

Datavetenskap I

Föreläsning 3

Vad är ett informationssystem och
vad är objektorientering?

” The future's not set. There's no fate but what we make for ourselves. ”



Karaktären John Connor i Terminator 2: Judgment Day

Idag

- Varför analys och design behövs
- Informationssystem
 - Vad är informationssystem
 - Vilka svårigheter möter man gällande informationssystem
 - Hur hanteras dessa svårigheter
- Objektorientering
 - Objekt och klasser
 - Meddelanden och inkapsling
- Praktiskt exempel
- Läsanvisningar
- Inför workshop fredag

Kurslitteratur analys och design

- Kurslitteraturen behövs för din egen fördjupning och reflektion.
- Är på engelska – ska man arbeta inom detta ämnesområde så måste man kunna arbeta på och med engelska.
- Är lite tjockare sedan VT11. Två orsaker:
 - Lärare har sett att studenterna behöver börja öva på att bearbeta text tidigare. De teoretiska kurserna andra året upplevs då förhoppningsvis som mindre främmande.
 - Denna litteratur förklarar diagrammen tydligare och utförligare samt har mycket fler och mer komplexa exempel. Tidigare kurslitteratur hade för enkla exempel för att vara ett bra stöd i grupp- och projektarbeten.

Är ni redo?

Vi startar första föreläsningen...

