cphbusiness

Datastrukturer

Java Collections Framework

Datastrukturer – hvordan flere dataelementer opbevares som en enkelt enhed

- I kender `array` og `ArrayList`

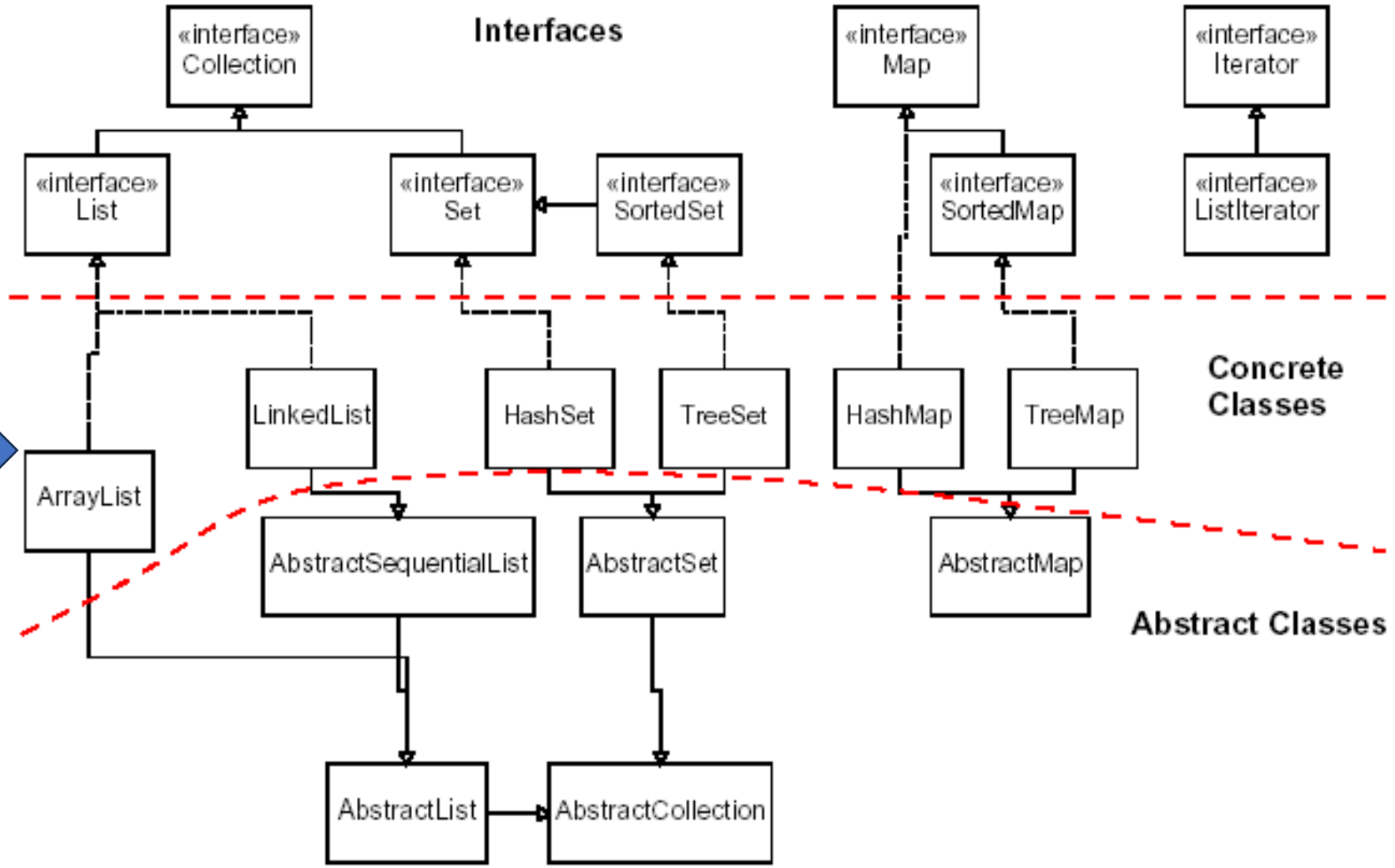
- I Java findes masser af andre datastrukturer med hver deres egenskaber
  - Collections (The Java™ Tutorials)
  - Collections Framework Overview

