> velkommen til gruppetime i in1010!

> gruppe 1 - design

> dagens tema: arv og subklasser part II
> polymorfi
> komposisjon vs. arv
> object
> konstruktører

Oppvarmingsoppgave - hva printes her?

```
1   public class Main {
        Run | Debug
2       public static void main(String[] ags) {
3           int[] ints = new int[10];
4           int whatValueIsThis = ints[0];
5
6           System.out.println(whatValueIsThis);
7       }
8   }
9
```

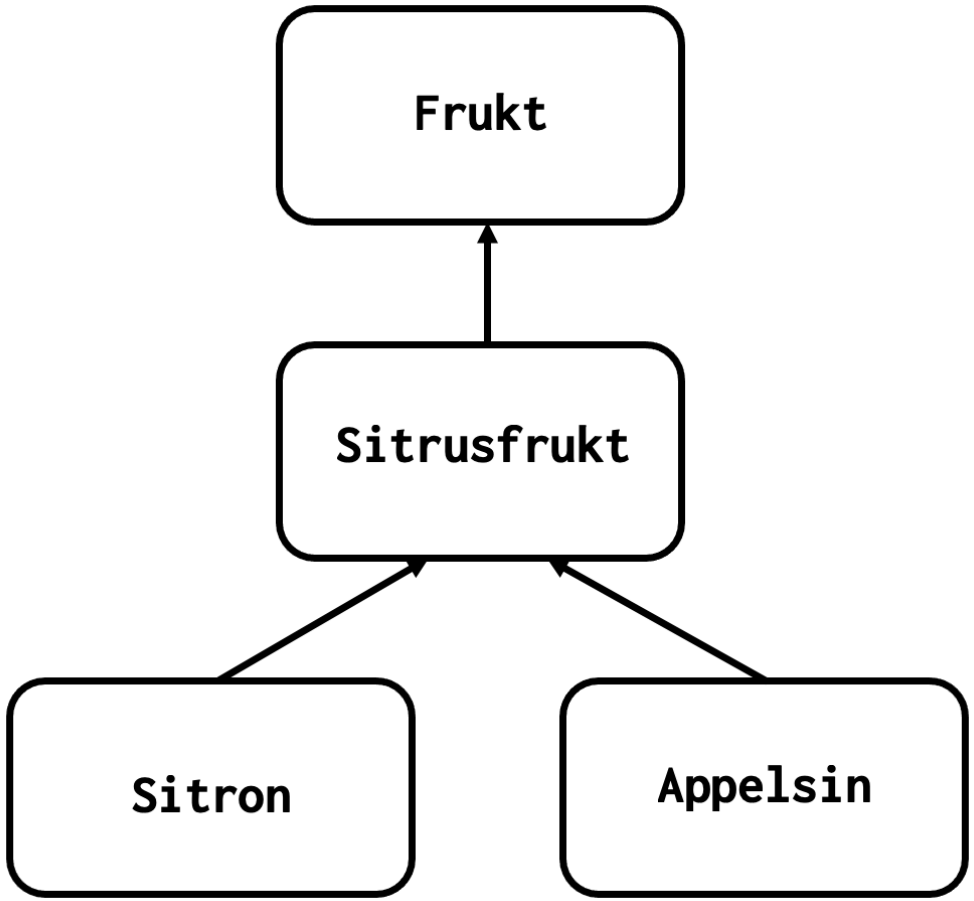
Litt repetisjon fra sist

```
Frukt frukten = new Frukt();

Sitrusfrukt sitrus = new Appelsin();

Appelsin appelsin = new Sitron();

Sitron sitron = new Sitrusfrukt();
```