Analys och design

VAD

analys

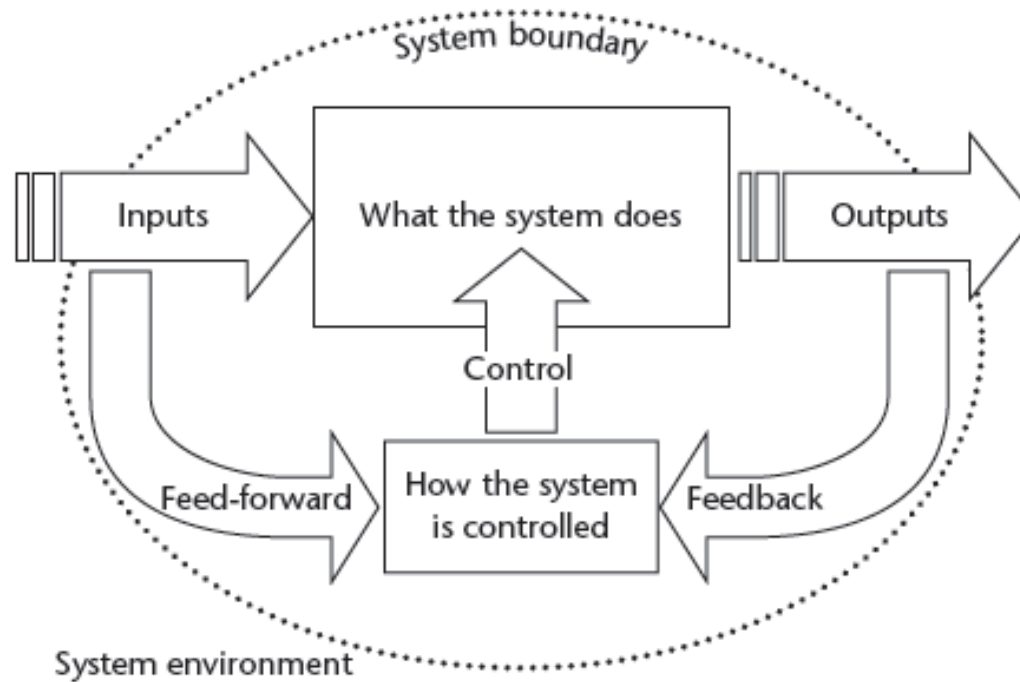
HUR

design

Informationssystem

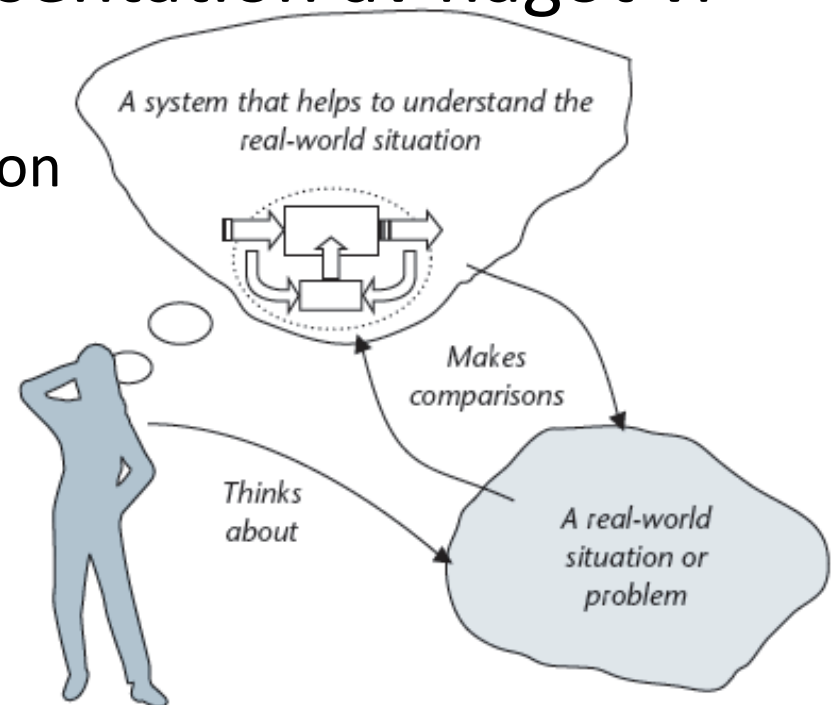
- IS skapas av människor
- Är inte nödvändigtvis digitala (även om moderna IS ofta är detta)
- Använd av en mängd olika organisationer
- Finns i vårt vardagliga liv

IS är *System*



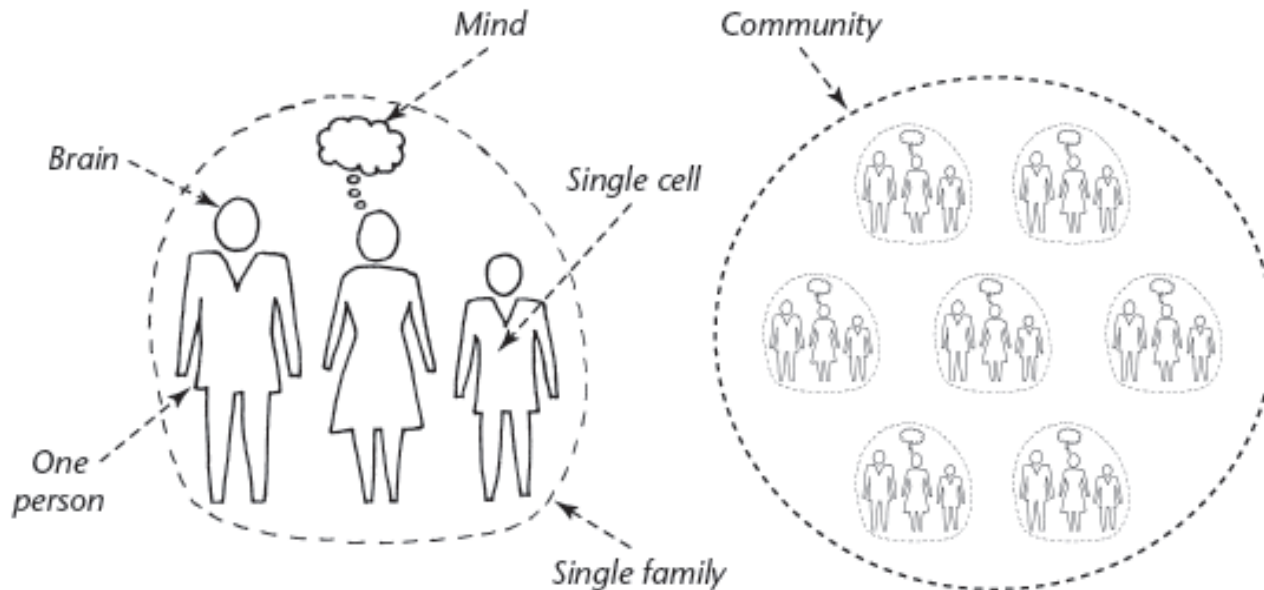
System och verkligheten

- Systemet är inte den verkliga världen utan vår representation av världen.
- Systemet kan ha fysiska delar, men systemet är fortfarande en representation av något vi behöver hjälp med.
 - Lagra och sortera information
 - Göra beräkningar
 - Underlätta kommunikation
 - ...