- I dag kigger på vi på `List`, `Set` og `Map`. Det er alle tre interfaces, der specificerer egenskaber for de klasser, der implementerer dem

cphbusiness

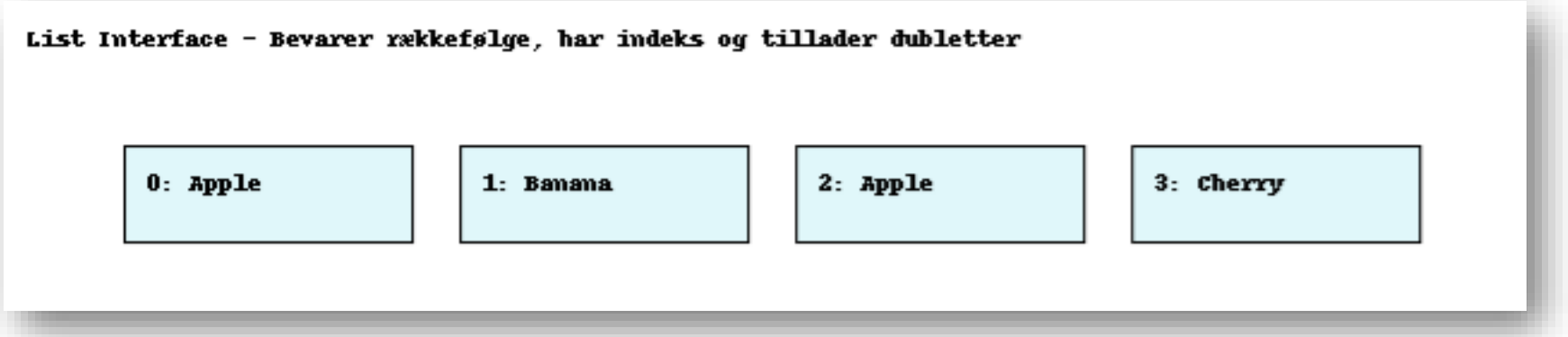
Java Collections Framework

Vi skal arbejde med de konkrete klasser, men også forstå sammenhængen med deres interfaces



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` og `LinkedList`



Opgave

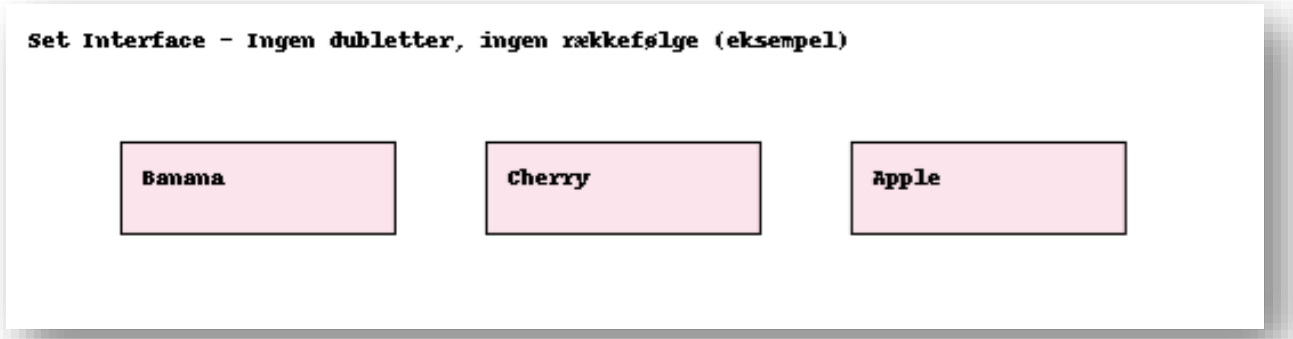
- Klon projektet
  https://github.com/Dat1Cphbusiness/CollectionsF2025

