

Tillämpad programmering ID1218



Introduktion
Johan Montelius

Kursmål



- programmerings-
 - modeller,
 - koncept,
 - språk och,
 - miljöer som behövs i modern mjukvarukonstruktion.

Ett redskap - en lösning



Utöka verktygslådan

objektorienterade

scripting

funktionella

imperativa

flertrådade

maskinnära

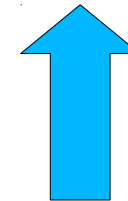
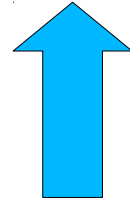
parallella



effektiv resurshantering

distribuerade system

C++

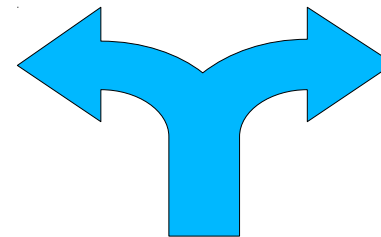


- imperativt
- objektorienterat
- maskinnära

Erlang

- funktionellt
- flertrådat

Java



- imperativt
- objektorienterat



omgivning

prestandamätning

pakethanterare

bibliotek

debugger

kompilatorer

typgranskare

editorer



<http://www.wikihow.com/Ride-a-Bicycle>



?



upplägg

- Föreläsningar
 - 14 st, måndagar och fredagar
- Övningstillfällen
 - Zandra, Karl , Mattias - IT-ing 2009
 - (tisdag,) onsdagar, 4 grupper
 - hjälp med programmering
- Seminarier
 - Leif Lindbäck
 - redovisning av uppgifter



planering

- v43 intro, Erlang
- v44 funktionell-prgm, uppg 1
- v45 flertrådad-prgm, uppg 2
- v46 intro C++
- v47 objekt, klasser, uppg 3
- v48 minneshantering, uppg 4
- v49 sammanfattning
- v50 tenta
- v2 rest



uppgifterna

- Tågklarering
 - tänk rekursivt, funktionellt i Erlang
- Ätande filosofer
 - tänk flertrådat, Erlang
- Ljudfiler
 - använd ett C-bibliotek i C++
- Sudoku
 - minneshantering i C++



inlämning

- En kort redogörelse för hur uppgiften löstes, vilka problem som fanns och hur man löste dem.
- Samarbeta med lösningen, skriv rapporten enskilt.
- Använd mall i LaTeX.

registrering



- Signera vid ert namn på listan.
- Om ni inte är med på listan så är det ett problem.
 - kursval
 - terminsregistrering