System och subsystem

- System samverkar med andra system.
System som arbetar tillsammans för att ge ett mervärde utöver de enskilda systemen kan ses som subsystem i ett större system.

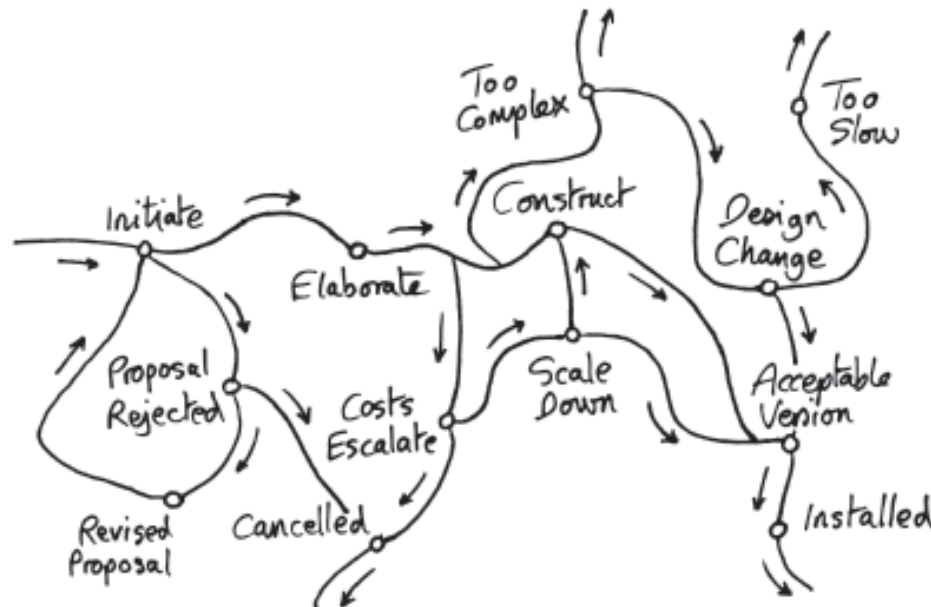


System och subsystem

- Ett subsystem kan verka fristående från helheten.
- Subsystem kan vara digitala, men inte nödvändigtvis.
- En organisation kan ses som ett system med flera subsystem.
- IS skall stödja en organisation att nå sitt mål.

