

# DA112T Föreläsning 9 extra materiel

## Inledning

Logik är läran om korrekt resonemang – att kunna dra korrekta slutledningar utifrån det man vet. Vi gör detta ständigt utan att tänka på det men inom programmering så blir det ofta mer påtagligt.

Det finns informell logik som är logiskt resonemang uttryckt med naturligt språk. Det är detta vi använder till vardags när vi argumenterar eller resonerar om vardagliga frågor. Här hamnar även det vi kallar pseudokod även om detta är en mer strikt form av naturligt språk.

Sedan har vi formell logik där formen för det korrekta resonemanget är fångat i ett formellt språk som exempelvis satslogik. De formella språken är begränsade av att man måste följa språkets regler och man kan inte uttrycka lika mycket med ett formellt språk som med naturligt språk. Java och andra programmeringsspråk kan ses som formella språk.

## Pseudokod

Pseudokod som kommer att bli ett oundvikligt redskap när du gör lite mer komplexa program. Pseudokod är egentligen bara naturligt språk strukturerat lite kortare och liknande det sätt man strukturerar källkod på. Du använder dig av satser och indrag men en sats kan vara betydligt större i sitt omfång än vad som är vanligt i det formella programmeringsspråket. Indragen blir mycket viktiga i pseudokod då dessa visar vilka satser som är underordnade andra satser. I pseudokod använder man sig sällan av något tecken för att starta eller sluta ett block.

Pseudokod används för att bryta ner ett problem i mindre delar innan man sätter sig och skriver källkod. Detta gör man för att upptäcka hur man lämpligt kan dela in koden i metoder och kontrollstrukturer samt för att identifiera nödvändig indata. Pseudokod kan också användas för att testa en algoritm innan man kodar denna. Genom att kontrollera att pseudokoden löser problemet så sparar man tid innan man kodar och slipper risken att en massa kod är gjord förgäves och ändå måste slängas.

Antag att vi har följande problem: Du är på biblioteket och vill låna boken ”Populärmusik från Vittula” av Mikael Niemi. Tyvärr så sker det underhåll på deras elektroniska katalog så du kan inte via denna ta reda på om boken finns eller ej utan måste använda dig av en lite ålderdomligare metod att helt enkelt se om boken finns i hyllan eller inte. Finns boken i hyllan så tar du den och går till disken för att låna denna annars så går du till disken för att reservera boken. Då det är en skönlitterär bok så vet vi att den finns på hylla Hc.

Om vi nu skall skriva pseudokod för hur vi löser problemet ovan så ställs vi inför några frågor: Vilka delar kan vi bryta ner det här problemet i? Hur mycket detaljer skall vi visa? Hur vi vet att boken finns är ganska lätt men hur vet vi om boken inte finns?

De två första frågorna hör ihop. Ju fler detaljer vi vill visa desto fler delar måste vi bryta ner problemet i. Vi vill inte göra en för abstrakt pseudokod:

```
gå till hyllan där boken borde finnas
OM boken finns
    Ta boken och låna den
ANNARS
    Reservera boken
```

Detta hjälper ju inte oss särskilt mycket. Det är för generellt. Mer detaljer då:

```
TILLS vi kommer till första hyllan
    Lyft fot 1
    Placera fot 1 framför fot 2
    Lyft fot 2
    Placera framför fot 1
...
```

Det blir lite väl mycket detaljer.

Konsten är alltså att hitta en nivå där man får med nödvändiga detaljer men kan utelämna sådant som är självklart. Vilka är våra viktiga delar då: vi måste hitta rätt hylla, vi måste hitta rätt författare och vi måste hitta rätt titel. Ordningen på ett bibliotek hjälper oss då vi kan utgå från att hyllorna är sorterade efter det vanliga bibliotekssystemet och att författare är sorterade i bokstavsordning efter efternamn och böckerna för varje författare efter titel. Om vi under vårt letande kommer till en författare som finns efter Niemi i bokstavsordning eller en bok av Niemi som finns efter den vi söker så fanns inte boken "Populärmusik från Vittula".

Fortsättning på nästa sida...

