**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Hochschule Mannheim**

Paul-Wittsack-Straße 10

68163 Mannheim

Coding Aufgabe PR2 TUT

**9. April 2024**

# ÜBERSICHT

Ihre Aufgabe besteht darin, eine generische Linked List in Java zu implementieren. Ihre Implementierung sollte verschiedene Operationen unterstützen, wie das Hinzufügen von Elementen, das Entfernen von Elementen, das Abrufen von Elementen an bestimmten Positionen und das Überprüfen, ob ein bestimmtes Element in der Liste vorhanden ist.

# ZIELE

1. Implementieren Sie eine generische Klasse LinkedList, die das ListInterface implementiert und die folgenden Methoden unterstützt:
   1. void add(E element): Fügt ein Element am Ende der Liste hinzu.
   2. void add(int index, E element): Fügt ein Element an der angegebenen Position in die Liste ein.
   3. E get(int index): Gibt das Element an der angegebenen Position in der Liste zurück.
   4. boolean contains(E element): Überprüft, ob das angegebene Element in der Liste vorhanden ist.
   5. int size(): Gibt die Anzahl der Elemente in der Liste zurück.
   6. void remove(int index): Entfernt das Element an der angegebenen Position aus der Liste.
   7. void clear(): Löscht alle Elemente aus der Liste.
   8. void printList(): Textuelle Ausgabe der gesamten Liste auf der Konsole.
   9. void clone(): Erzeugt einen tiefen Klon der Liste
   10. void toString(): Liefert die gesamte Liste als String.
2. Testen Sie Ihre Implementierung, indem Sie verschiedene Szenarien testen, einschließlich Hinzufügen, Entfernen, Abrufen von Elementen und Überprüfen der Größe.
3. Kommentieren Sie Ihren Code ausführlich im JavaDoc Format, um die Funktionsweise der Methoden zu erklären und das Verständnis zu erleichtern.

# SPEZIFIKATIONEN

Die Methoden des ListInterface Interface sollen unverändert bleiben und exakt gemäß ihrer Notation übernommen werden.