

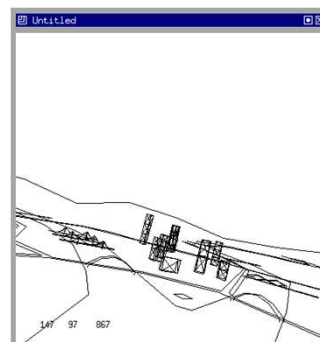
# Strukturerad Programmering

# Algorithms + Datastructures = Programs

[illegible]

# IOCCC : International Obfuscated C Code Contest

“It is an quasi-annual contest to see who can write the most unreadable, unintelligible, unmanageable, but legal C program.”



[<http://blog.aerjockey.com/post/iocccsim>]



Att kunna ett programspråk  
är **INTE** detsamma  
som att kunna programmera!

[Jfr : Trafikregler vs Bilkörning]

# JSP

## Jackson Structured Programming

## JSP-princip:

- Data kan beskrivas i **strukturdiagram**
- Programmet ska följa datat.

Algorithms +  
Datastructures =  
Programs

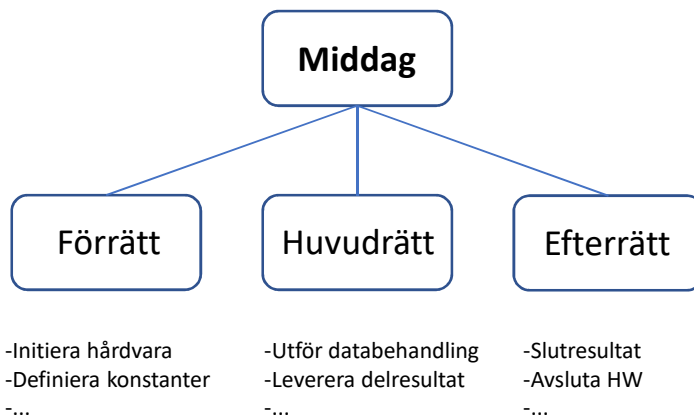
## JSP: Strukturdiagram Byggblock

Sekvens

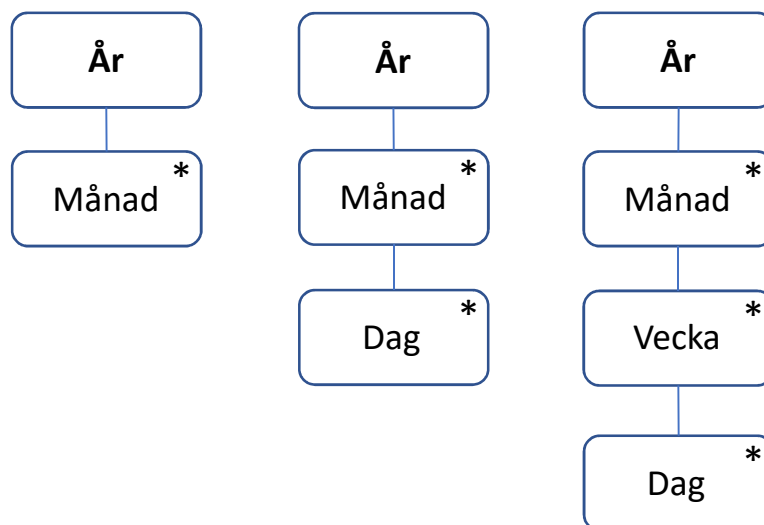
Iteration \*

Selektion <sup>○</sup>

