Dynamische Datenstrukturen (engl. collections):

Nachteile von Arrays:

* fixe Größe
* Zugriffe selbst verwalten 🡪 Aufwand 🡪 Fehler

Vorteile von Collections:

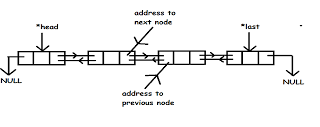
* Performance
* Namen standardisiert 🡪 API

|  |  |
| --- | --- |
| Grundformen 🡪 Interfaces | Konkrete Implementierungen |
| List =Array | ArrayList, LinkedList |
| Set =Menge | HashSet, TreeSet |
| Map =Key-Value | HashMap, TreeMap |
| Queue/Deque =FiFo/LiFo | Ungeordnet / geordnet |

ArrayList:

* Einfach mit .add ein neues Objekt hinzufügen

LinkedList:

* Jeder Eintrag zeigt auf den nächsten und den davor
* Jeder Eintrag zeigt auf ein Objekt
* Es gibt noch 2 Zeiger, die auf den ersten und den letzten Eintrag zeigen

Set:

* Die Referenzen werden nicht intern abgespeichert, sie müssen berechnet werden
* Man muss nicht jeden Eintrag durchgehen, sondern hat direkt den berechneten

Map:

* Nicht nummerische Referenzen (nicht Index, sondern Key)
* Die gespeicherten Daten heißen Values