Introduktion till UML, OOAD & OOP, del 2

Data- och informationsvetenskap: Objektorienterad programmering och modellering för IA

Johan Holmberg 2019-09-09

MALMÖ UNIVERSITET

Dagens agenda

- Förra föreläsningen
- Tillståndsdiagram
- Sekvensdiagram

MALMÖ UNIVERSITET

Grundläggande OOP-koncept

Inom objekt-orienterad programmering stöter du på ett grundläggande koncept som du måste behärska. Dessa är:

- Objekt och klasser
- Meddelanden
- Inkapsling
- Komposition
- Arv
- Polymorfism

MALMÖ UNIVERSITET

Klassen och objektet

En **konstrukt** som beskriver mallen för ett **objekt**. Ett objekt är en **datastruktur** som har ett **tillstånd** och är knutet till **beteenden**. Objekt kommunicerar med **meddelanden**.

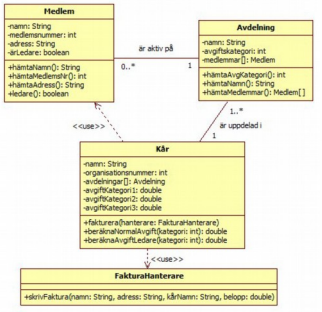
Vad är UML?

- UML (Unified Modeling Language) är ett modelleringsspråk som används för att beskriva objekt-orienterade mjukvarusystem.

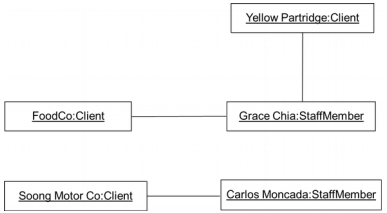
  Språket beskriver ett antal diagramtyper. Dessa diagramtyper kan användas för att beskriva ett mjukvarusystems:

  - Strukturer
  - Beteenden
  - Interaktioner

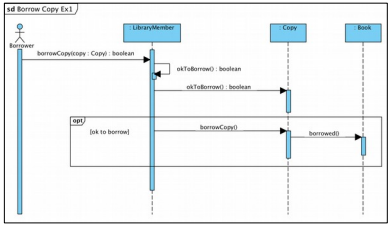
Diagramtyper



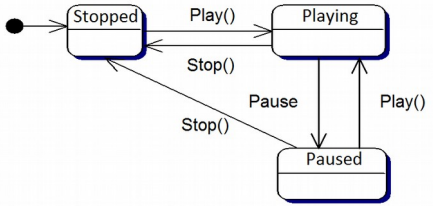
Klassdiagram



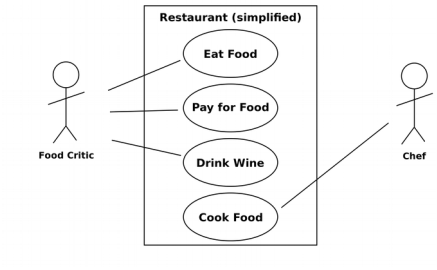
Objektdiagram



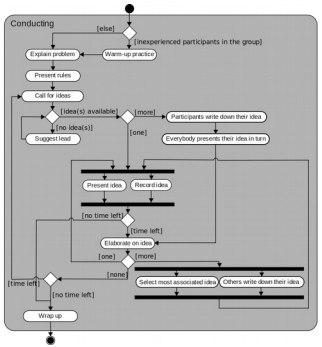
Sekvensdiagram



Tillståndsdiagram



Use case-diagram



Aktivitetsdiagram

MALMÖ UNIVERSITET

Klassdiagram

Ett klassdiagram beskriver strukturen hos ett mjukvarusystem, eller delar av ett sådant system.

Klassdiagrammet visar vilka **klasser** som ingår i systemet, deras **attribut** och **metoder**, och deras **relationer** till varandra.

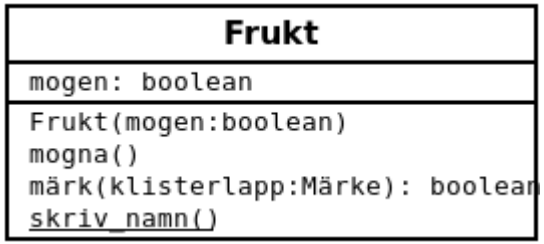
Klassdiagram: klassen

Klassen är den grundläggande komponenten i klassdiagrammet. Här ser vi en klass med ett **attribut** och två **metoder**. Returtyper och argument har märkts upp.

MALMÖ UNIVERSITET

Klassdiagram: statiska metoder

Statiska metoder kan anropas direkt från klassen utan att den instansierats till ett objekt.



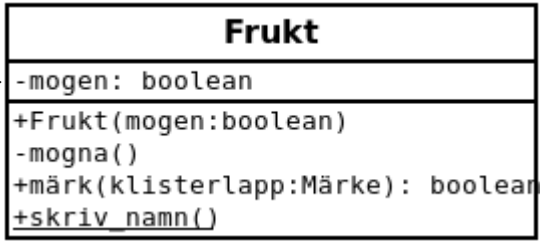
MALMÖ UNIVERSITET

Klassdiagram: synlighet

I språk som har språkliga konstruktioner för att hantera inkapsling, kan attribut och metoder vara “osynliga” för andra objekt. Osynliga attribut och metoder sägs vara **privata** och synliga **publika**.

Minustecken betecknar **privat** åtkomst →

Plustecken betecknar **publik** åtkomst →



MALMÖ UNIVERSITET

Klassdiagram: konstanter

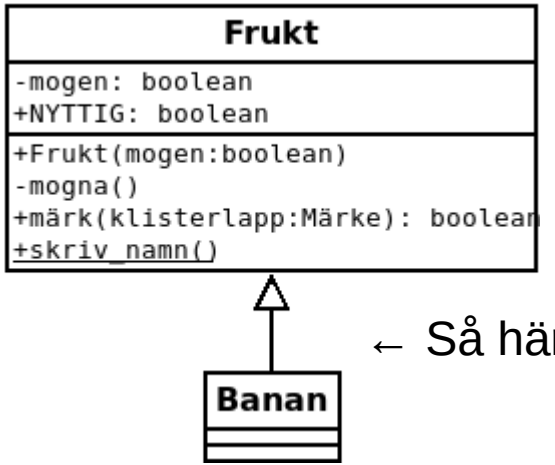
Klasskonstanter är attribut hos en klass som inte förändras. De är gemensamma för alla objekt av samma klass.

← Så här modelleras en konstant i UML

MALMÖ UNIVERSITET

Klassdiagram: arv och generalisering

Arvet gör att en klass ärver egenskaper och attribut från en annan klass. Fenomenet kallas **generalisering**.

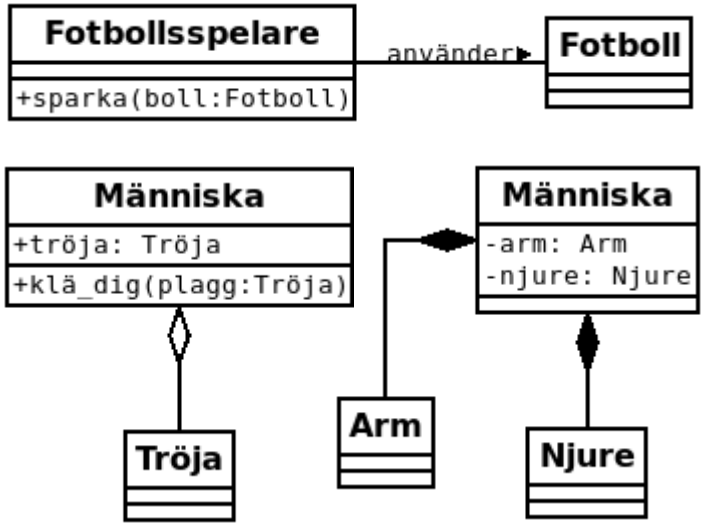


← Så här modelleras ett arv i UML

MALMÖ UNIVERSITET

Klassdiagram: associationer

Tre viktiga **relationer** är **associationen**, **aggregationen** och **kompositionen**.