**Sekvens:** först den ena, sen det andra och sen ...

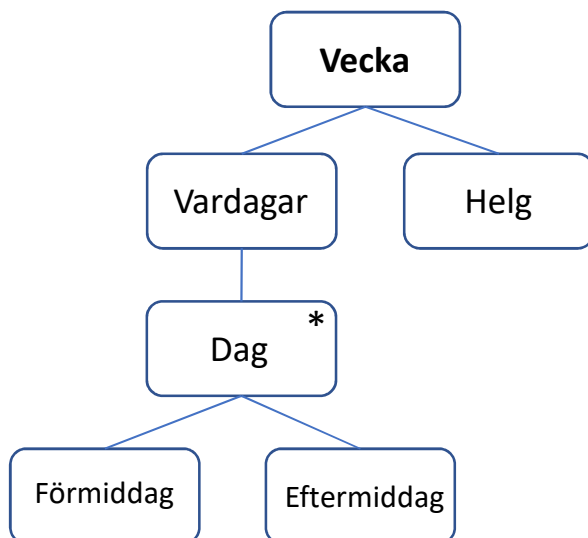


**Iteration:** om och om igen och igen ...

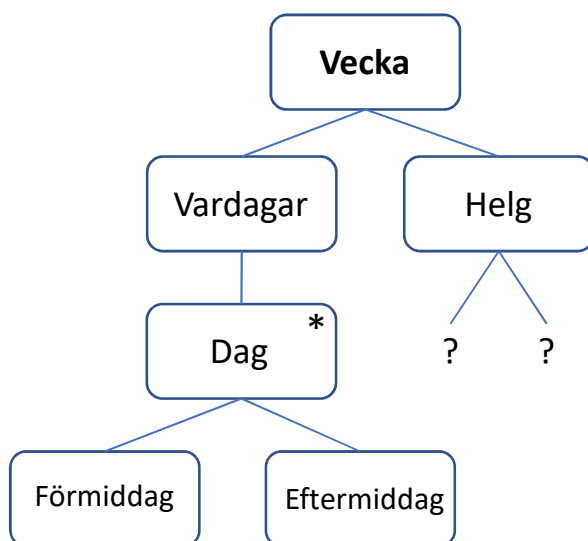


Iterationsdelen kan  
Utföras **NOLL** gånger!

## Kombinerad sekvens och iteration

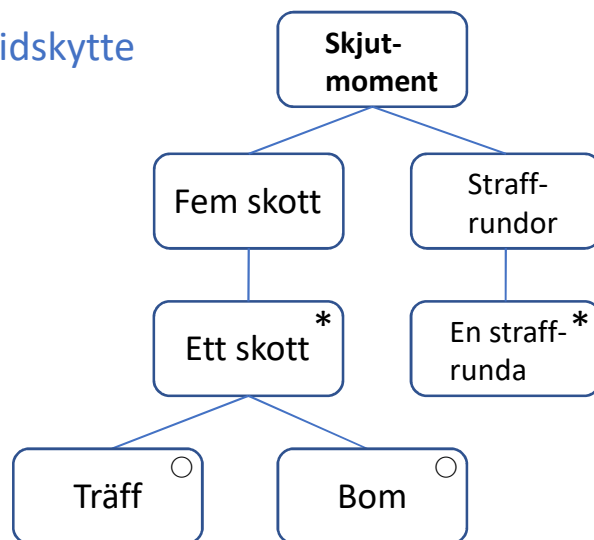


## Kombinerad sekvens och iteration



**Selektion:** det ena eller det andra eller ...

Skidskytte



Iterationsdelen kan  
Utföras **NOLL** gånger!

## JSP-Regler

En komponent får endast ha delar av samma typ:

- En **sekvens** ska endast bestå av **sekvensdelar**
- En **selektion** ska endast bestå av **selektionsdelar**
- En **iteration** ska endast bestå av en (1) **iterationsdel**

Delarna kan dock utgöras av komponenter av annan typ.

En iteration kan ske noll gånger:

- Tänk **while()** inte **if()**

En selektion måste ha minst två delar:

- Ett val måste ha mer än ett alternativ
- Ofta ett defaultalternativ också

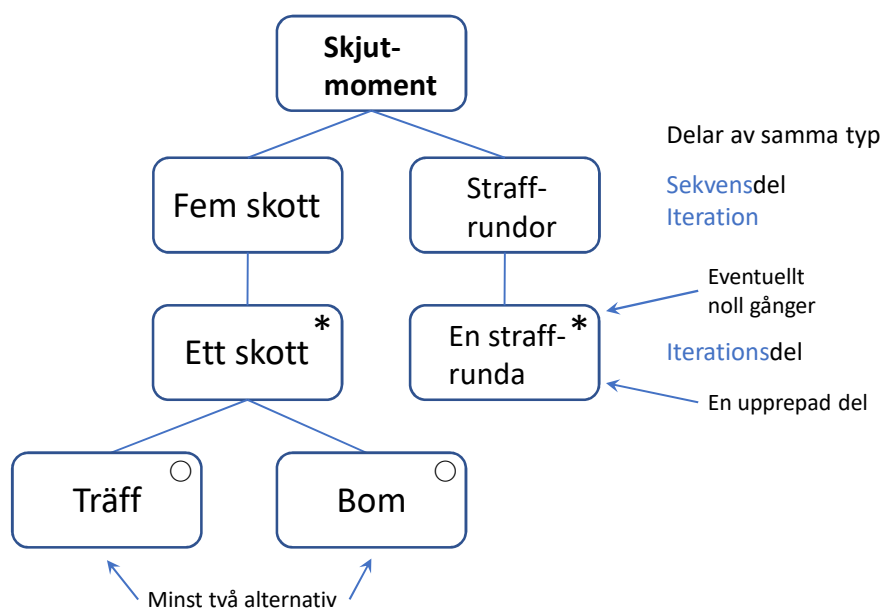
## Korrekt enligt JSP

Sekvens

Sekvensdel  
Iteration

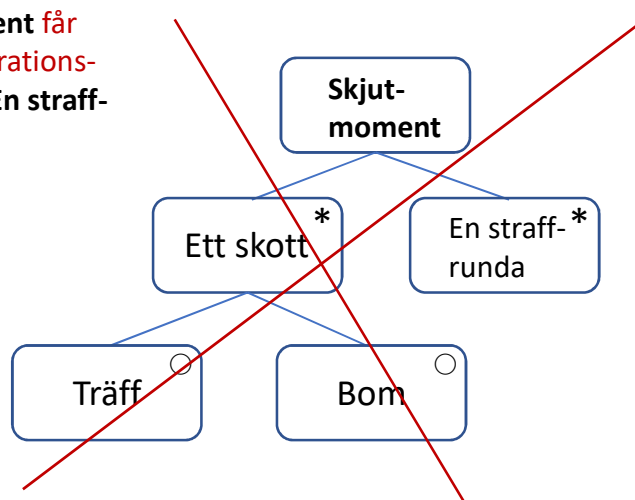
Iterationsdel  
Selektion

Selektionsdelar



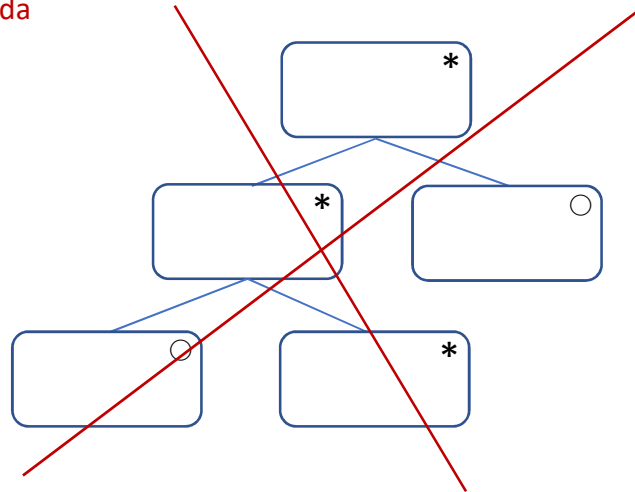
## Inte korrekt enligt JSP

Sekvensen **Skjutmoment** får  
t ex inte utgöras av iterations-  
delarna **Ett skott** och **En straff-  
runda**!!

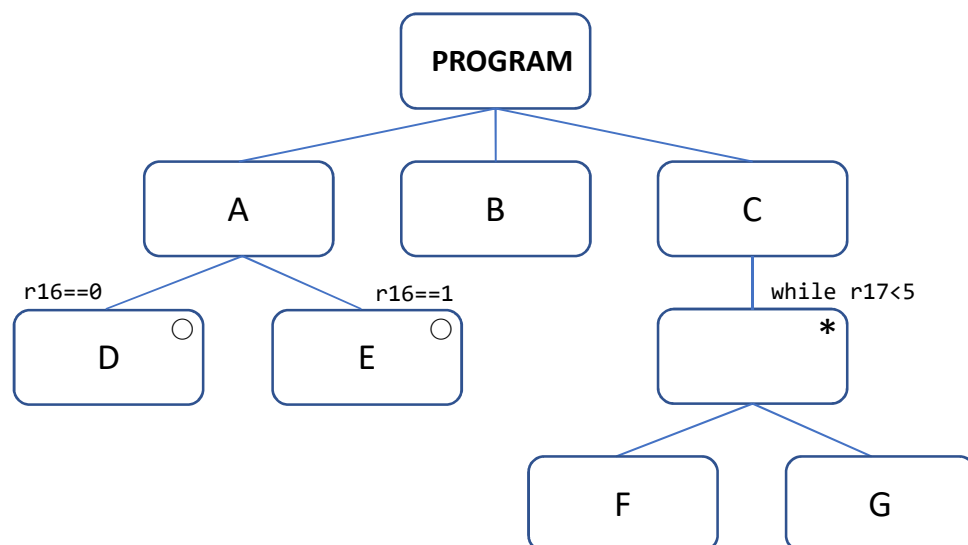


## Inte korrekt enligt JSP

Man kan ej heller blanda  
delar av olika typ!!



## Från JSP till assembler



**Sekvens**

```
PROGRAM:
    call A
    call B
    call C
END:
    jmp  END

B:
    ...
    ret
```

**Selektion**

```
A:
    cpi  r16,0
    brne A1
    call D
    jmp  A_DONE

A1:
    cpi  r16,1
    brne DEFAULT
    call E
    jmp  A_DONE

DEFAULT:
    ...
A_DONE:
    ret
```

**Iteration**

```
C:
    clr  r17
AGAIN:
    cpi  r17,5
    breq C_DONE
    call F
    call G
    inc  r17
    jmp  AGAIN
C_DONE:
    ret
```