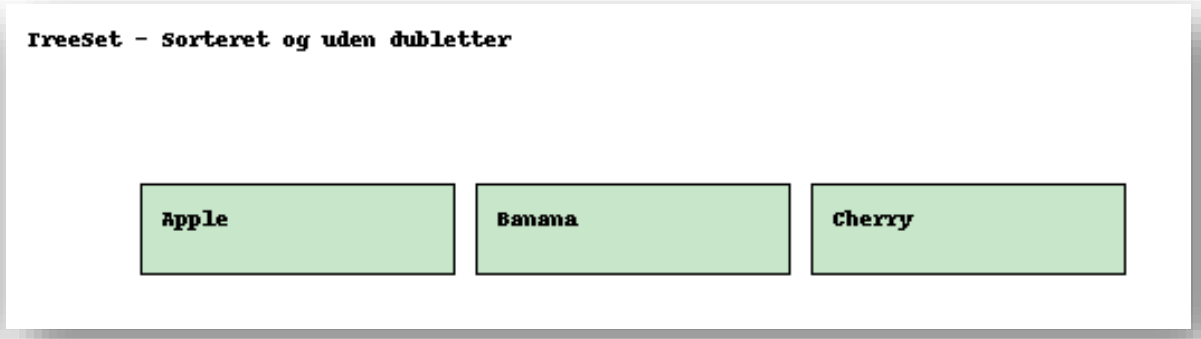
- Find klassen **`PlayingWithCollections`**

- Udfyld metoden **`playWithLists()`** og kør 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 bruger vi fx `HashSet` (usorteret) og `TreeSet` (sorteret)