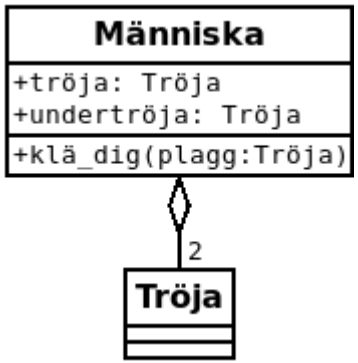


MALMÖ UNIVERSITET

Klassdiagram: multiplicitet

Multipliciteten anger hur många **instanser** av ett objekt en given instans av något objekt känner till:

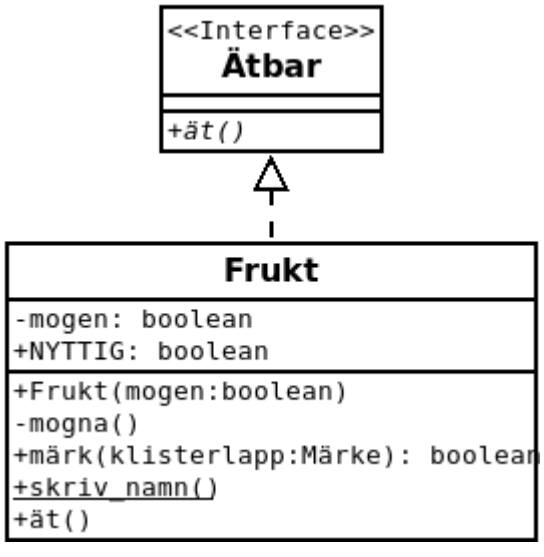
- 0, 1, n (dvs ett exakt antal)
- 0..*, * (dvs ”0 eller fler”)
- ~~+ (dvs en eller fler)~~
- m..n (dvs ett intervall)

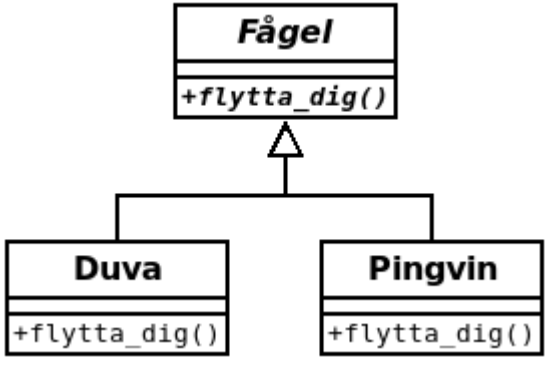


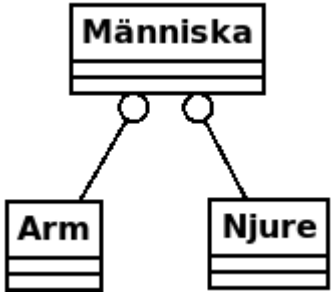
MALMÖ UNIVERSITET

Klassdiagram: speciella klasstyper

Vi diskuterade **abstrakta** och **inre klasser**, samt **gränssnitt**.



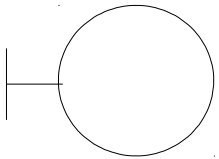


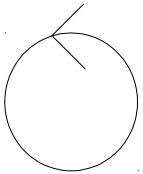


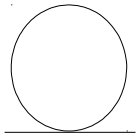
MALMÖ UNIVERSITET

Klassdiagram: stereotyper

Beskriver **roller** som klasser kan ha. Tre vanliga, som dock inte ingår i UML-standarden är **boundary**, **control** och **entity**.



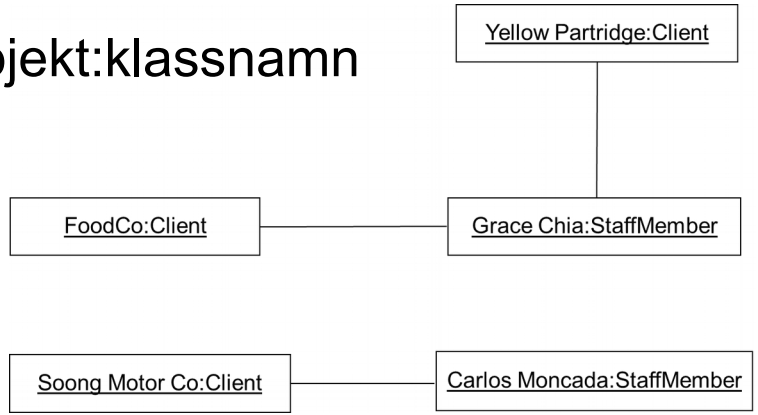




MALMÖ UNIVERSITET

Objektdiagram

- Objektdiagram visar systemets objekt vid en given tidpunkt under exekvering.
- Associationen mellan objekt kalls länk.
- Namn på objekt:klassnamn



MALMÖ UNIVERSITET

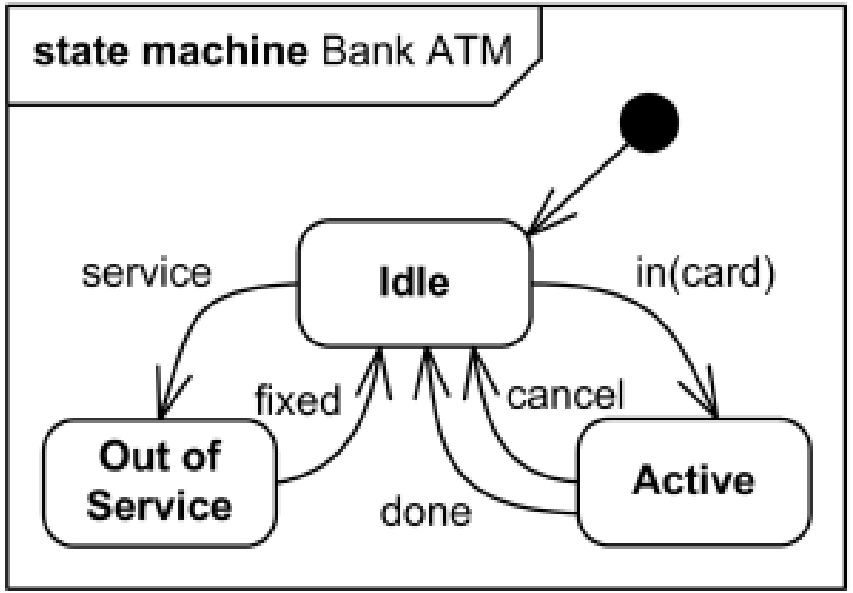
Tillståndsdiagram

MALMÖ UNIVERSITET

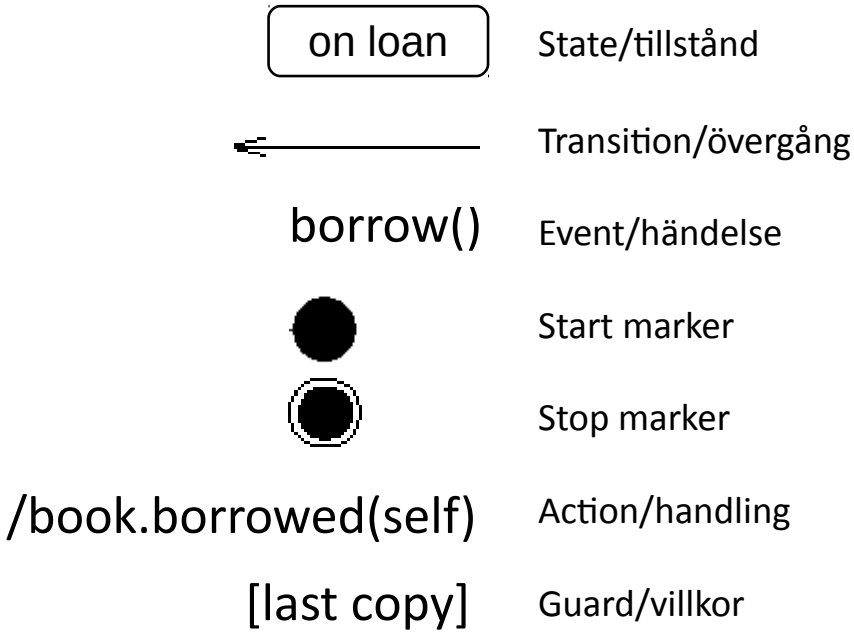
Tillståndsdiagram

Tillståndsdiagram beskriver beteendet hos en avgränsad del av ett system.

Diagrammet visar hur tillståndet hos (del)systemets ingående objekt kan förändras.

