



Forelesning 12

INF101 - 21/2 2023





Objektorientert programmering



Abstraction

- Fokuser på det viktigste og gjør det enkelt



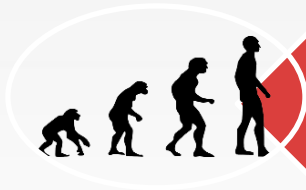
Modularity

- Lag kode som er byggeklosser



Encapsulation

- Skjul detaljer, koden blir tryggere å bruke



Gjenbruk av kode

- Arv - Gjenbruk av klasser
- Polymorphism – metoder virker på flere typer

SIDE 2



Encapsulation

- Handler om å unngå bugs.
- Hver del av koden skal ha ansvar for en ting.
- Hvordan kan vi sørge for at annen kode ikke messer med det vår kode har ansvar for?





Trafikklys

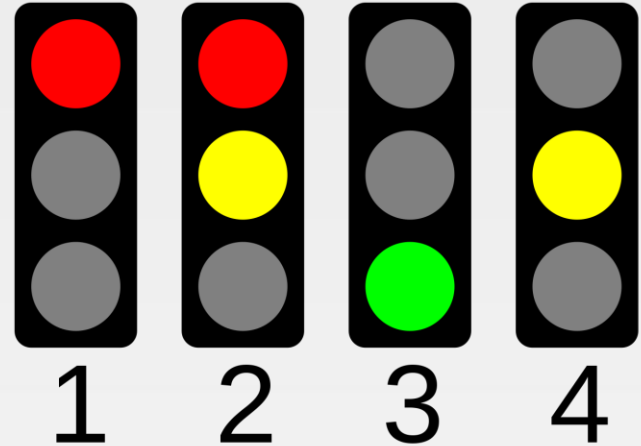
- Vi skal implementere styringssystem for et trafikklys
- Vi har derfor bestemt oss for å lage en klasse for trafikklys
- Senere kunne vi laget et objekt for hvert lys i et kryss osv.





Abstraksjon

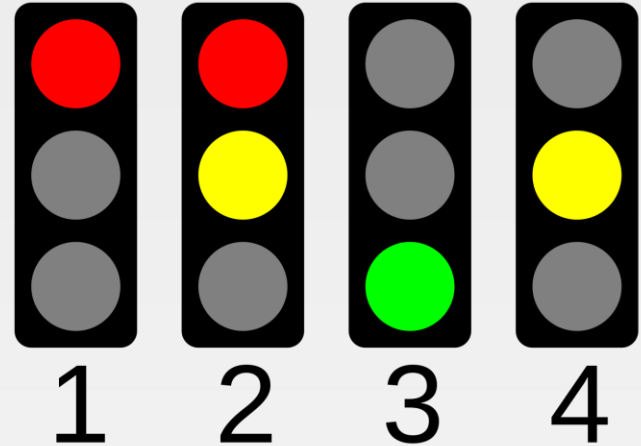
- State er hvilke lys som er på
- Behavior er overgang til neste lyssignal
- Merk at noen tilstander er ikke lovlige





Abstraksjon

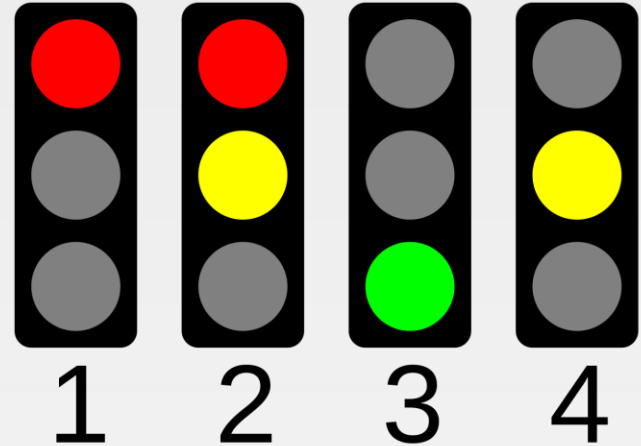
- Hvordan skal vi beskrive et trafikklys?
 - 3D modell av lyset?
 - Bilde med rød, gul og grønn farge?
 - True/false verdier som sier hvilken farge som lyser
 - Tall 1,2,3 eller 4 som sier hvilken tilstand det er (se bildet)





Implementasjon

- La oss implementere et trafikklys objekt





Encapsulation

- Kun det som er nødvendig er tilgjengelig.
- Kompliserte detaljer brukes kun i en klasse.
 - Det blir lettere å bruke.
 - Hvis noe går feil er det lettere å finne feilen.
 - Det blir mindre sjanser for feil.
- Enkleste måte å oppnå encapsulation er access modifiers.





