

Interpretation

Unterschied zwischen ArrayList und LinkedList

In diesem Dokument wird auf die Unterschiede von ArrayListen und LinkedListen mit Praxisbezug eingegangen. Grundsätzlich implementieren beide Listen das List interface und fungieren somit sehr ähnlich. Trotzdem gibt es einige relevante Unterschiede.

AddEnd

```
AddEnd for ArrayList: 4ms  
AddEnd for LinkedList: 18ms
```

An dem gemessenen Ergebnis erkennt man, dass die ArrayList deutlich effektiver ist. Sie benötigt nur ein viertel der Zeit um die Operationen durchzuführen. Dies liegt

daran, dass bei der Linked List alle Teilnehmer untereinander verbunden (=gelinked) sind und somit das hinzufügen von Elementen sehr aufwendig ist.

AddFirst

Hier kann man genau das Gegenteil des zuerst angesprochenen Effektes sehen. Dadurch das ein Element an den Anfang hinzugefügt wird und die einzelnen

```
AddFirst for ArrayList: 45ms  
AddFirst for LinkedList: 11ms
```

Elemente miteinander verbunden sind benötigt die LinkedList wesentlich weniger Zeit als die ArrayList.

DeleteHalf

```
DeleteHalf for ArrayList: 25ms  
DeleteHalf for LinkedList: 168ms
```

Hier wird die Hälfte der angegebenen Liste gelöscht, die ArrayList ist deutlich effizienter und schneller als die Linked List weil auch hier die

Elemente erneut miteinander gelinked werden müssen.

SummeIterator

Hier wird die Summe der Elemente mithilfe eines iterators berechnet. Man sieht sofort, dass dies bei der LinkedList äußerst

effektiv ist und extrem schnell geht. Auch bei der ArrayList funktioniert es relativ gut.

```
SummeIterator for ArrayList: 6ms  
SummeIterator for LinkedList: 1ms
```

SummeIndex

```
SummeIndex for ArrayList: 4ms  
SummeIndex for LinkedList: 179ms
```

und soviel ineffizienter ist.

Hier sieht man, dass der Index für Arraylisten ein wenig schneller ist, allerdings bei LinkedListen unglaublich viel länger braucht

ListIterator

Hier wird eine Liste mithilfe eines Iterators durchgegangen und die Elemente in einen String geschrieben.

```
ListIterator for ArrayList: 12ms  
ListIterator for LinkedList: 16ms
```

ListIndex

```
ListIndex for ArrayList: 9ms  
ListIndex for LinkedList: 6ms
```

Hier wird die Liste mithilfe eines Indexes durchgegangen. Im Gegensatz zum Iterator ist hier die LinkedList effektiver und schneller.