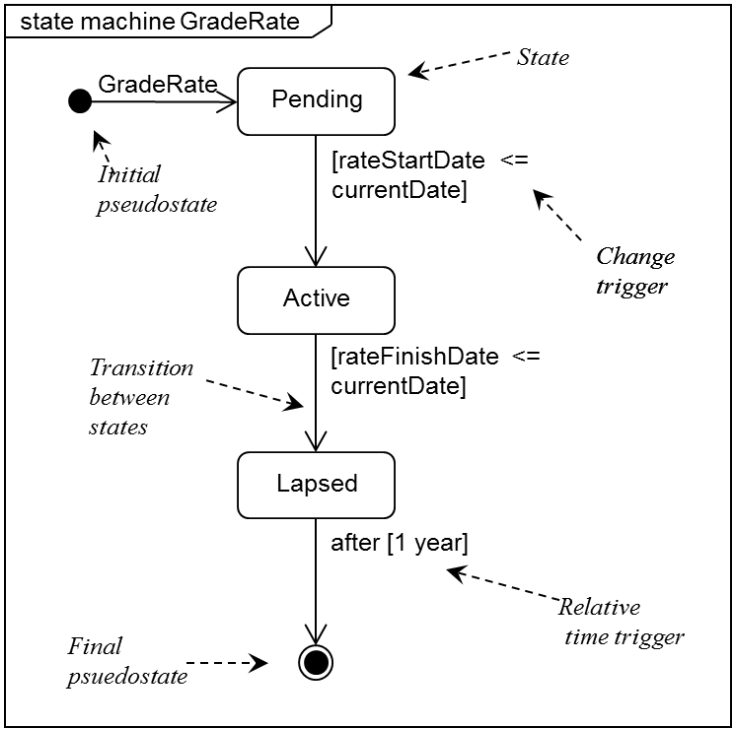


Tillståndsdiagram



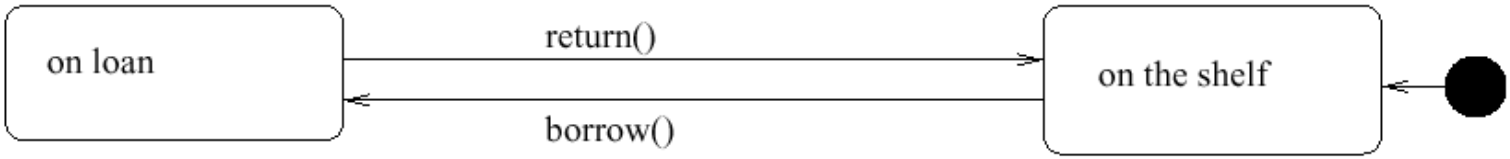
MALMÖ UNIVERSITET

Tillståndsdiagram, exempel



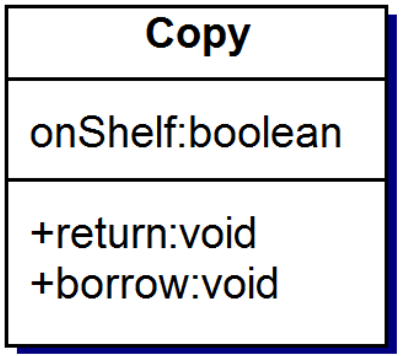
MALMÖ UNIVERSITET

Modellering av ett lån



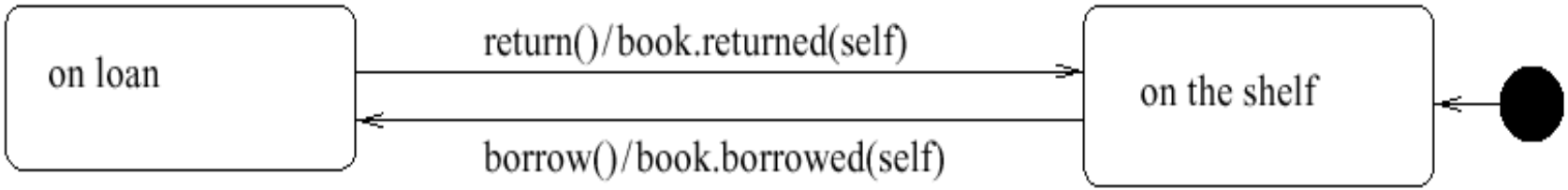
`onShelf = False` `onShelf = True`

Tillståndsdiagram för klassen Copy, med händelser



MALMÖ UNIVERSITET

Modellering av ett lån



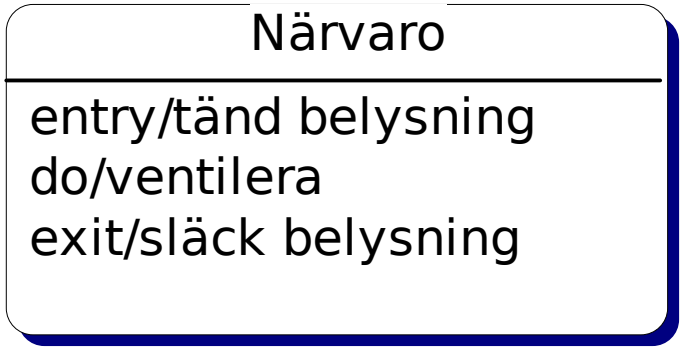
Copy-objektet skickar **meddelandet** `borrowed(this)` och `returned(this)` till dess associerade Book-objekt, som en reaktion på mottagandet av meddelandet `borrow()` respektive `return()`.

En serie av handlingar separeras med flera /.

Tillståndsdiagram för klassen Copy, med händelser och handlingar

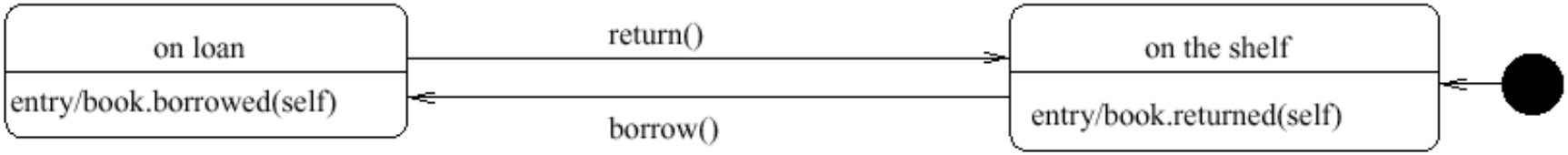
Actions i ett tillstånd

- **entry/** är en aktivitet som inträffar direkt när tillståndet uppträder
- **do/** är beskrivningen på en aktivitet som skall pågå hela tiden objektet befinner sig i tillståndet.
- **exit/** är en aktivitet som inträffar när objektet lämnar tillståndet.