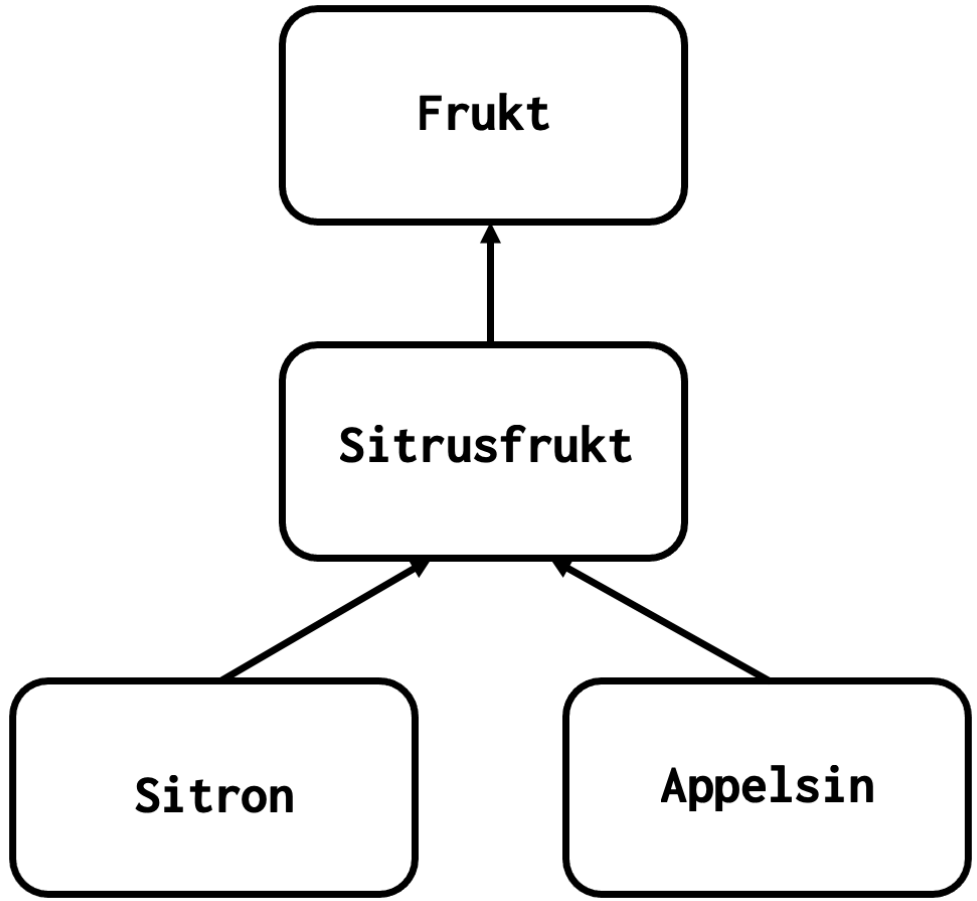
Litt repetisjon fra sist

Upcasting

```
Sitrusfrukt sitrus = new Sitron();
Frukt frukt = sitrus;
```

Downcasting

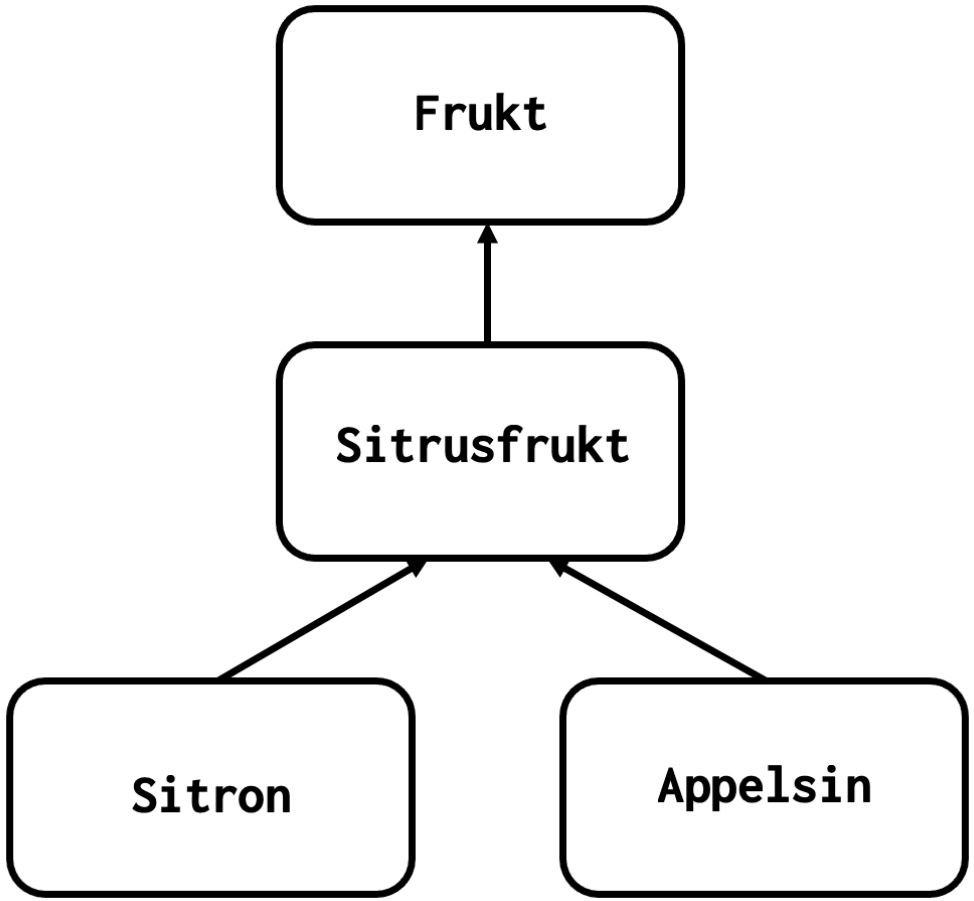
```
Sitrusfrukt sitrus = new Sitron();
Sitron sitron = (Sitron) sitrus;
```



Dagens tema: Polymorfi

Hvilke metoder har dette objektet tilgang på? ->

```
Frukt frukt = new Sitrusfrukt();
```



Overloading

- En klasse har lov til å ha metoder med samme navn, så lenge de har forskjellige parametre

Arv vs. komposisjon

Vi lager et program for et biblioteksystem. Burde vi bruke arv eller komposisjon i følgende tilfeller:

- Klasser for bøker av forskjellige sjangre
- Klasser for hver bokhylle i biblioteket med hvilke bøker den inneholder
- Klassen «Bibliotek» som inneholder alle bokhyllene
- Klassen «Barnebibliotek», som er et bibliotek som bare inneholder barnebøker

Object

Alle klassers superklasse er Object. Denne inneholder følgende metoder:

- toString
- equals
- hashCode

```
class Punkt2 {
   protected int x, y;
   Punkt2(int x0, int y0) {
      x = x0;  y = y0;
   }
   @Override
   public String toString() {
      return ("x = "+x+" y = "+y);
   }
   @Override
   public boolean equals(Object o) {
       if (!(o instanceof Punkt2))
          return false;
       Punkt2 p = (Punkt2) o;
       return x == p.x && y == p.y;
   }
}
```

Når vi lager overrides så må signaturen være lik, derfor equals(Object o) og ikke equals(Punkt2 p)

Konstruktører ...

...

Egenarbeid

in1010 -> v25 -> grupper -> gruppe 11 -> uke 4