Access modifier

Hvilke andre klasser får bruke denne?

	public	protected	default	Private
Samme klasse	Ja	Ja	Ja	Ja
Samme pakke arvet	Ja	Ja	Ja	Nei
Samme pakke	Ja	Ja	Ja	Nei
Annen pakke arvet	Ja	Ja	Nei	Nei
Annen pakke	Ja	Nei	Nei	Nei





Sortering og søking

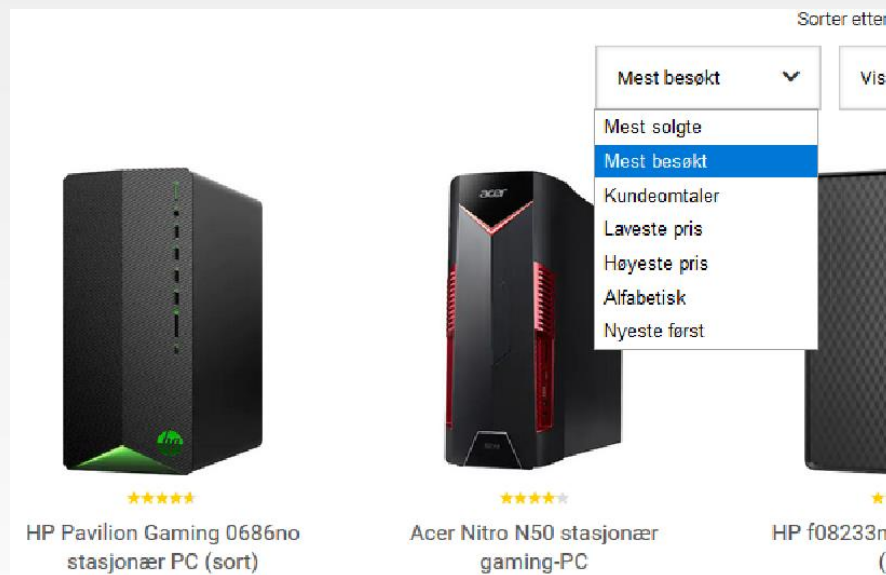
- Hvorfor sorterer vi?
 - Oversiktlig liste
 - Det blir lettere å finne ting.
- Sortering brukes ofte for å gjøre kode raskere
 - Dette er pensum for INF102
- Vi prøvde `Collections.sort()` på forrige forelesning, hvordan virket egentlig den?





Sortering

- Hva er sorter rekkefølge?
 - Største først?
 - Minste først?
 - Varer sorteres etter:
 - Pris
 - Alfabetisk
 - Anmeldelser
 - Mest solgt





Sortering

- Kan vi endre hvordan Objekter sorteres?
 - Ja! Vi bruker compareTo() metoden.
 - Interfacene Comparable og Comparator
 - Sortering bruker metoden compareTo().
 - Vi må implementere metoden toString for å få System.out.println() til å virke som vi vil.
 - Vi må implementere metoden compareTo for å få sortering til å virke som vi vil.





Searching

- `List.indexOf(item)` virker på alle lister
 - `indexOf` bruker metoden `equals()`
- Sortering virker bare på `Comparable`.
 - Eller med `Comparator`
 - Bruker metoden `compareTo()`
- `binarySearch()` virker kun hvis input allerede er sortert. (Pensum i INF102)





La oss prøve soting

- Vi implementerer compareTo i CellPosition
- Vi implementerer en Comparator eller to.





```
public class Thanks{
    private String text;

    public Thanks(String text){
        this.text = text;
    }

    public String toString(){
        return this.text;
    }

    public static void main(String[] args){
        Thanks end = new Thanks("Takk for forelesning!");
        System.out.println(end);
    }
}
```