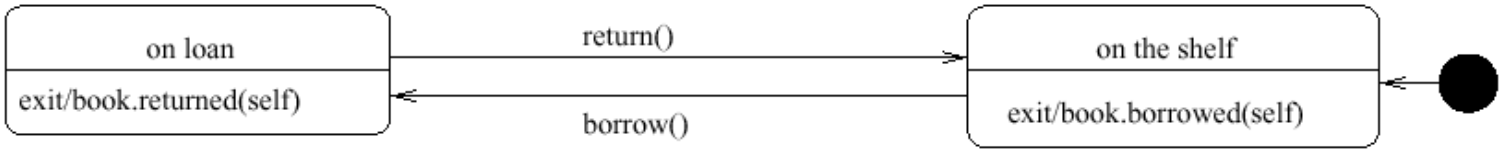


MALMÖ UNIVERSITET

Modellering av ett lån

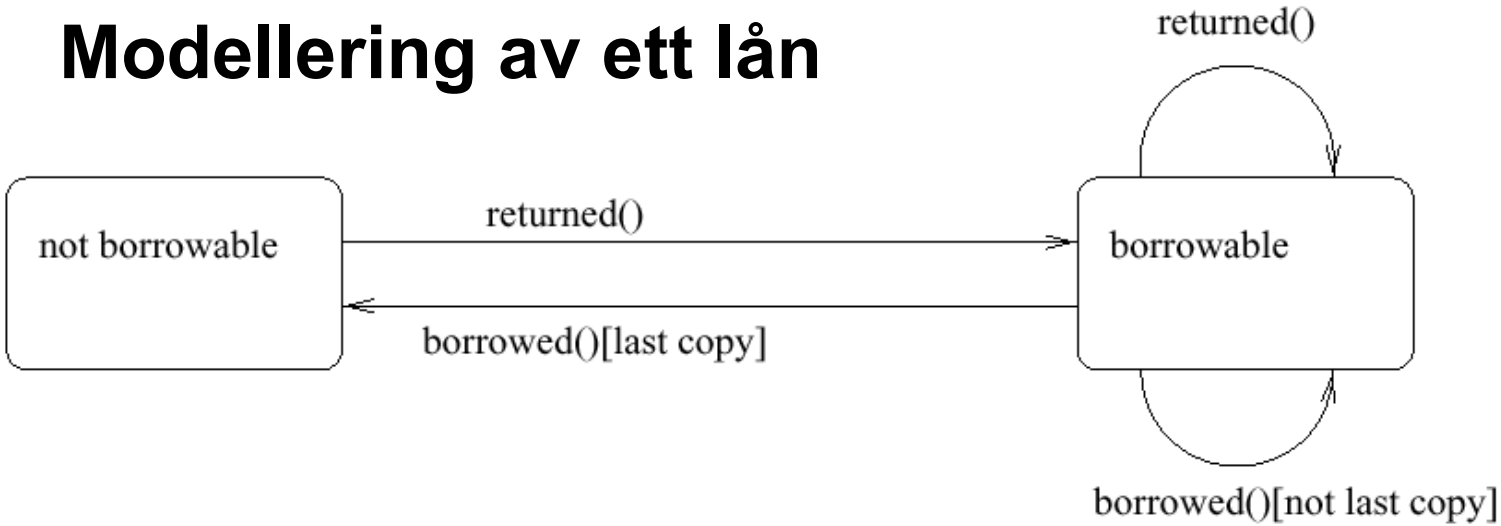


Tillståndsdiagram för Copy, med händelser och ingångshandlingar



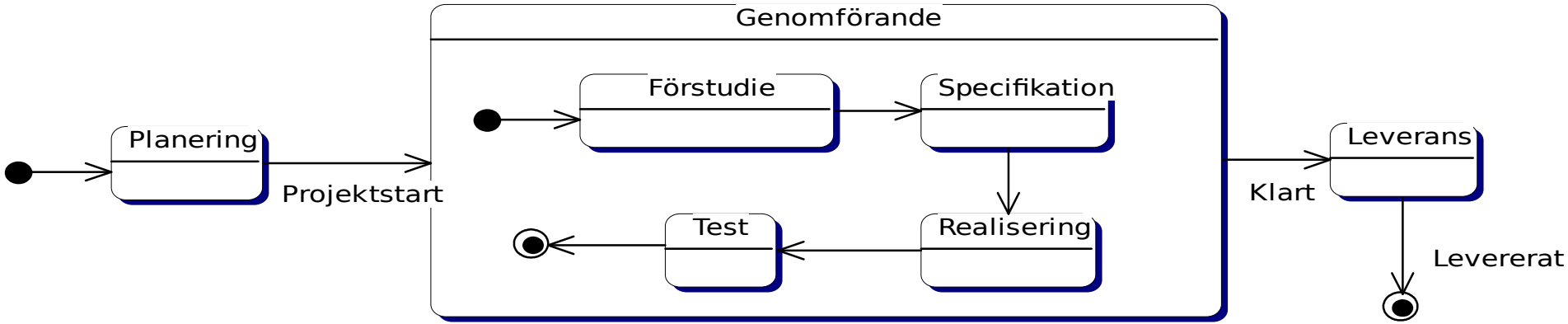
Tillståndsdiagram för Copy, med händelser och utgångshandlingar

Modellering av ett lån



Tillståndsdiagram för Copy, med händelser och villkor

Nästlade tillstånd



MALMÖ UNIVERSITET

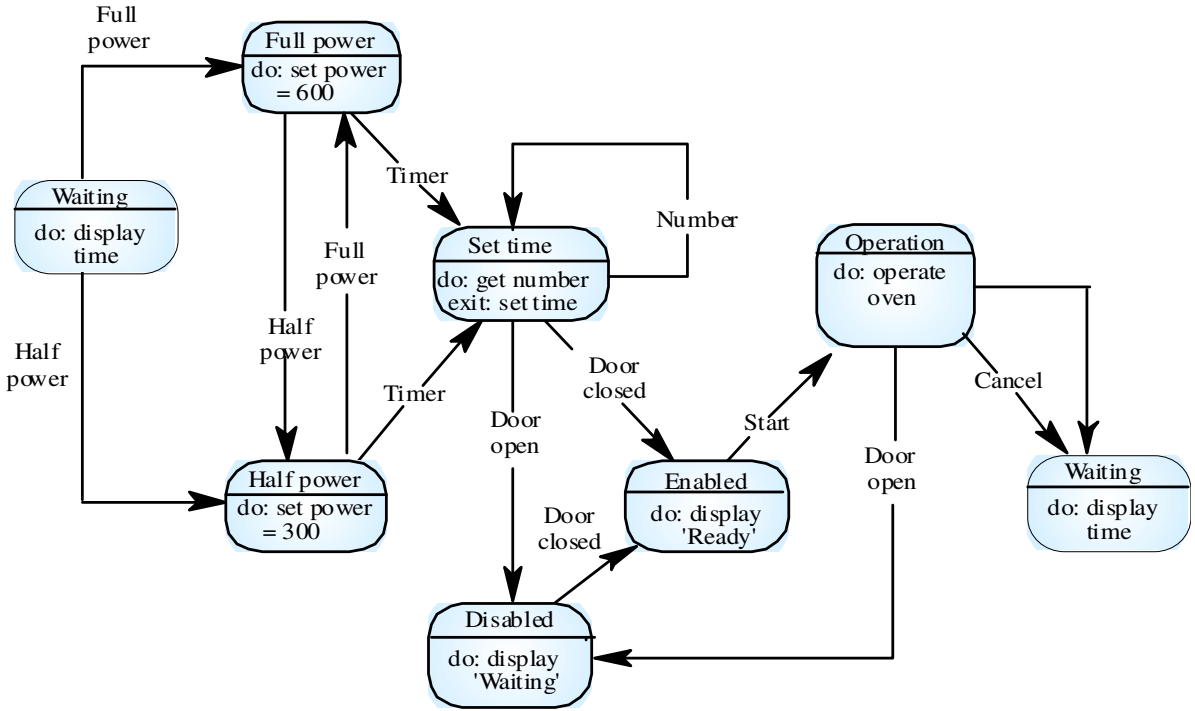
Exempel: mikrovågsugnen

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for 5 seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

Exempel: mikrovågsugnen

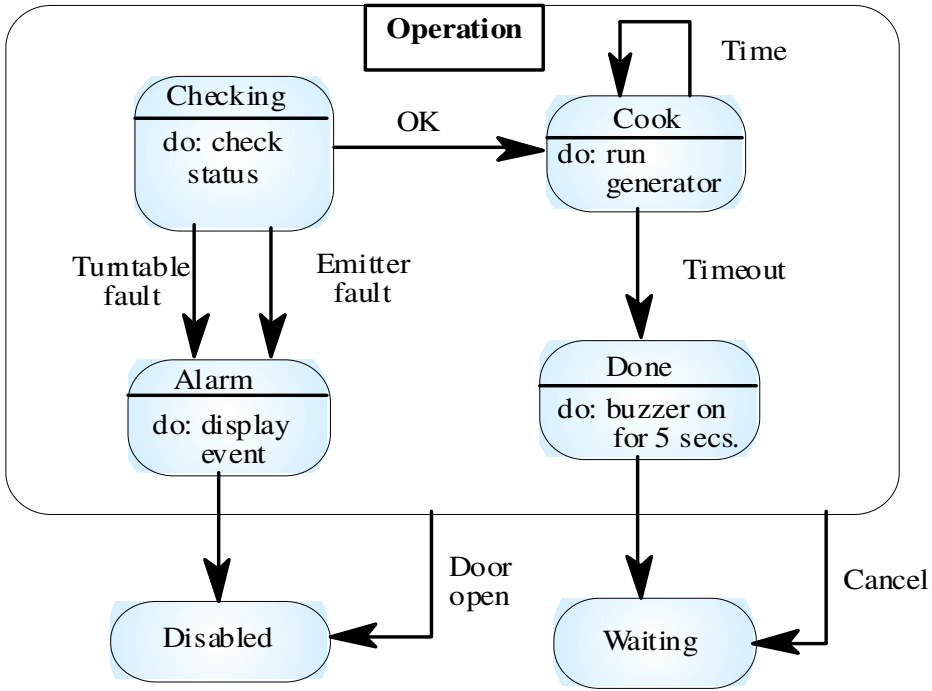
Stimulus	Description
Half power	The user has pressed the half power button
Full power	The user has pressed the full power button
Timer	The user has pressed one of the timer buttons
Number	The user has pressed a numeric key
Door open	The oven door switch is not closed
Door closed	The oven door switch is closed
Start	The user has pressed the start button
Cancel	The user has pressed the cancel button

Exempel: mikrovågsugnen



MALMÖ UNIVERSITET

Exempel: mikrovågsugnen

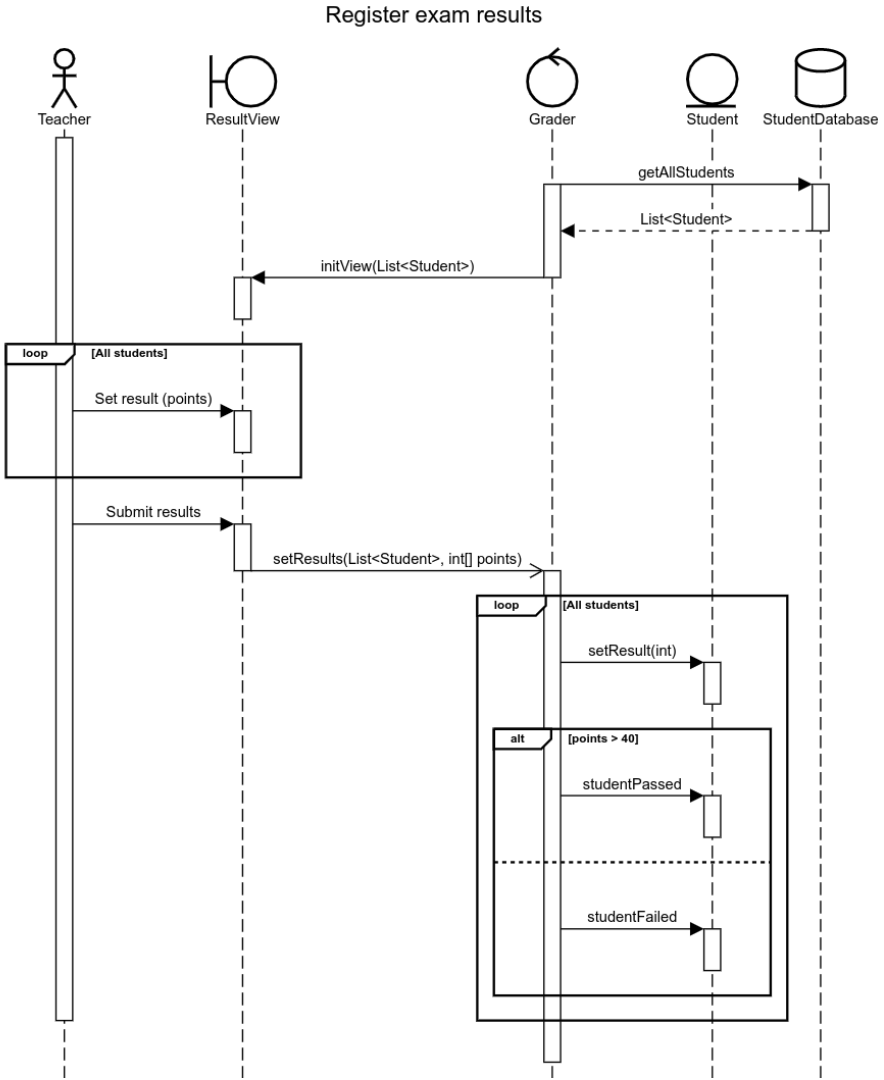


Här är det möjligt att kapsla in ett tillståndsdiagram i ett enda tillstånd: *Operation*. Detta gör att vi kan förenkla detaljnivån.

