

En kort introduktion till pseudokod och flödesschema

Pseudokod och/eller flödesscheman är det första steget till ett färdigt program. Med hjälp av pseudokod eller flödesschema gör man en "skiss" till programmet, men utan att använda programkod, dvs. utan att skriva i C++ eller något annat programmeringsspråk.

Pseudo betyder "falsk", "nästan" eller "oäkta". Pseudokod är alltså "nästan kod", inte riktig kod, utan ett mellanting mellan programkod och skriftspråk. Det är uppställt på ett speciellt sätt och skrivs enligt vissa regler och därför brukar man kalla det ett *formaliserat språk*.

Både flödesschemat och pseudokoden talar om i vilken ordning saker görs, vilka valmöjligheter som finns och vilka villkor som ska vara uppfyllda innan man går vidare. Kanske kan man likna ett flödesschema vid de pilar på golvet som många IKEA-varuhus är utrustade med. Följer man pilarna (dvs. IKEA's "program") ställs man inför en massa val. Ska jag köpa en soffa eller ett bord? En gryta eller en matta? Slutet av banan består av ett villkor: Om man har tillräckligt med pengar och lämnar in dem i kassan får man lämna varuhuset.

Ett flödesschema eller en pseudokod talar alltså om vad programmet egentligen gör, utan att gå in i på hur det ska uttryckas i ett programmeringsspråk.

Endast enkla instruktioner

För att datorn ska kunna lösa ett problem behöver den instruktioner och flera instruktioner tillsammans bildar ett program. Dessa instruktioner måste de vara enkla och elementära för att fungera. De måste säga exakt vad som ska göras och inte innehålla några outtalade förutsättningar eller underinstruktioner. En enkel instruktion är av typen "ta ett steg framåt" eller " $5\,664,786 + 822,61$ ". Komplexa instruktioner är t.ex. "öka företagets vinst" eller "vårda din bil". De talar inte om exakt hur man gör och är därför inte användbara i ett program.

Strukturelement

Både pseudokod och flödesschema använder s.k. strukturelement. Det finns tre slags strukturelement:

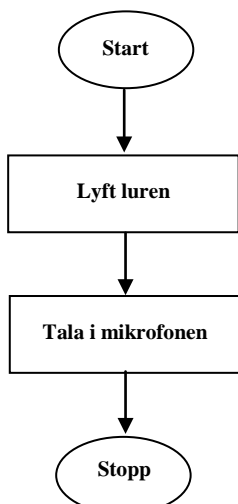
- Sekvens
- Selektion
- Repetition

Sekvens innebär att pseudokoden eller flödesschemat talar om i vilken ordning instruktioner ska utföras. Om pseudokoden handlar om hur man svarar i telefonen, kommer instruktionen "lyft luren" före instruktionen "säg hallå i mikrofonen". Att först säga hallå i mikrofonen och sedan lyfta luren är ingen bra lösning på hur man svarar i telefonen.

För oss människor är detta självklart, men pseudokoden och flödesschemat är avsedda för datorer, och en dator vet ingenting om hur världen fungerar. Sekvens visar man i pseudokod genom att skriva en instruktion efter den andra och ordningen räknas ovanifrån och neråt:

Start
Lyft luren
Tala i mikrofonen
Slut

I flödesschema visar man sekvens med pilar.



Selektion

Selektion betyder val, dvs. att man kan ta en av två vägar vid en punkt. Vilken väg man tar beror på om ett villkor är uppfyllt eller inte. Det låter kanske komplicerat men är egentligen enkelt.