i stora drag



imperativ programmering



imperativ [im'p-] s. -en, -er,
uppmaningsform

imperativa språk

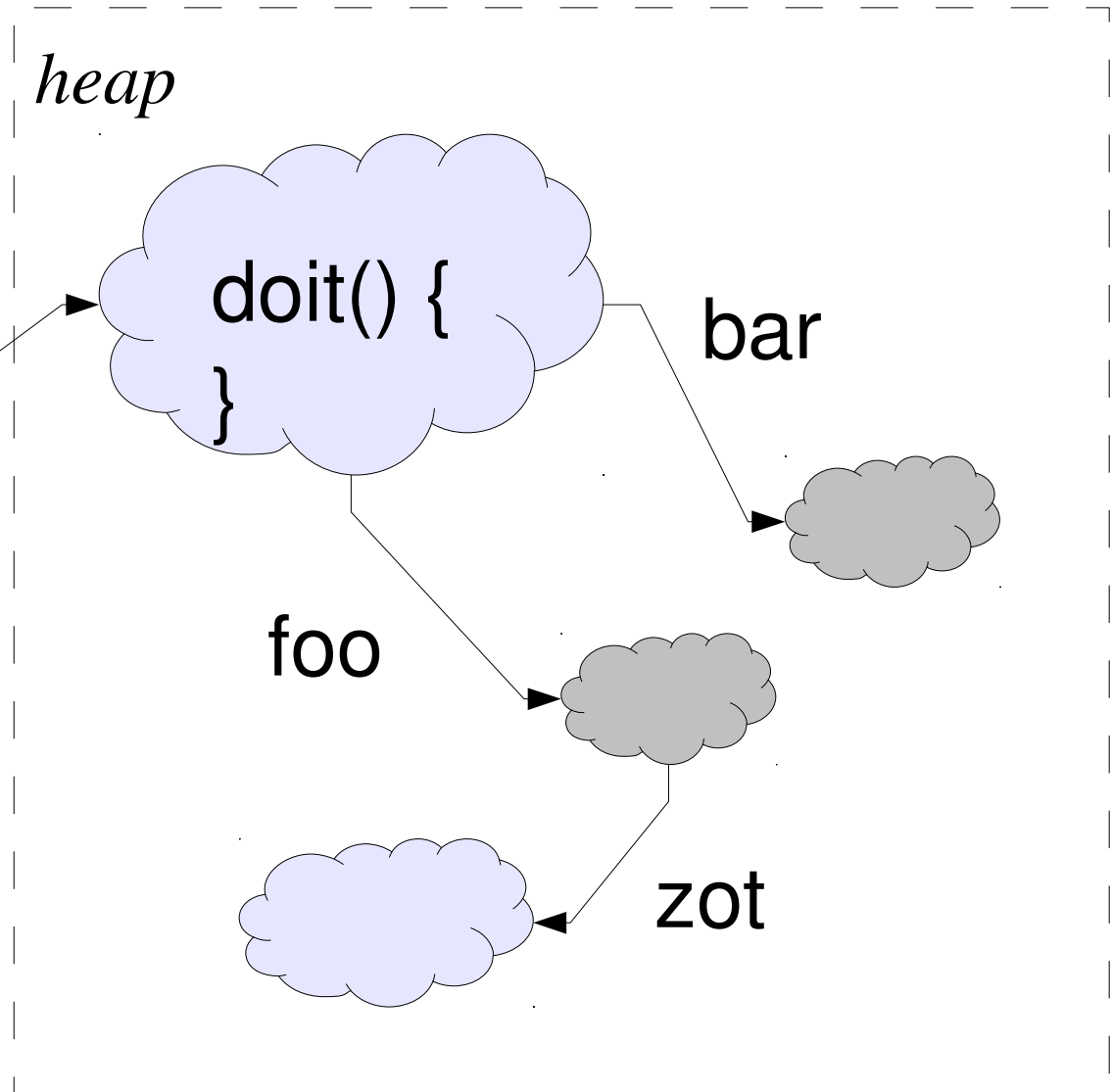


- Ett program är en sekvens av kommandon som förändrar ett tillstånd.
 - Hur ser tillståndet ut?
 - Vilka förändringar kan jag göra?