MALMÖ UNIVERSITET

Sekvensdiagram

MALMÖ UNIVERSITET



Sekvensdiagram

En typ av interaktionsdiagram som visar hur objekt i ett system interagerar med varandra.

Varje diagram visar hur ett enda användningsfall ser ut i systemet

Diagrammet visar vilka objekt som ingår i interaktionen, vilka meddelanden som utbyts och i vilken ordning dessa meddelanden utbyts.

MALMÖ UNIVERSITET

Snälla, snälla student!

Du kommer vid något tillfälle lockas att använda **Lucidchart** för att rita dina UML-diagram eller lära dig mer om UML. Undvik dem, de har inte alltid riktig koll på hur UML verkligen ska användas och kommer i slutänden bara att förvirra dig.

MALMÖ UNIVERSITET

Vårt arbetsexempel

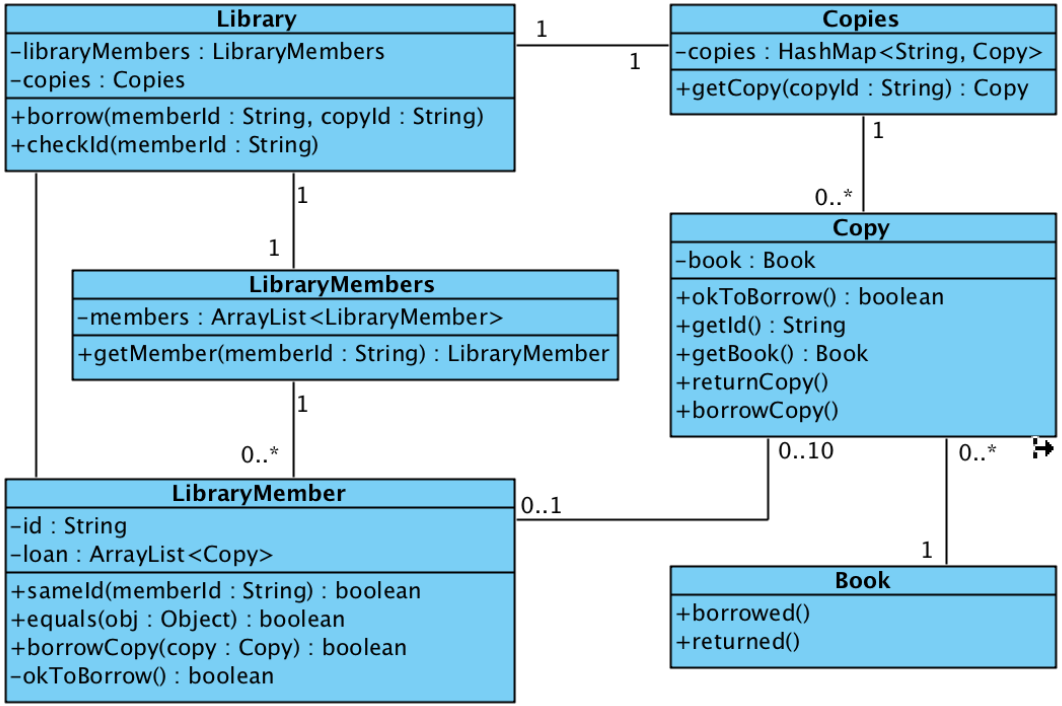
Låna en bok från biblioteket

MALMÖ UNIVERSITET

Vikten av konsistenta diagram

- Klassdiagram och interaktionsdiagram måste vara **konsistenta**. Det får inte finnas några motsägelser eller konflikter mellan diagram (eller koden).
- Det får inte finnas instanser av klasser i interaktionsdiagram som inte har någon motsvarighet i ett klassdiagram i modellen.
- Man kan endast skicka meddelanden till objekt man känner till.

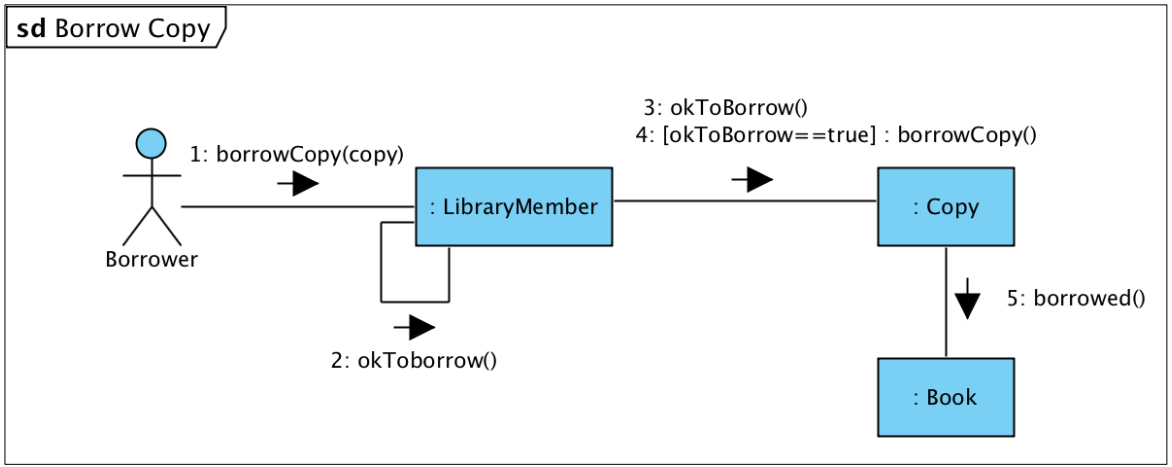
Klassdiagram: biblioteket



MALMÖ UNIVERSITET

Först: ett kommunikationsdiagram

En äldre version av ett interaktionsdiagram. Det visar alla ingående objekt och interaktioner. Interaktionerna är numrerade efter den ordning de sker i.