I pseudokod kan man uttrycka ett val så här:

OM *det regnar ute*

 Ta på regnkläder

ANNARS

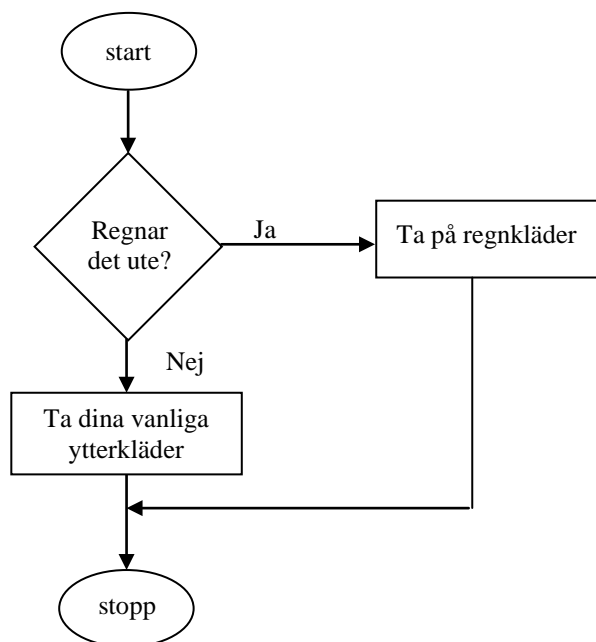
 Ta på dina vanliga ytterkläder

Villkoret i denna pseudokod är "det regnar ute". Är villkoret sant utför man instruktionen som finns på raden under OM: "Ta på regnkläder". Är villkoret falskt, dvs. om det inte regnar ute, utför man instruktionen som finns under ANNARS: "Ta på dina vanliga kläder". I det här fallet gäller valet vilka ytterkläder man ska ta på sig.

Observera att OM och ANNARS är skrivna med stora bokstäver. Det är helt avsiktligt och beror på att OM – ANNARS är ett strukturelement. Ett strukturelement ger struktur åt ett program och det finns ett mycket begränsat antal strukturelement att tillgå. Exempelvis kan man i flödesschema, pseudokod och färdiga program bara uttrycka selektion på två sätt, varav OM – ANNARS är det vanligaste. Det andra sättet lär vi oss längre fram. Strukturelementet OM – ANNARS finns inbyggt i både C++ och Java och datorn vet precis hur den ska behandla en selektion (ett val). OM – ANNARS skrivs i program som IF – ELSE.

Värt att notera är också att villkoren då skrivs i *kursiv* stil.

I ett flödesschema betecknar man ett val med en romb (en sned fyrkant). I romben står villkoret skrivet och därifrån går två pilar som betecknar sant eller falskt – ja eller nej. Ovanstående pseudokod skriven i ett flödesschema skulle se ut så här:



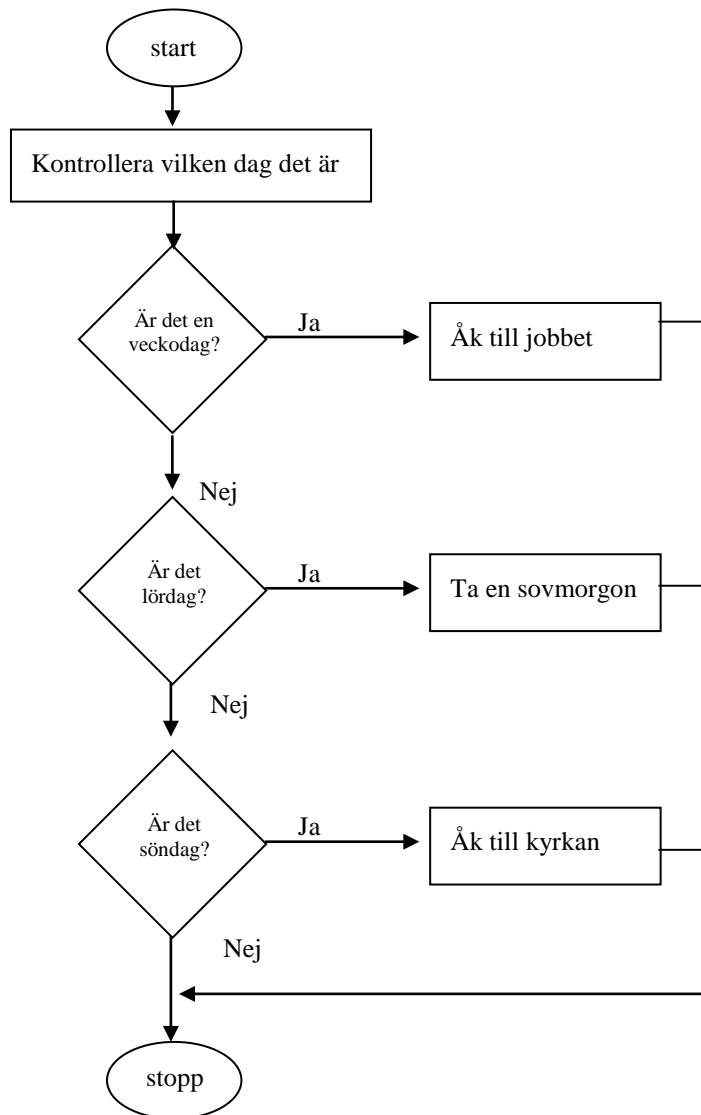
Endera tar man på sig regnkläder eller vanliga ytterkläder, inte båda, enligt detta flödesschema. När valet är klart och en instruktion är utförd sammanförs pilarna igen så att man kan utföra nästa instruktion i flödesschemat.

