

```

Testlauf Daniel Ofosuene - Anzahl der Elemente: 100000
Füllen für die Listen (Anfügen am Ende)
Zeit für die LinkedList (ms): 6
Zeit für die ArrayList (ms): 4

Löschen der Hälfte der Elemente
Zeit für die LinkedList (ms): 6
Zeit für die ArrayList (ms): 317

Füllen für die Listen (Anfügen am Beginn)
Zeit für die LinkedList (ms): 5
Zeit für die ArrayList (ms): 1231

Aufsummieren mit Hilfe des Index
Zeit für die LinkedList (ms, Summe = 7417214): 24077
Zeit für die ArrayList (ms, Summe = 7415336): 3

Aufsummieren mit Hilfe des Iterators
Zeit für die LinkedList (ms, Summe = 7417214): 4
Zeit für die ArrayList (ms, Summe = 7415336): 2

Auflisten mit Hilfe des Index
Zeit für die LinkedList (ms): 14900
Zeit für die ArrayList (ms): 1889

Auflisten mit Hilfe des Iterators
Zeit für die LinkedList (ms): 1701
Zeit für die ArrayList (ms): 1972

Auflisten mit Hilfe StringBuilder
Zeit für die LinkedList (ms): 10
Zeit für die ArrayList (ms): 5

```

1. Füllen am Ende einfügen:
 - a. LinkedList und ArrayList ca. Gleich, weil add() automatisch immer am Ende hinzufügt
2. Löschen der Hälfte der Elemente:
 - a. LinkedList schneller weil nur Vorgänger und Nachgänger Information geändert werden muss
 - b. ArrayList langsamer, weil intern extra immer ein neues Array erstellt werden muss
3. Füllen am Beginn:
 - a. LinkedList schneller, weil Vorgänger und Nachgänger Information bekannt sind und damit es leichter die erste Stelle zu finden
 - b. Wie bei 2.) es muss extra ein neues Array erstellt werden aber diesmal mit Platzhalter für das neue Array der dann hinzugefügt werden muss
4. Aufsummieren mit Hilfe von Index:
 - a. LinkedList langsamer, weil bei LinkedList Zugriff mit Index nicht direkt möglich ist. Es wird immer von Beginn begonnen und Nachgänger und Vorgänger überprüft bis man zum gegebenen Element kommt
 - b. ArrayList schneller es wird mit Index direkt auf das Element zugegriffen
5. Aufsummieren mit Hilfe von Iterator:
 - a. Beide ca gleich schneller weil beide von Beginn der Liste beginnen
6. Auflisten mit Hilfe des Index:
 - a. LinkedList wieder langsamer wegen 4.a.

- b. ArrayList auch wegen Grund in 4.b aber dauert bisschen länger weil es extra in String Variable gespeichert wird
- 7. Beide wieder gleich lang wegen Grund in 5.a
- 8. Stringbuilder arbeitet intern mit Array deswegen ist es bei ArrayList schneller