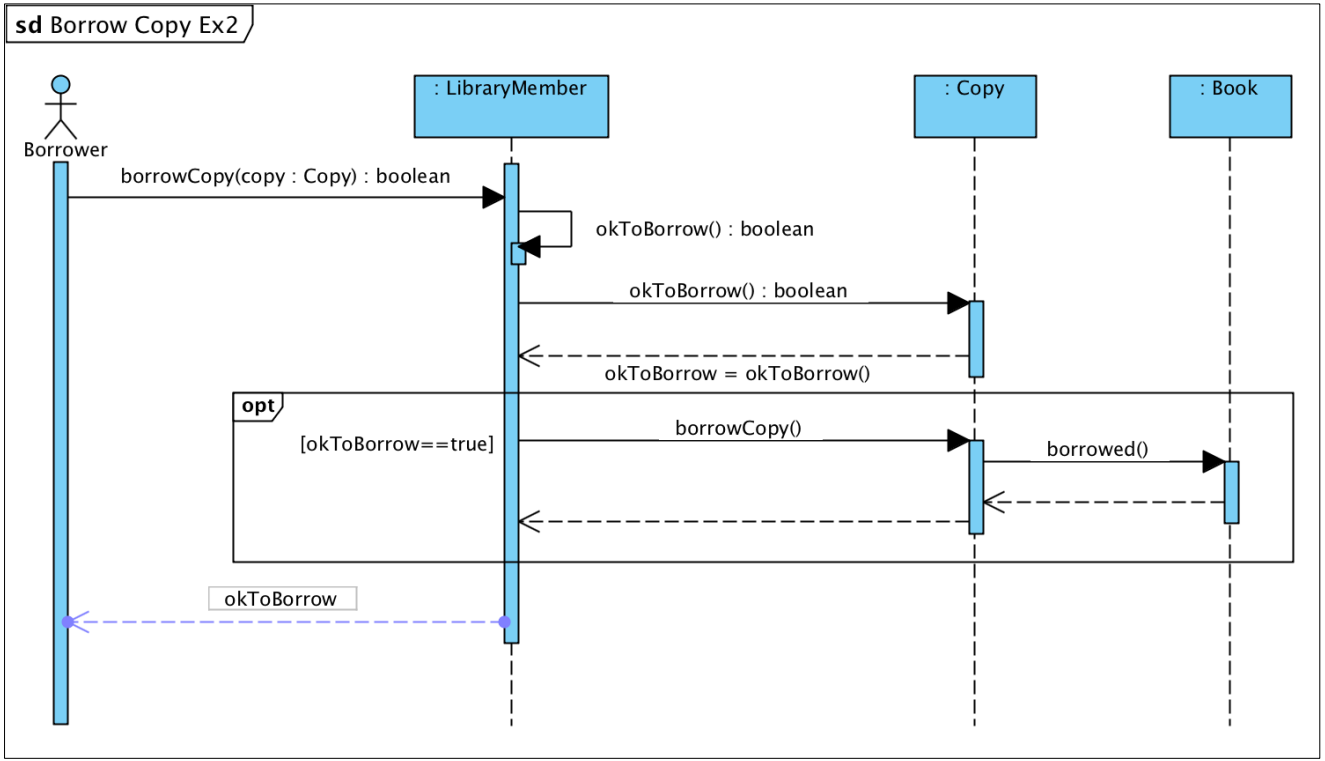


MALMÖ UNIVERSITET

Sekvensdiagrammet

- Sekvensdiagram visar **objekt,** deras **livstid** och deras **aktivitet** under en viss **uppgift**.
- Meddelanden numreras inte utan ordningen mellan dessa visas genom var på livstidsaxeln de sker. Tiden startar högst upp och rör sig nedåt i diagrammet.

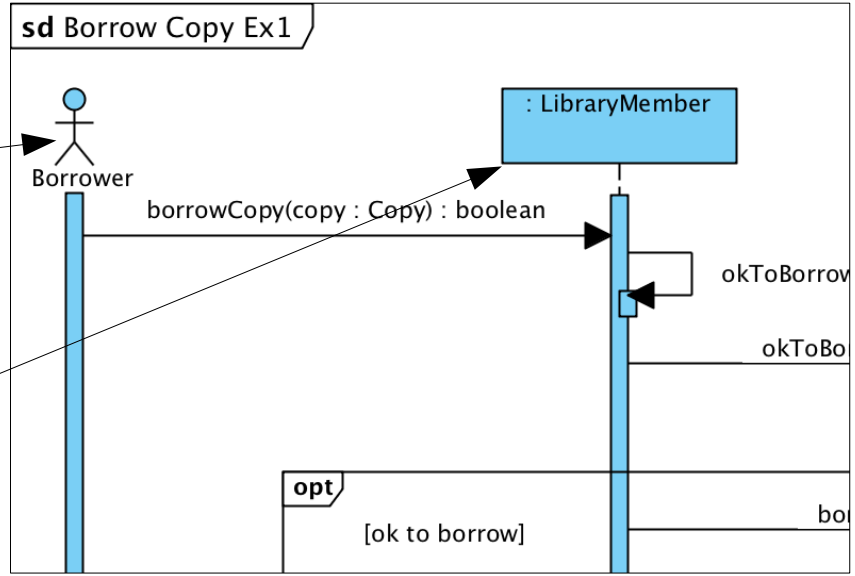
Användningsfall: låna en bok



MALMÖ UNIVERSITET

Sekvensdiagrammets beståndsdelar

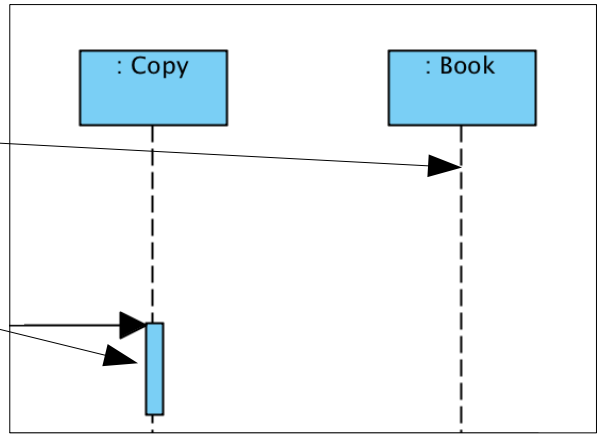
- Figuren längst upp till vänster är en **aktör**. Detta är vanligtvis en användare.
- Boxarna representerar de **objekt** som ingår i systemet.



MALMÖ UNIVERSITET

Sekvensdiagrammets beståndsdelar

- Linjen från rutan som representerar objektet visar objektets **livstid**. Detta är en **livlina**.
- När ett objekt är **aktivt** så visas en rektangel över linjen. Detta är en **aktivitetsbox**.
- Aktivt objekt – kod i klassen som objektet tillhör körs.
- När ett objekt skickar ett meddelande till sig själv så visas detta som en extra överlappande rektangel.



MALMÖ UNIVERSITET

Sekvensdiagrammets beståndsdelar

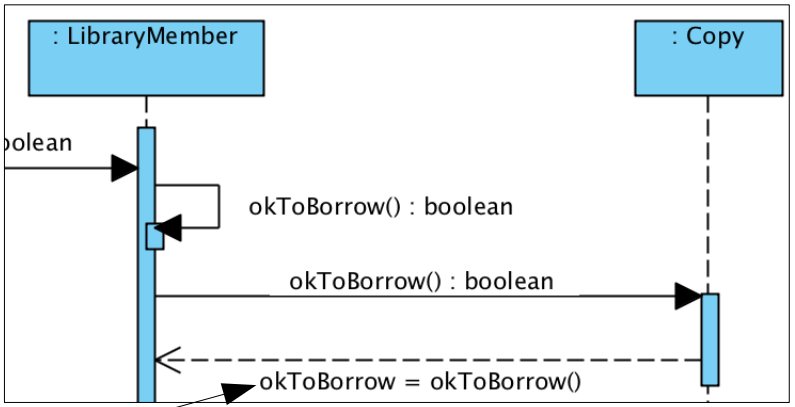
Ett svar på ett meddelande skickas när rektangeln som visar att objektet är aktivt slutar. Man kan även visa detta med pilar för tydlighetens skull.

Ibland vill man visa returvärdet genom att skriva ut detta vid returpilen.

Exempel: Vi har en metod okToBorrow() som kontrollerar om det går bra att låna en kopia. Returvärdet tilldelas variablen `okToBorrow`:

```
    okToBorrow = okToBorrow()
```

vid returpilen.



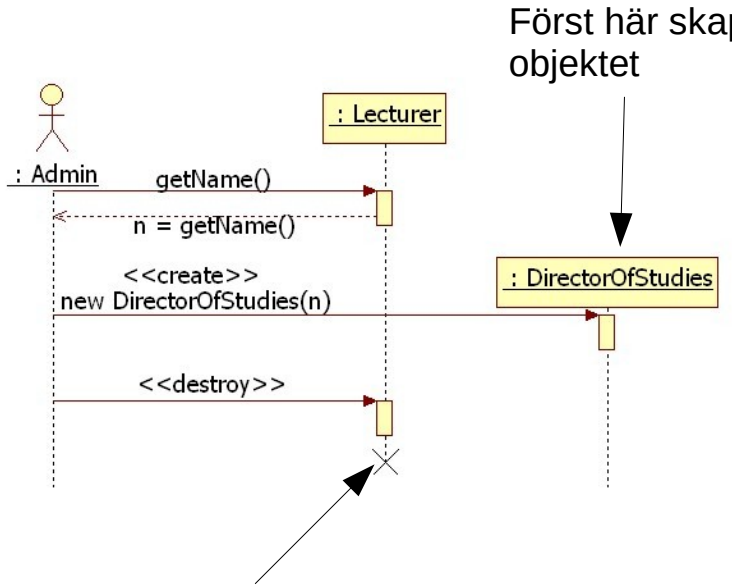
MALMÖ UNIVERSITET

Att skapa och förstöra objekt

I sekvensdiagram kan vi även visa att objekt skapas eller förstörs.

Detta visas på flera sätt:

- som stereotyper `<<create>>` eller `<<destroy>>` på pilarna
- genom att en pil pekar på objektet i sig istället för dess livslinje när objektet skapas och när ett objekt förstörs så pekar pilen på ett kryss som avslutar livslinjen
- som en streckad öppen pil (samma som retur)



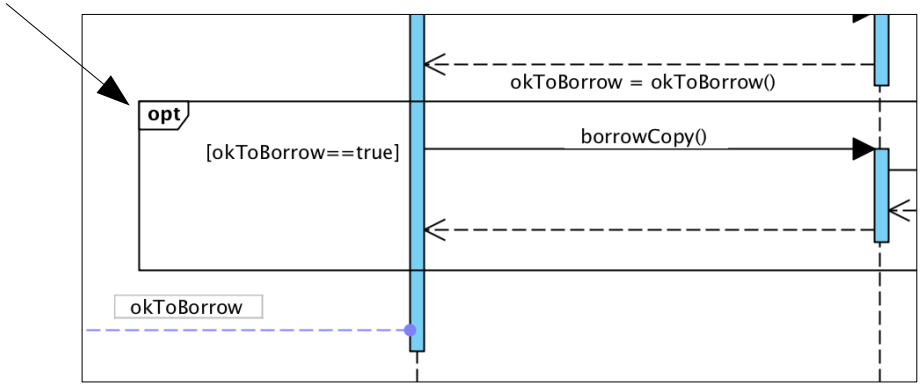
MALMÖ UNIVERSITET

Sekvensdiagrammets beståndsdelar

Ett **interaktionsfragment** är en speciell region i ett sekvensdiagram. Det kan referera till andra sekvensdiagram eller införa kontrollstrukturer i ett sekvensdiagram.