Utveckling av system är en resa

- Resan har många möjliga vägar
- Utvecklare behöver vägvisare



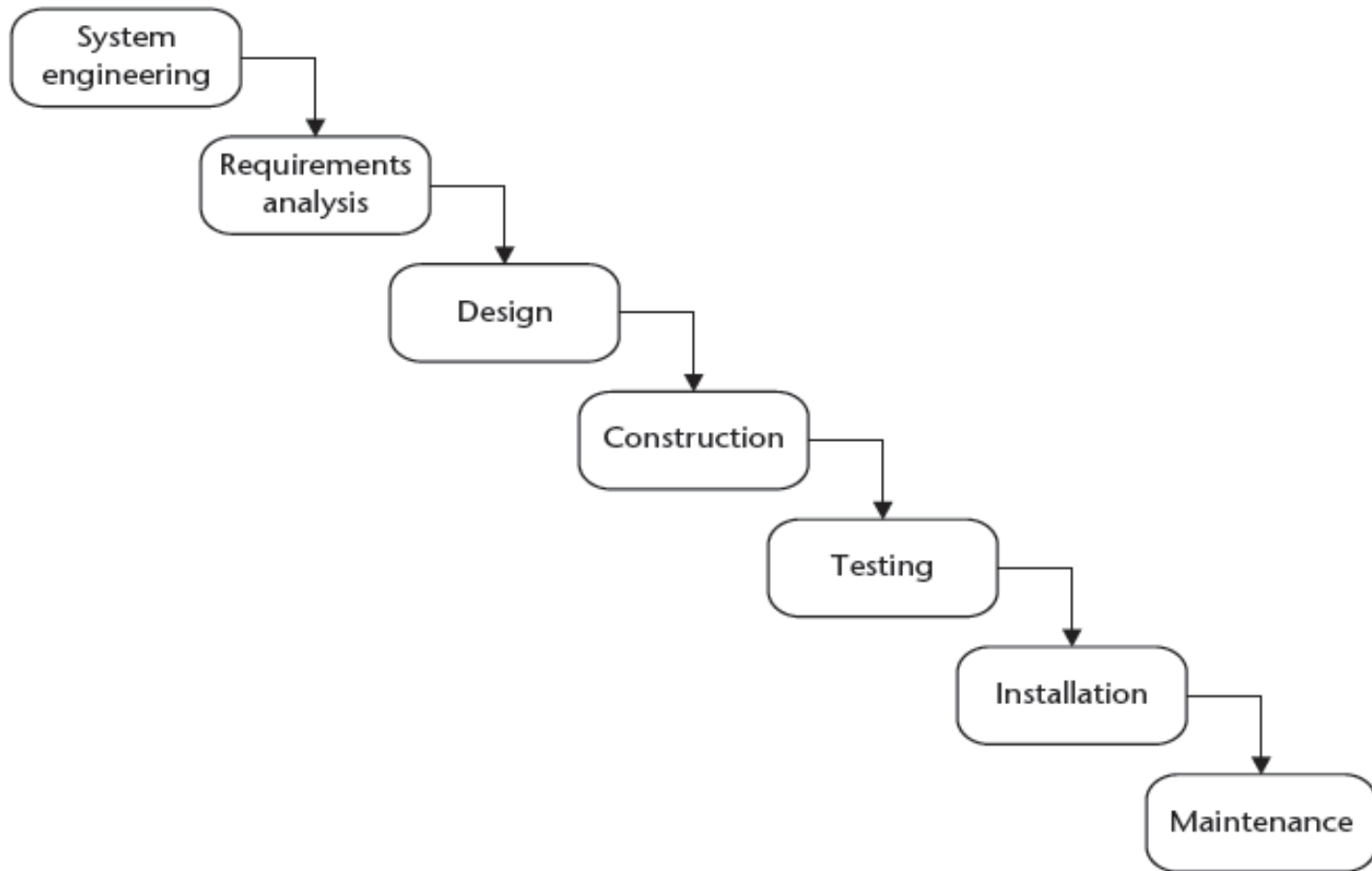
Vanliga problem

- Problem med kvalitet
 - Hanteras ofta genom att strukturera utvecklingsprocessen
- Problem med installation och drift
 - Ofta en konsekvens av kvalitetsproblem
- Problem med produktivitet
 - Åter igen hanteras detta genom att strukturera processen

