

Ukeoppgave i uke 10, INF2440 – v2015

I ukeoppgavene denne uka skal vi først se på problemer med at en tråd i en lokal klasse vi har laget terminerer før vi 'får tak i data' som ligger i den tråden.

Oppgave 1 – erstattet hacket med noe bedre (?) når vi starter opp TegnUt for å få tegnet punktene og innhyllinga.

På forelesninga Uke 8 ble det foreslått flg:

```
TegnUt tu = new TegnUt (this, koHyll);  
for (int i =1; i > 0; i--)i+=2; // forsinkelse
```

a) TegnUt tegner ut punktene og innhyllinga den antar er i en ArrayList <Integer> koHyll. .

Forsinkelseslinja gjør at klassen med main-tråden ikke terminerer og fjernes før TegnUt-tråden får lest data fra main-tråden og tegnet dem ut.

Problemet er at pekeren til både en punktmengde som er i et objekt som angis her med : this, og en DKI (den konvekse innhyllinga) som her er i en ArrayList (eller en IntList) . I begge tilfeller overføres bare pekere til to objekter, **ikke** dataene inne i objektene. Hvis da det/de objektene som virkelig inneholder data terminerer og ryddes bort av søppeltømmeren, er det ikke noe data og hente. Hacket som foreslåes i Oblig3 (sitert ovenfor) virker, men er vel ikke en bra/'riktig' løsning. Prøv flg. 2 måter å forbedre dette hacket:

- Overfør til klassen TegnUt en Semafor vent, som er deklartert og initiert i main-tråden. Skriv kode for det før og etter oppstart av TegnUt-objektet fra main, og modifier klassen TegnUt, slik at den sier `release()`; på denne semaforen når den er ferdig med uttegninga. Prøv også å minimere og maksimere tegnevinduet etterpå.
- Legg inn en kall på input fra klassen Scanner som da venter på input fra tastaturet rett etter `TegnUt tu = new TegnUt (this, koHyll);` som en erstatning for forsinkelseslinja. (`Scanner sc = new Scanner(System.in); sc.nextLine();`)Main tråden vil da ikke terminere før du har gitt et CR. Prøv også her å minimere og maksimere tegnevinduet før du gjør det.

Hvilken av de tre måtene foretrekker du og hvorfor?

Oppgave 2 – Oblig3 –parallell versjon

Vi så på forelesninga i Uke 9 at oppgaven å finne DKI besto av 3 faser, og at dere skulle parallellisere den siste fasen, det å rekursivt finne de resterende punktene på DKI når de 4 første (minx, maxx og de to punktene som var i størst avstand over og under linja maxx-minx) var funnet .

Hvis dere har løst fase 3 parallelt, kan dere forsøke dere med også en parallell løsning på fase 1 – det å finne maxx og minx; ellers jobber dere videre med fase 3.