Samtliga berörda objekt täcks in av fragmentet.

Fragmentet inleds ofta med en **grind** – ett **villkor**.



MALMÖ UNIVERSITET

Fragment

Dessa är de fem vanligaste fragmenten:

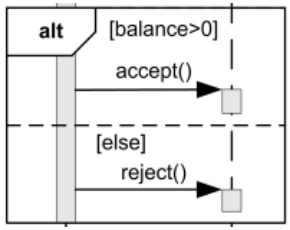
alt (alternative): en if-else-sats

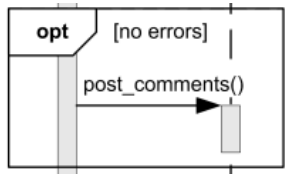
opt (optional): en if-sats

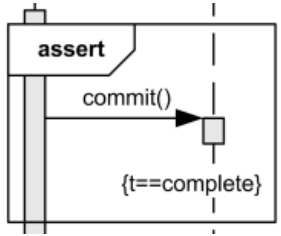
loop: en loop

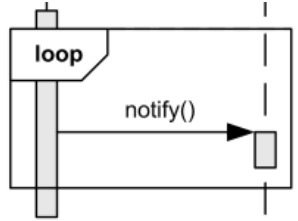
assert: säkerställ att något är sant

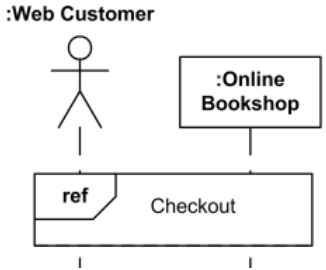
ref: referera till ett annat sekvensdiagram – ett **interaktionsfragment**











MALMÖ UNIVERSITET

Fragment

Övriga fragment:

break: bryter en loop

par: kör saker parallellt

strict: kör saker i strikt ordning

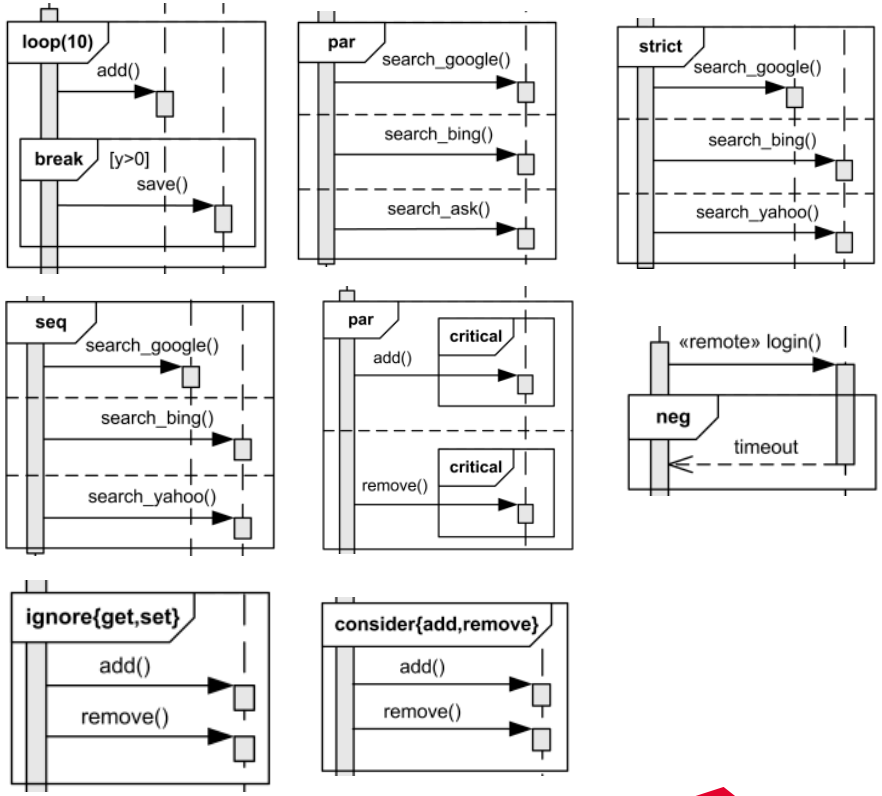
seq: kör saker i ordning

critical: en kritisk region, hantera atomärt

ignore: strunta i detta meddelande

consider: strunta i alla andra meddelanen

neg: Innebär att något går fel.



MALMÖ UNIVERSITET

Exempel: if-else

I ett sekvensdiagram vill vi visa alternativt beteende motsvarande en *if-else*-sats eller upprepade *if - else if- else* …

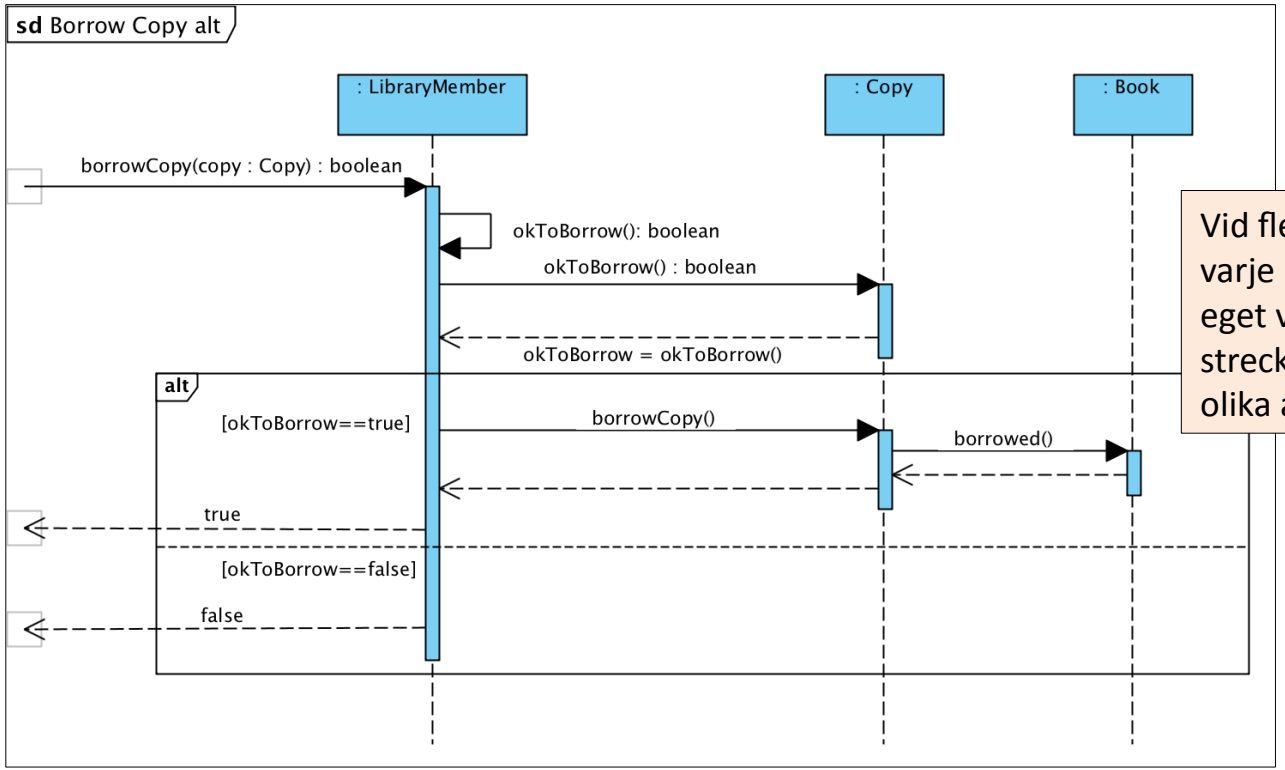
Detta görs genom att man ritar en ram – ett **Combined fragment** - runt det alternativa skeendet. Ramen ges operatorn **alt**.

Med ett **villkor** (**guard** eller **interaction constraint**) anger man när selektionen ska utföras respektive alternativet utföras.

Om *else*/*if else* saknas, används istället **opt**.

MALMÖ UNIVERSITET

Användningsfall: låna en bok



Vid flera val visar man varje möjlighet med ett eget villkor där en streckad linje skiljer de olika alternativen åt.