Så om vi utgår från stegen ovan:

```
1  gå till närmsta hylla
2  TILLS hylltypen är Hc //vi förutsätter att det finns en hylla Hc
3    Gå till nästa hylla
4  //vi har nu hittat hylla Hc
5  FÖR VARJE författare på Hc
6    OM namn är Mikael Niemi
7      FÖR VARJE bok av Mikael Niemi
8        OM titel är "Populärmusik från Vittula"
9          ta boken
10         AVBRYT för varje bok av Mikael Niemi
11       AVBRYT för varje författare på Hc
12  ANNARS
13    OM författare kommer efter Niemi i bokstavsordning
14      AVBRYT för varje författare på Hc
15 //vi har nu hittat boken eller konstaterat att den inte fanns
16 gå till lånedisken
17 OM vi har boken med oss
18   låna boken
19 ANNARS
20   reservera boken
```

Radnumreringen är inte nödvändig utan gör det bara lättare att referera till delar av pseudokoden nu i denna text. Så låt oss nu analysera denna pseudokod och se vilka slutsatser vi kan dra från vår algoritm för att låna eller reservera en viss bok.

I den första loopen (2) så gör vi ett antagande att det finns minst en hylla av typen Hc. Om det inte skulle finnas någon hylla av typen Hc så har vi en oändlig loop. Men då vi vet att det är ett vanligt bibliotek så kan detta vara ett rimligt antagande. Vi gör också ett antagande att det finns något sätt att ordna hyllorna i en ordning eller hålla reda på vilka vi tittat på eller inte. Vi har inte nästlat in senare händelser i denna loop utan när raden efter loopen startar (5) så kan vi utgå från att vi står vid första hyllan av typen Hc.

I nästa loop (5) så tittar vi på alla författare som finns på Hc. Villkoret är här inte "tills vi hittar Niemi" utan vi måste använda en break-sats för att avbryta loopen när vi hittat Niemi eller vet att Niemi inte finns (11) respektive (14). Det händer egentligen ingenting avgörande om vi skulle fortsätta att titta på alla författare efter att vi hittat boken eller vet att den inte finns men det är ganska onödigt. Om vi i denna loop hittar Niemi så startas en ny loop där vi tittar på titlar(7).

Loopen som tittar på titlar (7) avbryts om vi hittar rätt bok(10) men om boken vi söker inte finns så kommer vi att titta på alla böcker av Niemi även om vi gått förbi den sökta boken i bokstavsordning. Vi får här avgöra om det är nödvändigt att lägga in detta avbrott eller inte. Är det värt en extra selektion för varje titel jämfört med någon iteration för mycket?

När rad (16) inleds så har loopen som letar bland författare och titlar på Hc avslutats på något sätt och vi går till lånedisken då vi oavsett resultatet i looparna ovan skall dit. Väl framme vid lånedisken så avgör vi vad vi ska göra utifall vi har boken med oss eller inte. Detta kallas att man sätter en flagga och är vanligt när man har loopar som letar efter något. Alternativet hade varit att flytta upp denna kod till loopen på följande sätt:

8	OM titel är "Populärmusik från Vittula"
9	ta boken
9.1	gå till lånedisken och låna boken
10	AVBRYT för varje bok av Mikael Niemi

samt

13	OM författare kommer efter Niemi i bokstavsordning
13.1	gå till lånedisken och reservera boken
14	AVBRYT för varje författare på Hc
15	//vi har nu hittat boken eller konstaterat att den inte fanns

Här har vi ganska korta följder beroende på om vi hittar boken eller inte och det blir fortfarande läsbar kod. Men om det hade inneburit längre kodavsnitt så hade metoden med en flagga varit bättre.

Lägg märke till hur viktigt det är att indragen ovan stämmer i pseudokoden för att avgöra vad som ingår i en viss loop. Om indragen inte stämmer blir pseudokoden fullständigt meningslös.

Motsvarigheter till reserverade ord skrivs ofta med versaler i pseudokod för att öka läsbarheten då dessa ord lätt försvinner i mängden vanliga ord annars.

Så som påpekats tidigare så är svårigheten att hitta den grad av detaljer som behövs. Det är bättre att börja med lite för lite detaljer för att se vad man behöver förtydliga när man skriver pseudokod än att börja med att grota ner sig i en pseudokod som i princip är källkod direkt och ta onödigt lång tid på sig. Det gäller att komma ihåg att pseudokoden skall vara en förenklad av källkoden.