Java

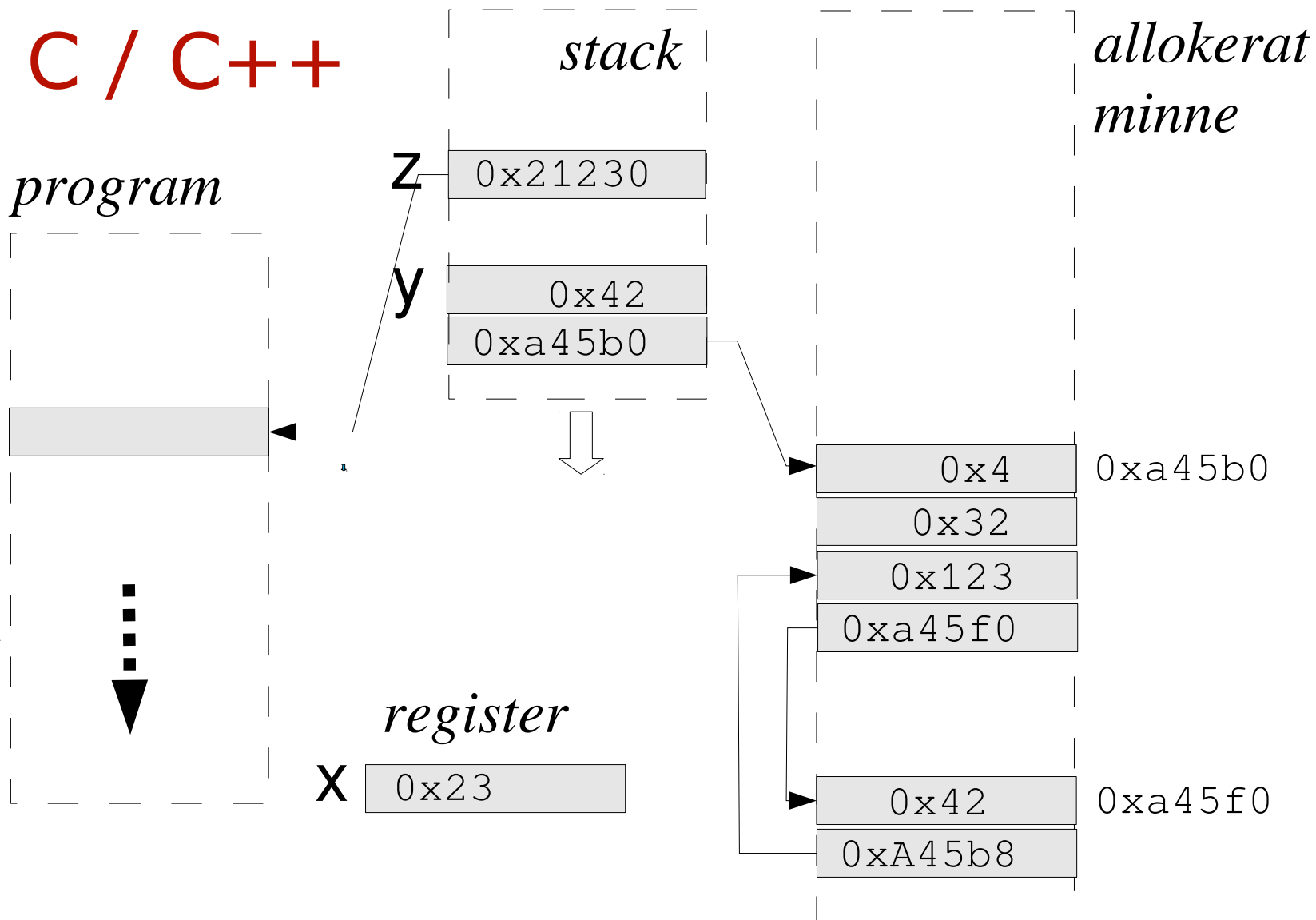


{x, y, z, ...}





C / C++



varför

- Mycket nära hur datorer idag är uppbyggda.
- Enkelt att få saker gjort.



funktioner



$$fib(n) = \left\{ \begin{array}{l} 1, n=0 \\ 1, n=1 \\ fib(n-1) + fib(n-2), n > 1 \end{array} \right\}$$

uttryck

$$fib(n) = \begin{cases} 0, n=0 \\ 1, n=1 \\ fib(n-1) + fib(n-2), n>1 \end{cases}$$

fib(4)

fib(3) + fib(2)

fib(2) + fib(1) + fib(1) + fib(0)

