Händelsestyrda program och Repetition

Vecka 3, Bildserie 2

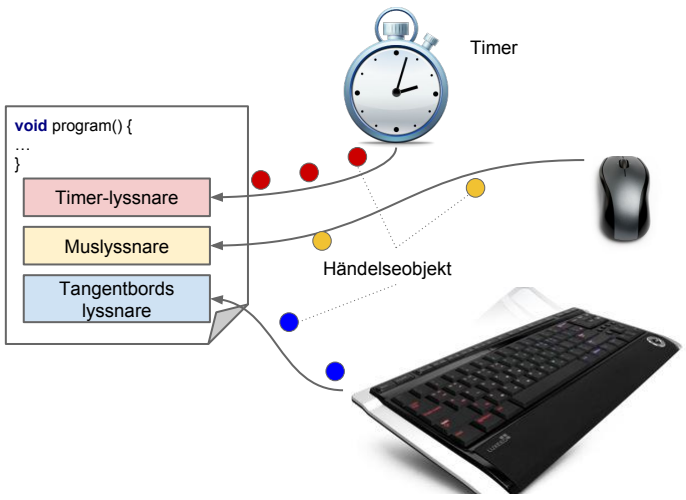
Innehåll

- Händelsestyrda program
- Mer om labben (Simulation)
- Saker vi missat och/eller saker som kan förklaras mer (eller igen)

Att Läsa i Boken

- Det som finns i boken (15.1-15.4) är onödigt komplicerat
  - Se mina kodexempel i stället

Händelsestyrda Program



Inmatningen till våra program har hittills använt en kommandorad.

- De flesta program fungerar inte på det sättet …
- .. de är istället **händelsestyrda**

Ett händelsestyrt program kommer att ta emot indata då olika typer av yttre händelser inträffar

- Man klickar på musen, trycker på en tangent eller använder en timer
  - Exakt hur det går till kan vi inte gå in på, vi säger att ett **händelsesystem** sköter det hela
  - Vi konstaterar att en "händelse" i form av ett objekt kommer att skickas som en parameter till en **"lyssnar"-metod** då vi t.ex. trycket på en tangent
    - Objektet innehåller information om vilken typ av händelse som inträffat och data om händelsen
      - T.ex. En mushändelse och muspekarens position, en tangenthändelse och vilken tangent, etc
  - En klass som deklarerar en lyssnarmetod kallar vi för en **lyssnare**
- En lyssnare (ett objekt av lyssnarklassen) måste registrera sig som mottagare av händelseobjekt
  - Så att händelsesystemet vet vart händelseobjekten skall skickas (nämligen till lyssnarmetoden som objektet har).

En Timer

```
final int delay = 1000;

void update(){
    // Do the program logic
}

void initEvents() {
    // Create Timer
    Timer timer = new Timer(delay, e -> {
       update();
       repaint();  // Will call paint (in Java 2D)
   });
   timer.start();
}
```

En timer är ett objekt som med jämna mellanrum genererar en händelse.

- Vi måste skapa en Timer ungefär som Random eller Scanner.

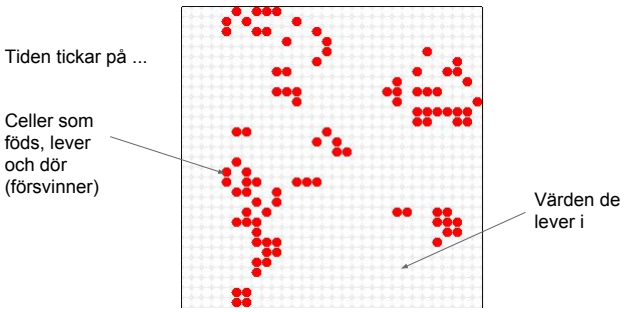
Vi gör en metod, initEvents(), som skapar och startar timern

- .. och sätter initial fördröjning och tid mellan timers signal (i ms)
- .. samt startar timern.
- Efter detta kommer metoderna update() och repaint() (som is sin tur anropar paint()) att anropas varje gång timern "går av".
    - I detta fall 1 ggn/sek.

För den nyfikne

- Uttrycket e -> { …. } betecknar en namnlös metod (kallas lambda-uttryck)
    - e är parametern till metoden (kommer att skapas och initieras automatiskt av händelsesystemet.
    - { .. } är metodkroppen
- Metoden anropas automatiskt av timern

Inför Övning 3



Vi gör en version av Game of Life

- Fortsatt övning på funktionell nedbrytning och abstraktion.
- Mer tekniska saker: enum, matriser, generiska metoder.