**Praktikum 3 – Verkettete Liste**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Wie werden Elemente der Liste hinzugefügt?

* Neu hinzugefügte Elemente referenzieren auf das nächste Element in der Liste, solange die Liste nicht leer und somit first != null ist.

Wodurch wird das Ende einer einfach verketteten Liste gekennzeichnet?

* Durch null

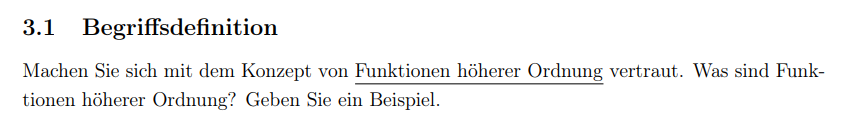
Wann ist die verkettete Liste leer?

* Wenn der erste Knoten == null ist

Wann ist eine einfach verkettete Liste anstelle einer MutableList (bezogen auf Kotlin) geeigneter und wann nicht?

* Vorteil: Elemente können schneller hinzugefügt oder gelöscht werden, da nur Referenzen der Nachbarn geändert werden müssen
* Nachteil: auslesen einer bestimmten Stelle der Liste ist langsamer, da man durch die komplette Schleife iterieren muss

Somit sollte Entscheidung von Art und Anzahl an Zugriffen festgelegt werden



In Kotlin wird eine Funktion, die eine Funktion als Parameter akzeptiert oder eine Funktion zurückgeben kann, als Funktion **höherer Ordnung bezeichnet**. Anstelle von Integer, String oder Array als Parameter für die Funktion übergeben wir anonyme Funktionen oder Lambdas. Häufig werden Lambdas der Einfachheit halber als Parameter in Kotlin-Funktionen übergeben.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

O(1) 🡪 Laufzeit linear für jedes Element z.B. Liste mit 1000Elementen jedes Element wird einmal angesprochen, also unabhängig von der Problemgröße

O(n) 🡪 n-fache Problemgröße benötigt n-fache Zeit

O(n^2) 🡪 n-fache Problemgröße benötigt n^2-fache Zeit

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

addFirst(data: T) 🡪 O(1)

addlast(data: T) 🡪(O(n))

size() 🡪 O(n)

getFirst🡪 O(1)

get(index: Int) 🡪 O(n)

forEach (action: (T) 🡪 Unit) 🡪 O(n^2)