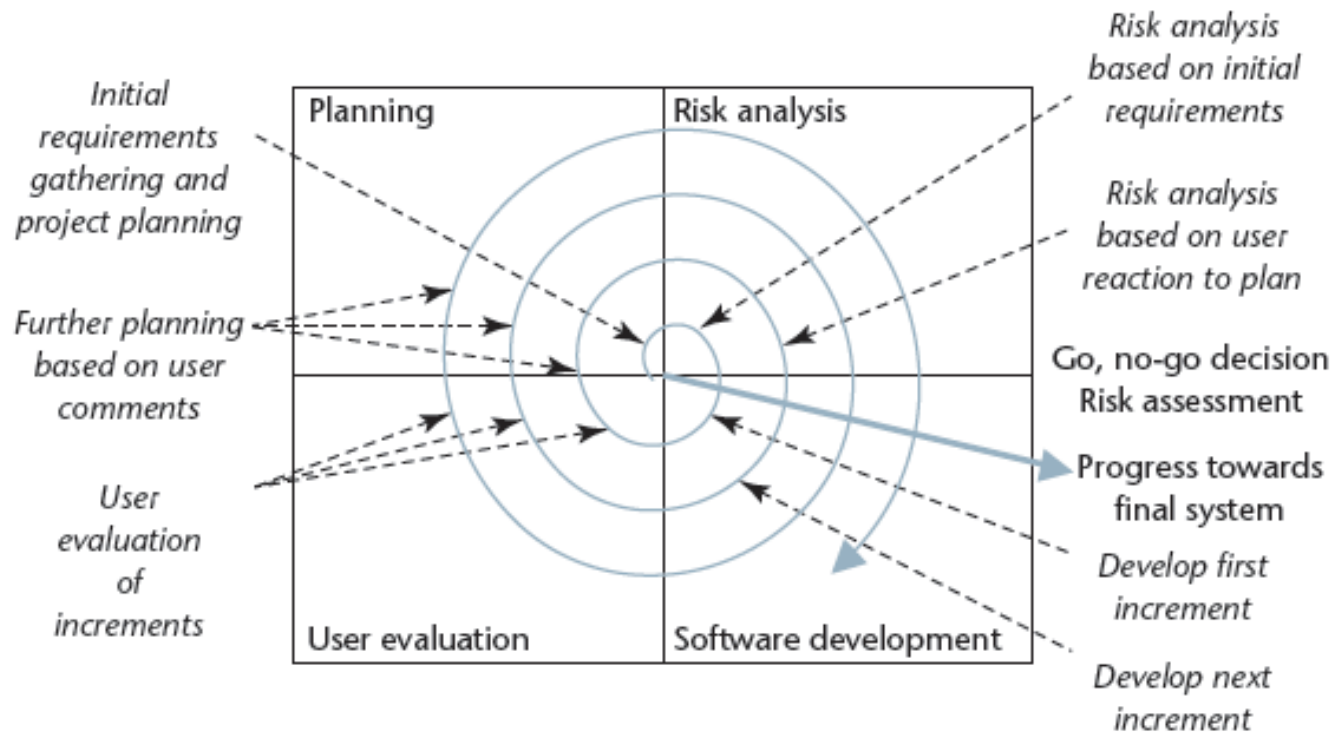
Problemlösning

- Problemlösning
 - Samla information om problemet
 - Analysera och omformulera problemet
 - Undersöka olika möjliga lösningar
 - Utvärdera lösningar/ bestämma sig för en lösning
 - Implementera lösningen