Man kan kombinera flera OM-satser efter varandra för att på så sätt göra ett selektion bland fler än två val. Exemplet nedan visar hur det går till:

Kontrollera vilken dag det är
OM *det är veckodag*
 Åk till jobbet
ANNARS
 OM *det är lördag*
 Ta en sovmorgon
ANNARS
 OM *det är söndag*
 Åk till kyrkan

Beroende på vilken dag det är, kommer en av OM-satserna att vara sann och utföras. Men istället för att, som i föregående exempel, skriva tre stycken envägsval efter varandra kan man kombinera ihop dessa till ett *flervägsval*. Ett flervägsval skrivs som "VÄLJ *fall* ur" och används på följande sätt i pseudokod och flödesschema.

Kontrollera vilken dag det är
VÄLJ *fall* ur
 dag veckodag: Åk till jobbet
 dag lördag: Ta en sovmorgon
 dag söndag: Åk till kyrkan



Repetition

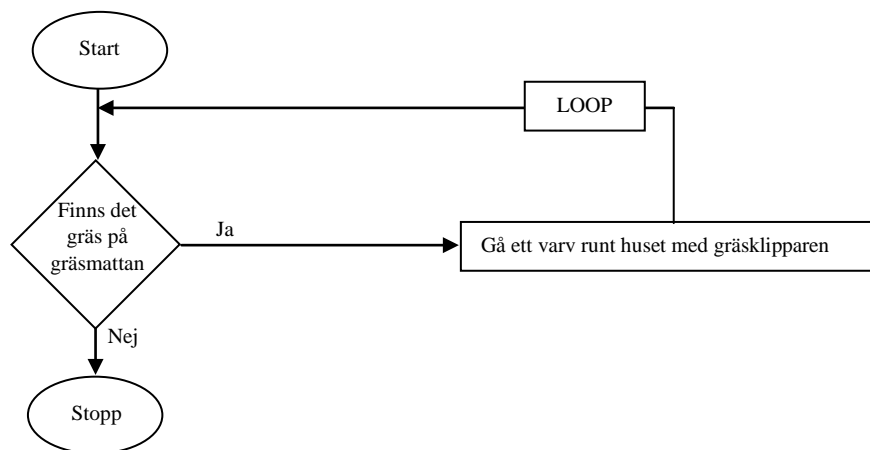
Repetition innebär att en instruktion utförs så länge ett villkor är sant. När villkoret är falskt upphör repetitionen. Repetition liknar alltså selektion, men vid en selektion kan en instruktion bara utföras en gång, till skillnad från repetition, där instruktionen kan utföras en eller flera gånger. Ett exempel på en repetition i en pseudokod är:

SÅ LÄNGE *det finns långt gräs på gräsmattan*

Gå ett varv runt huset med gräsklipparen

I det här fallet är **SÅ LÄNGE** strukturelementet och "*det finns långt gräs på gräsmattan*" villkoret. Om det är sant att det finns långt gräs utförs instruktionen som står under **SÅ LÄNGE**, "gå ett varv runt huset med gräsklipparen". Därefter kontrolleras villkoret igen och om det fortfarande är sant att det finns långt gräs på gräsmattan utförs instruktionen en gång till. Som du märker är det tänkbart att repetitionen bara kommer att utföras en gång. Det sker i det här fallet om gräsmattan är mycket smal eller om gräsklipparen är mycket bred. Det är också tänkbart att repetitionen aldrig kommer att utföras. Det sker om gräsmattan har en grässort som aldrig växer och alltid är kort, dvs. om villkoret alltid är falskt.

Uttryckt i ett flödesschema ser det ut så här



Observera att pilen går uppåt i flödesschemat, vilket är typiskt för repetitioner. Repetitioner kan uttryckas med tre olika strukturelement, varav vi bara tar upp **SÅ LÄNGE**. (De övriga kommer senare i kursen.)

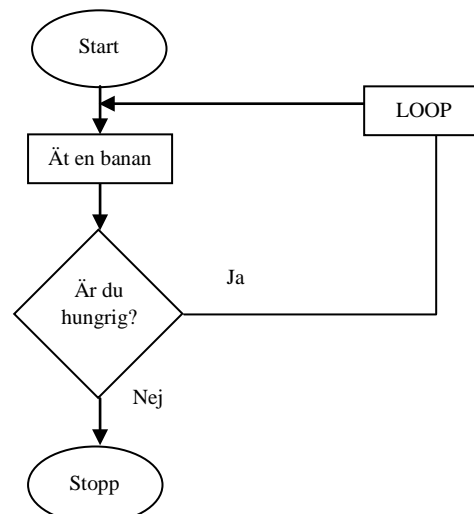
I programkod uttrycker man **SÅ LÄNGE** med ordet **WHILE** (i både C++ och Java).

Förtestad respektive eftertestad repetition

Man skiljer på två typer av repetition. Den *förtestade* repetitionen har vi precis gått igenom, dvs med hjälp av strukturelementet **SÅ LÄNGE**. En *förtestad* repetition innebär att villkoret kontrolleras först och om det uppfylls utförs slingan.

Den *eftertestade* repetitionen skrivs med strukturelementet **TILLS**. Här utförs alltid instruktionen/instruktionerna som ingår i slingan *minst en gång*, och sedan kontrolleras om villkoret fortfarande uppfylls. Det låter komplicerat, men om vi tittar på hur den eftertestade repetitionen skrivs i pseudokod och flödesschema klarnar det.

Ät en banan **TILLS** *du inte längre är hungrig*



Minst en banan kommer alltid att ätas, i och med att kontrollen görs efter instruktionen redan utförts. En eftertestad repetition skrivs som DO – WHILE i programkod.

Man kan även skriva eftertestade repetitioner på ett lite annat sätt:

UTFÖR		REPETERA
Ät en banan	eller	Ät en banan
SÅ LÄNGE <i>du är hungrig</i>		SÅ LÄNGE <i>du är hungrig</i>

Resultatet av koden och flödeschemat blir precis detsamma, det är bara ett annat sätt att skriva på.

Kombinationer

Selektioner (val), repetitioner (upprepningar) och fristående instruktioner kan förstås blandas i alla tänkbara kombinationer. Här har vi en instruktion och sedan en selektion:

Gå ut i skogen
OM *det finns kantareller*
 Plocka dem i en korg
ANNARS
 Fortsätt att gå

Man kan även använda fristående instruktioner, selektioner och repetitioner tillsammans:

Läs vad som står i tidningens nöjesbilaga
OM *det går en science fiction-film*
 Beställ biljetter till filmen
ANNARS
 Beställ biljetter till en äventyrsfilm
Gå till föreställningen
Köp godis
SÅ LÄNGE *det finns godis kvar*
 Ät godis