fib(1) + fib(0) + 1 + 1 + 0

1 + 0 + 2

3

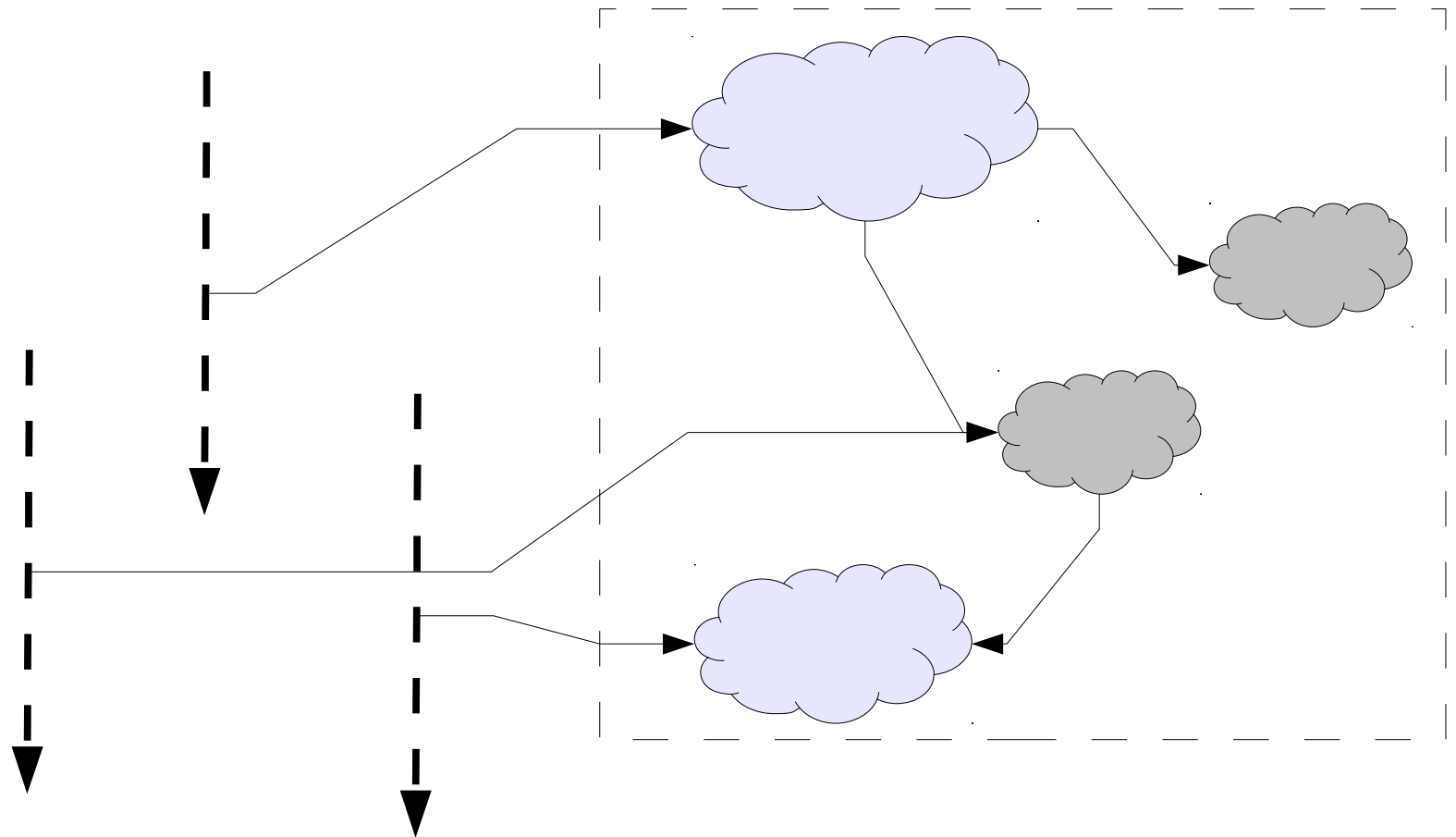


funktionell programmering

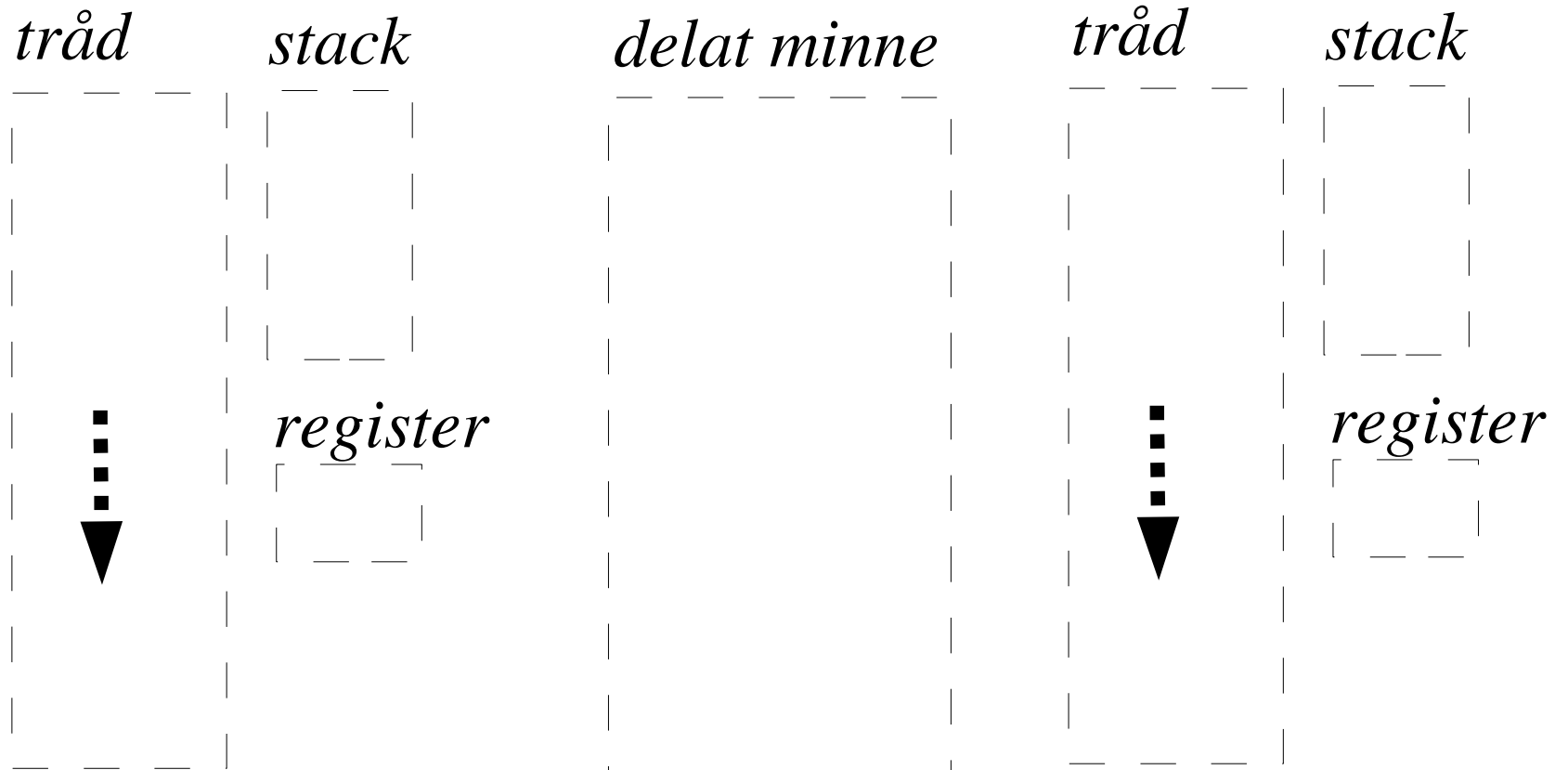


- Ett program kan tolkas som en beskrivning av vad som skall beräknas utan att tänka på hur detta beräknas.
- För att ett program skall vara effektivt måste funktionerna beskrivas med tanke på hur beräkningarna sker.