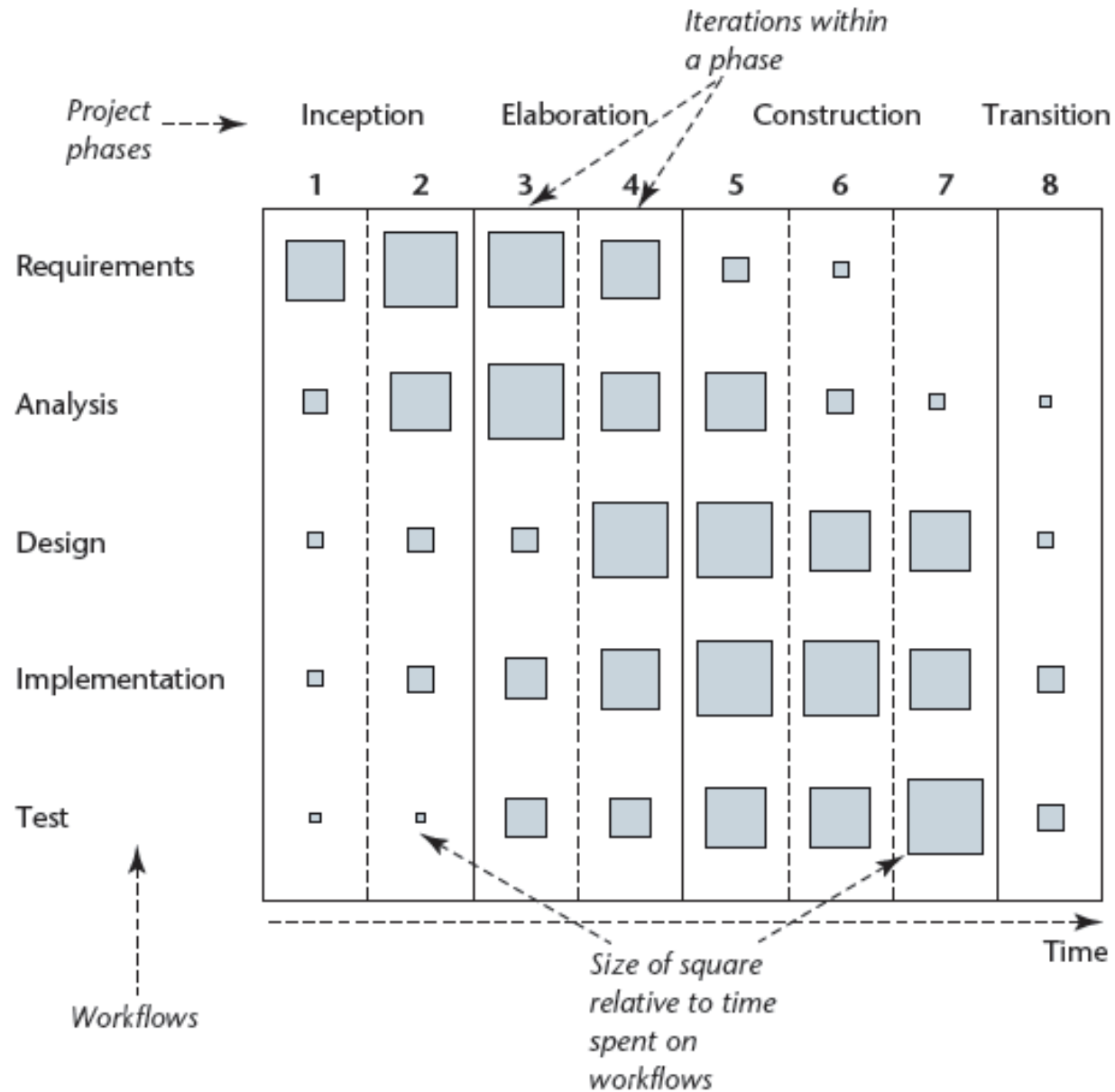
Waterfall Lifecycle Model



Spiral modell med inkrementell utveckling



RUP



Objekt

An object is:

“an abstraction of something in a problem domain, reflecting the capabilities of the system to

- keep information about it,
- interact with it,
- or both.”

Coad and Yourdon (1990)

Objekt

“Objects have state, behaviour and identity.”

Booch (1994)

- *Tillstånd (State)*: Det tillstånd som ett objekt befinner sig i vid ett givet ögonblick.
Tillståndet påverkar hur objektet kan bete sig.
- *Beteende (Behaviour)*: Vad ett objekt kan göra och hur det kan reagera på händelser.
- *Identitet (Identity)*: Varje objekt är unikt.

Exempel på objekt

Objekt	Identitet	Beteende	Tillstånd
En person	'Hussain Pervez'	Tala, gå, läsa.	Studerar, vilar, godkänd
En skjorta	Min favoritskjorta i blå bomul.	Krympa, fläckas, gå sönder	Struken, smutsig, sliten
En försäljning	säljnummer #0015, 18/05/05.	Erhålla bonuspoäng	Fakturerad, avbeställd
En flaska ketchup	<i>Den här</i> flaskan med ketchup.	Hälla	Osåld, oöppnad, slut

Objekt

- Egenskaper
 - Attribut – vilken data lagrar objektet
 - Operationer/metoder - vad kan objektet utföra , beteende
- Kan finnas hierarkier
 - Ger upphov till polymorfism
- Kommunikerar med meddelanden

Objekt och klass

- Objekt är instanser av någon klass
- En klass är en beskrivning av en mängd objekt med likartade:
 - Egenskaper
 - Semantik
 - Begränsningar

Meddelanden och inkapsling

‘Lager I en l k’ som
modell f r ett objekt:

Ett yttre lager av
definierade
operationer...

...ger  tkomst till ett
mellanlager av
operationer ...