MALMÖ UNIVERSITET

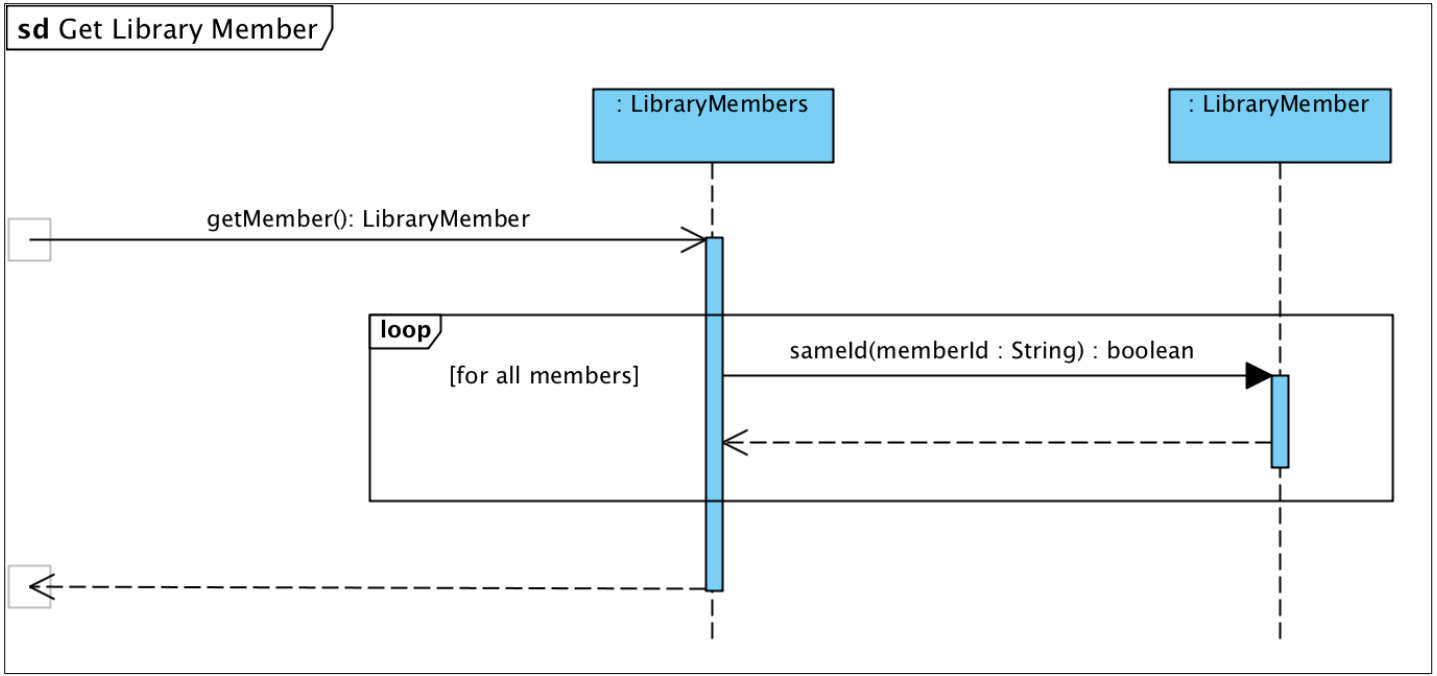
Exempel: loopar/iterationer

Ibland vill vi visa att något utförs upprepade gånger men vi kan inte säga exakt hur många gånger, då detta beror på variabler under körning.

Vi inför då ett **Combined Fragment** som visar att vissa anrop skall upprepas i en **loop**. Ramen ges operatorn **loop**.

Med ett **villkor** (**guard** eller **interaction constraint**) anger man för vilka element iterationen ska upprepas.

Användningsfall: låna en bok



MALMÖ UNIVERSITET

Asynkrona anrop

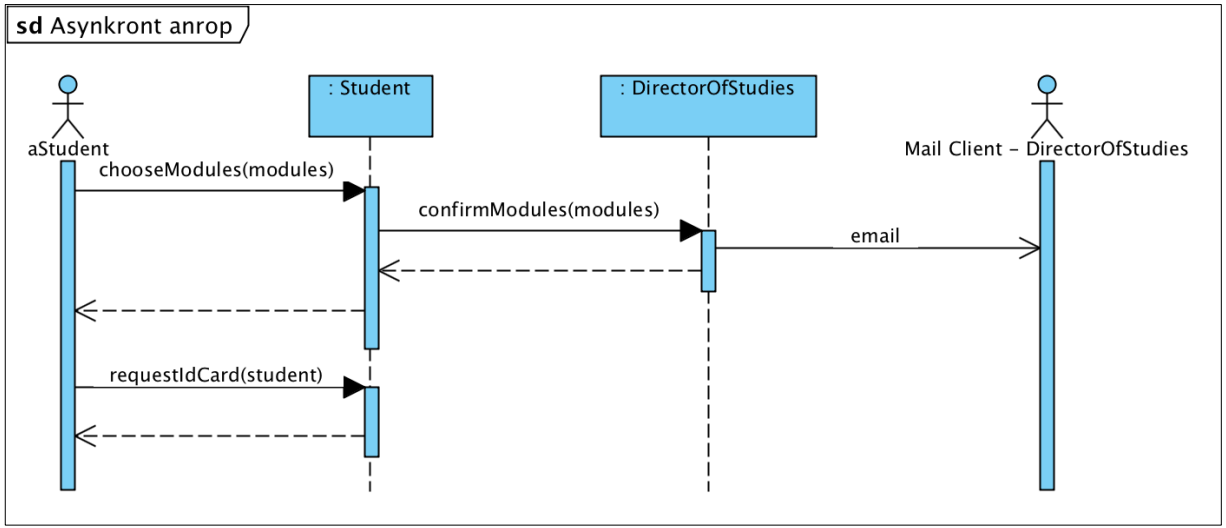
I sekvensdiagram kan man även visa **asynkrona anrop**, det vill säga vi skickar ett meddelande som vi inte väntar på något svar från.

Det objekt som skickar meddelandet förlorar inte kontrollen. Istället delas kontrollen i flera trådar som exekverar samtidigt.

Ett asynkront meddelande visas med en öppen pil till en aktör i systemet eller som systemet kommunicerar med.

Användningsfall: maila studierektorn

I anropet confirmModules(modules) skickas ett email till studierektorn. Efter utskicket fortsätter exekveringen i confirmModules. confirmModules avslutas och exekveringen i chooseModules fortsätter efter anropet till confirmModules.



MALMÖ UNIVERSITET

Slutligen

MALMÖ UNIVERSITET

Modellering: Gemensamt exempel

Ett program som håller ordning på en telefonbok skall skrivas. I telefonboken lagras namn, adress och telefonnummer för ett antal personer. Följande instruktioner skall finnas i programmet:

- Lägg in en ny person i telefonboken
- Tag bort en person
- Ändra uppgifter om en person
- Tag reda på alla uppgifter om en person
- Slå numret till en person

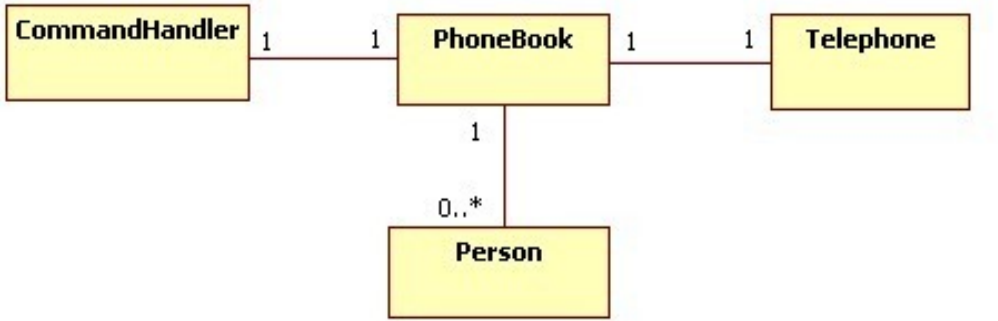
Modellering: Gemensamt exempel

Ett program som håller ordning på en telefonbok skall skrivas. I telefonboken lagras namn, adress och telefonnummer för ett antal personer. Man har funnit följande klasser lämpliga för programmet:

- PhoneBook – telefonboken som håller ordning på alla personerna.
- Person – en person med namn, adress och telefonnummer
- Telephone – telefonen i sig som är ansluten till vårt program.
- CommandHandler – vårt gränssnitt i vilket man ger kommandon till telefonboken

Modellering: Gemensamt exempel

Vi har följande klassdiagram:



- Vilka attribut och operationer är lämpliga?
- Hur skulle ett sekvensdiagram som visar hur man hittar en person i telefonboken se ut?

MALMÖ UNIVERSITET

Läsanvisningar

Object-Oriented Systems Analysis and Design Using UML

- 9.1-3 Sekvensdiagram
- 9.7 Om konsistens i diagram
- 11.1-3 Tillståndsdiagram

MALMÖ UNIVERSITET