

- Oracle The Java™ Tutorials, [Trail: Collections](#)
- Java ist auch eine Insel, [Einführung in Datenstrukturen und Algorithmen](#)
- Java Hungry: [Top 50 Java Collections Interview Questions and Answers](#)
- Lars Vogel: [Java Collections - Tutorial](#)