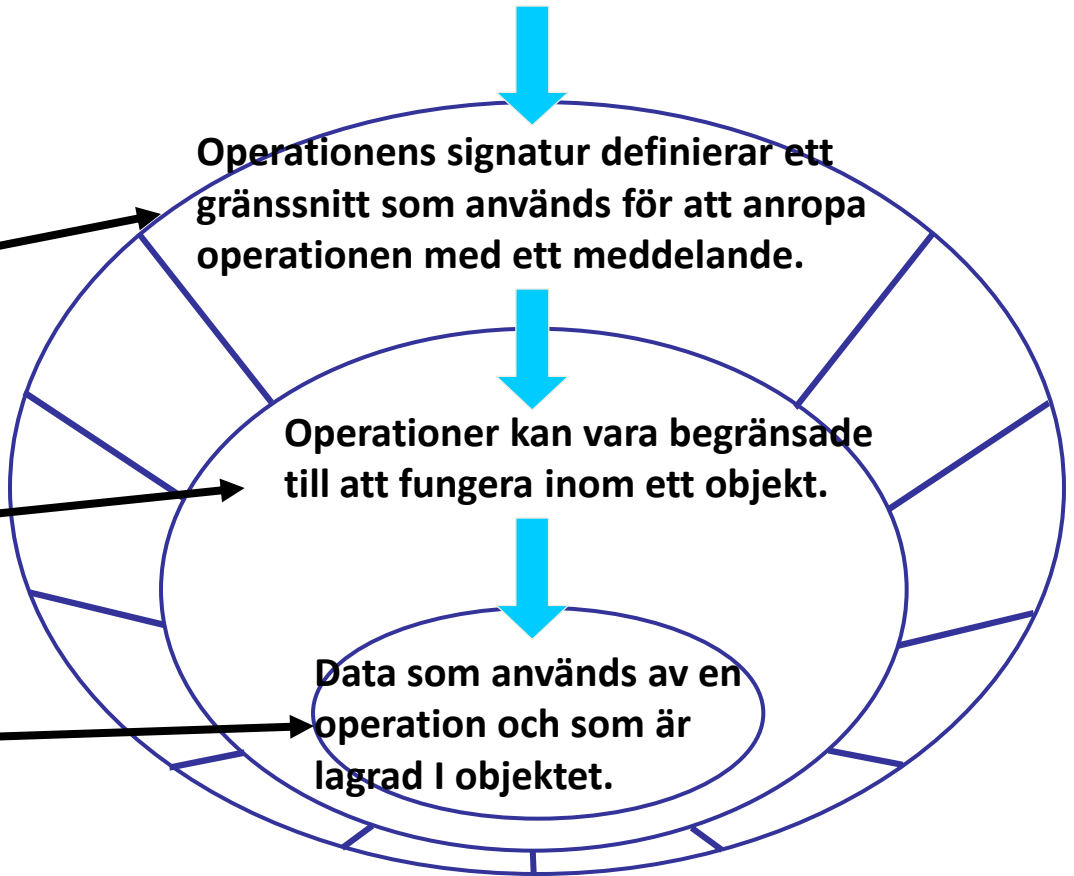
...som ger tillg ng till
k rnan av data

Meddelande fr n ett annat objekt
fr gar efter en t jnst.

Operationens signatur definierar ett
gr nssnitt som anv nds f r att anropa
operationen med ett meddelande.

Operationer kan vara begr nsade
till att fungera inom ett objekt.

Data som anv nds av en
operation och som  r
lagrad i objektet.



Dölja information: en princip

‘Lager i en lök’ som modell för ett objekt:

Det yttre lagret är synligt för andra objekt...

... och det är enda sättet att få tillgång till operationer ...

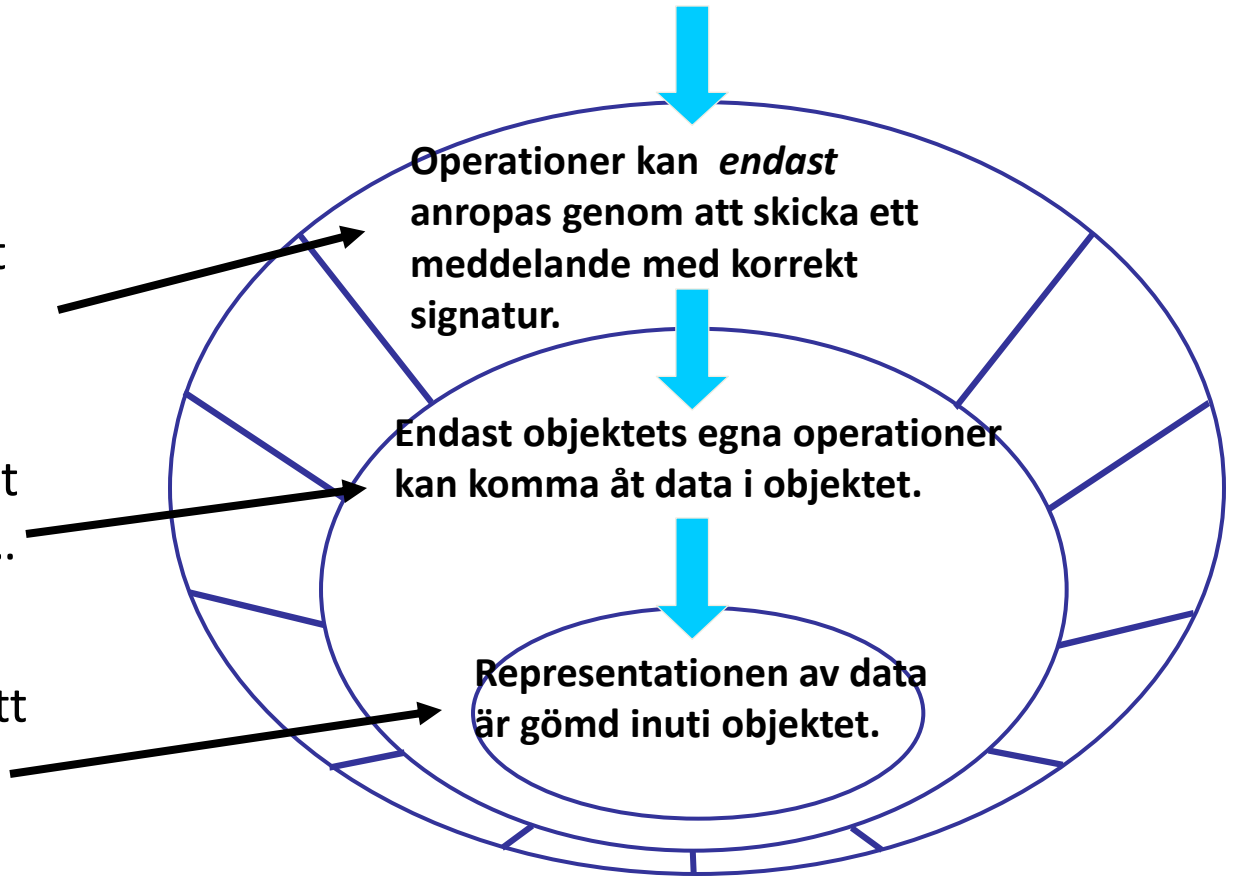
... som är enda sättet att få tillgång till dold data

