

Datastrukturer

Datastrukturer

- I kender `array` og `ArrayList`
- I Java findes masser af andre datastrukturer med hver deres egenskaber
- I dag kigger på vi på `List`, `Set` og `Map`. Det er alle tre interfaces, der specificerer egenskaber for de klasser, der implementerer dem

List

- Lister er kendetegnet ved, at de holder et antal objekter i en bestemt rækkefølge
- Det vil sige at hvert objekt har et indexnummer
- Som hovedregel tillader lister dubletter, altså to ens objekter i listen
- I Java bruger vi fx `ArrayList<E>` og `LinkedList<E>`

Opgave

- Klon projektet <https://github.com/Dat1Cphbusiness/Collections>
- Find klassen `PlayingWithCollections`
- Udfyld metoden `playWithLists()` og kød koden løbende

Set

- Set er kendetegnet ved, at de holder et antal objekter, men ikke nødvendigvis i en bestemt rækkefølge
- Objekterne har ikke noget indexnummer
- Set tillader ikke dubletter, men udelukkende unikke objekter
- I Java anvender vi fx `HashSet` og `TreeSet`

Vigtige metoder

- Alle set bruger metoden `equals()`
- `HashSet` bruger metoden `hashCode()`
- `TreeSet` bruger metoden `compareTo()`

Det er derfor afgørende at have en fornuftig implementation af de tre metoder.

Opgave

- Udfyld metoden `playWithSets()` og kød koden løbende
- Lav din egen klasse med nogle attributter. Det kan fx være en `Player` eller en `Creature` eller noget andet
- Giv din klasse fornuftige `equals()`, `hashCode()` og `compareTo()` –metoder
- Afprøv din klasse ved at putte objekter af den i henholdsvis et `TreeSet` og et `HashSet`.

Map

- Maps er kendetegnet ved, at de holder par af objekter, men ikke nødvendigvis i et bestemt rækkefølge
- Objekterne har intet indexnummer
- Parrene består af en *key* og en *value*. Hver key skal være unik
- Maps bruger metoden `equals()` på key-objekterne for at bestemme om to keys er ens
- I Java anvender vi fx `HashMap` og `TreeMap`

Opgave

- Udfyld metoden `playWithMaps()` og kør koden løbende