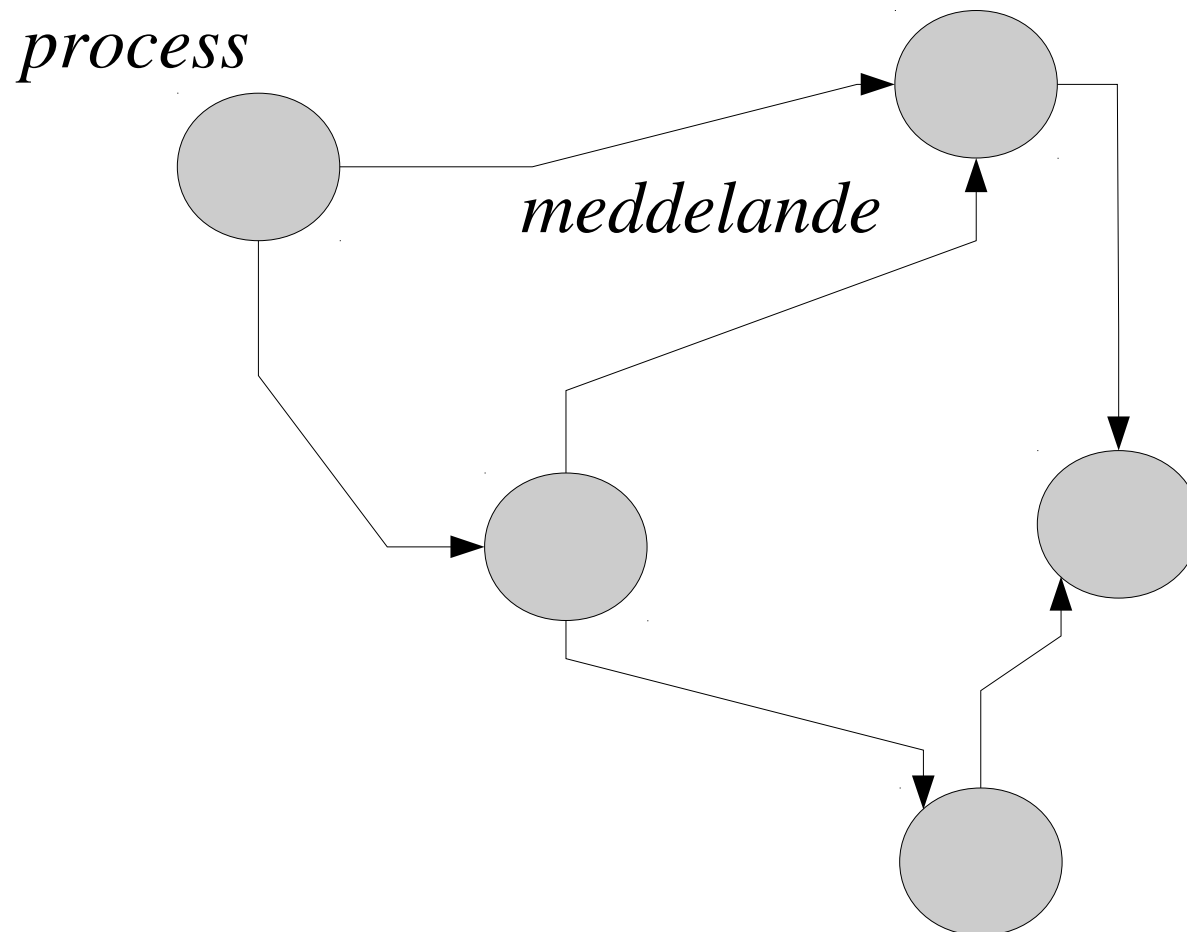
Java



C / C++



Actormodellen (Erlang)



Vertygslåda