Meddelande från ett annat objekt frågar efter en tjänst.

Operationer kan *endast* anropas genom att skicka ett meddelande med korrekt signatur.

Endast objektets egna operationer kan komma åt data i objektet.

Representationen av data är gömd inuti objektet.



Varför objektorientering?

- OO uppstod som ett svar på:
 - Ökad abstraktion, större system
 - Händelsehantering
 - GUI
 - Modulära system
 - Problem med utveckling
 - Återanvändning

Varför objektorientering?

- Kan spara arbete
 - Återanvändning av generella komponenter minskar arbete, kostnad och tid.
- Kan öka mjukvarukvalitén
 - Inkapsling ökar modulariteten
 - Subsystem blir mindre kopplade till varandra
 - Lättare att förstå övergångar mellan analys, design och kod

Varför objektorientering?

- När vill vi inte använda objektorientering?
 - Starkt databasorienterade system, i huvudsak lagring och processande av data
 - Mycket tunga algoritmer, simuleringar, beräkningar

Klassificera

- Vilka objekt har vi?
- Vilka likheter finns mellan dessa objekt?
- Vilka egenskaper är gemensamma?
 - Attribut
 - Operationer
- Finns det en hierarki?

Läsanvisningar

- Kapitel 1-3 OO läses igenom.
 - Dessa är en översikt för att ni ska få en bakgrund i problematiken som omger systemutveckling.
 - Ni behöver känna till det som står i dessa kapitel men det är inget ni behöver kunna återge i detalj.
 - Mycket av det som står i dessa kapitel återkommer i kurser inom Datavetenskap II-blocket med större djup.
 - Ni ska känna till definitionerna av system och informationssystem.
- Kapitel 4 OO
 - Kapitel 4.2.4 om generalisering och arv kan läsas översiktligt just nu. Vi återkommer till detta senare i kursen.
 - Resterande läses igenom grundligt.
 - Ni behöver kunna återge dessa begrepp och förstå dem för att förstå kommande delar.
- Kapitel 1.5 JF
 - Ni ska förstå koncepten.

Workshop

- Vi startar 8.15. Kom i tid! (Annars missar ni vad jag spelar upp vid inledningen).
- Läs igenom kapitel A1 och B1 i OO-boken som förberedelse.
 - Inscannade sidor finns som pdf-filer på It's L för de som inte har boken. Utskrivna kopior kommer att finnas på workshopen.
- Vi kommer att använda andra saker utöver dessa fallstudier på workshopen. Dessa behöver dock inte förberedas.
- Ta med något att anteckna på. Papper och penna, laptop, platta, ... spelar ingen roll så länge ni kan dokumentera vad ni kommer fram till