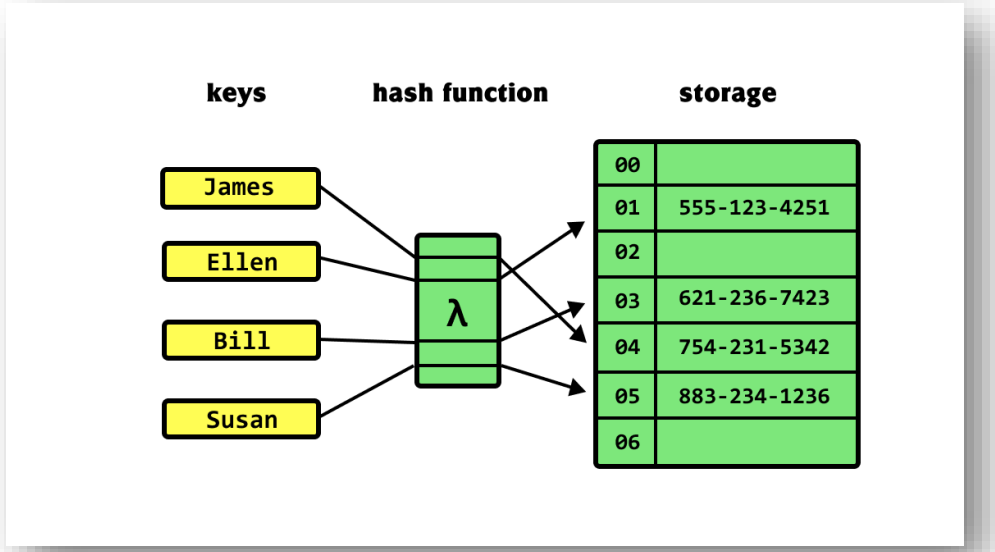
Vigtige metoder

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

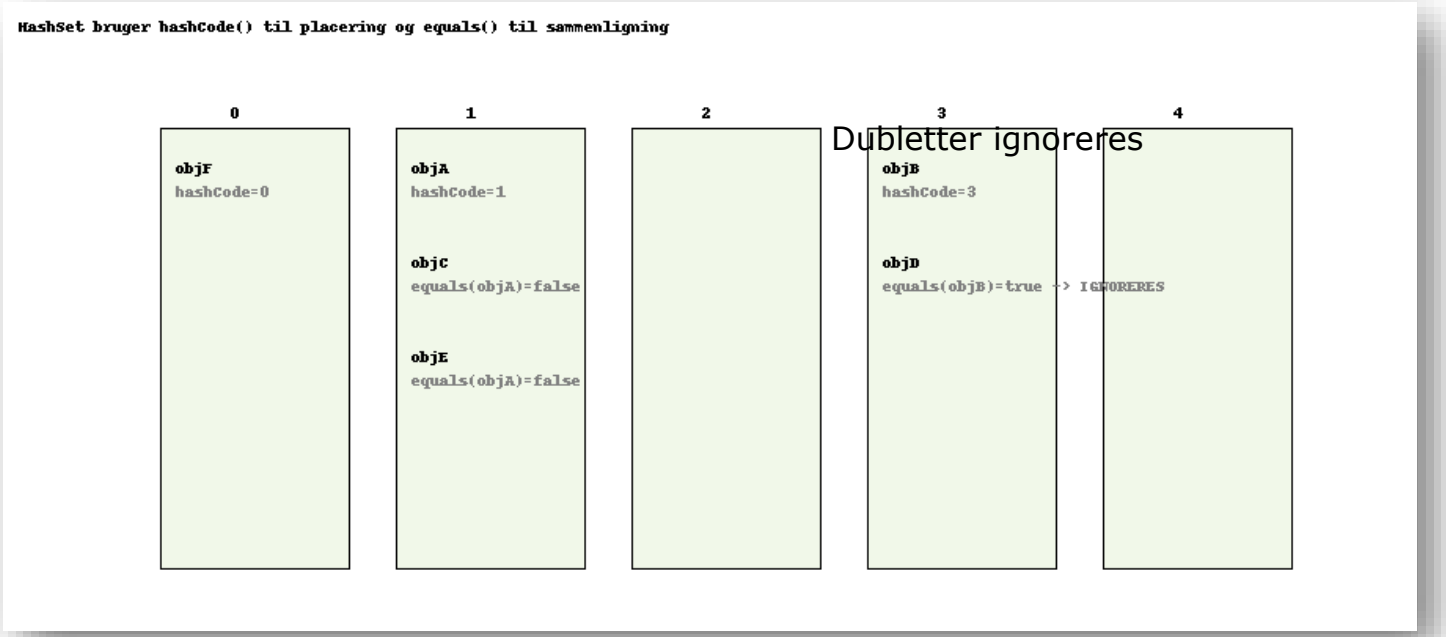
- Det er derfor afgørende at have en fornuftig implementering af de tre metoder.

Vigtige metoder

`hashcode()` bruges til at beregne placering (index) i datastruktur:



`equals()` bruges til tjek af dubletter:



Dubletter ignoreres

Vigtige metoder

`compareTo()` bruges til sortering

The *compareTo* method **compares the current object with the object sent as a parameter**.

When implementing it, we need to make sure that the method returns:

- A positive integer, if the current object is greater than the parameter object
- A negative integer, if the current object is less than the parameter object
- Zero, if the current object is equal to the parameter object

In mathematics, we call this a sign or a signum function:

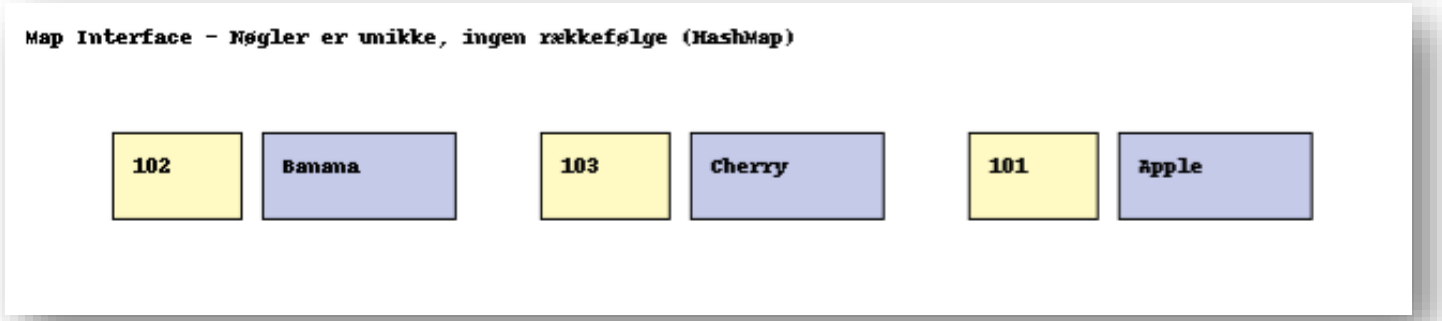
$$\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

Opgave

- Udfyld metoden **`playWithSets()`** og kør 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 **`equals()`** til at bestemme om to keys er ens.
- I Java anvender vi fx **`HashMap`** (usorteret) og **`TreeMap`** (sorteret)

Opgave

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