

**Extentamen i**  
**2D1359 Objektorinterad modellering programmering och analys**  
**Tisdag den 13 oktober 1998 kl. 11-13**

**Inga hjälpmedel tillåtna.**

**Poäng och betyg:**

För godkänt krävs minst 10 poäng på vardera av teoridelen och problemdelen.  
För betyget 3, 4 och 5 krävs 20, 26 respektive 32 poäng (av möjliga 40).

**Övriga anvisningar:**

*Teoridelen* besvaras på dom prickade linjerna under respektive fråga och lämnas in tillsammans med övriga svarsblad.  
Lösningar till *problemdelen* lämnas in på separata blad.  
Frågorna ej säkert i svårighetsordning!

**Lösningar:**

Lösningsskisser på <http://www.nada.kth.se/kurser/kth/2D1359>.

*Lycka till!*

**Namn:** .....

**Personnummer:** .....

**Teoridel (svaren direkt på lydelsen)**

1. (4) Arv är centralt i objektorienterade språk och utvecklingsmetoder.

a) Vad innebär det att en klass i ett objektorienterat språk ärver från en annan.

.....  
.....  
.....  
.....

Man kan också tala om olika typer av arv. Förklara med en mening per delfråga vad de olika typerna av arv i **b-g** nedan innebär:

b) Arv för specifikation (eng. inheritance for specification)

.....  
.....

Vänd!

**c)** Arv för specialisering (eng. inheritance for specialization)

.....  
.....

**d)** Arv för konstruktion (eng. inheritance for construction)

.....  
.....

**e)** Arv för utvidgning (eng. inheritance for extension)

.....  
.....

**f)** Arv för begränsning (eng. inheritance for limitation)

.....  
.....

**g)** Arv för kombinerings (eng. inheritance for combination)

.....  
.....

**2.** (5) Förklara vad följande begrepp innebär inom objektorienterad programmering:

**a)** Polymorfi

.....  
.....  
.....

**b)** Abstrakt klass

.....  
.....  
.....

**c)** Abstrakt metod

.....  
.....

.....  
**d) Överskrivning (eng. overriding)**  
.....  
.....  
.....

**e) Överlagring (eng. overloading)**  
.....  
.....  
.....

**3. (2) Förklara kort begreppen *datadriven* och *ansvarsdriven design* samt vilken inverkan respektive metod har på programkonstruktionen.**  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

**4. (3) Diskutera vilka konsekvenser användning av *komposition* respektive *arv* har på designen och underhållet av ett program.**  
.....  
.....  
.....  
.....  
.....  
.....  
.....



.....

.....

.....

.....

.....

.....

## Problemdel (svaren på separata blad)

7. (4) Använd den fragmentariska beskrivningen av klassen `Vector` nedan för att konstruera en klass `Stack` med metoderna (beskrivningen på formen *returtyp metodnamn(argumenttyp)*):

- `boolean isEmpty()` // Är stacken tom?
- `Object push(Object)` // Lägg ett element på stacken och returnera det
- `Object peek()` // Vad ligger överst på stacken
- `Object pop()` // Ta bort översta objektet från stacken

<pre>public class Vector {     //Är vektorn tom?     public boolean isEmpty() {...}      //Hur stor är vektorn?     public int size() {...}      //Ingår elem i vektorn?     public boolean contains(Object elem) {...}      //Vilket element finns vid index?     public Object elementAt(int index) {...}</pre>	<pre>//Första elementet public Object firstElement() {...}  //Sista elementet public Object lastElement() {...}  //lägg till obj sist i vektorn public void addElement(Object obj) {...}  //ta bort obj från vektorn public boolean removeElement(Object obj) {...} ... }</pre>
---	---

Genom att:

- a) implementera stacken som subclass till `Vector`
- b) använda komposition via en instans av `Vector`.

8. (4) Klass- och samarbetsdiagram:

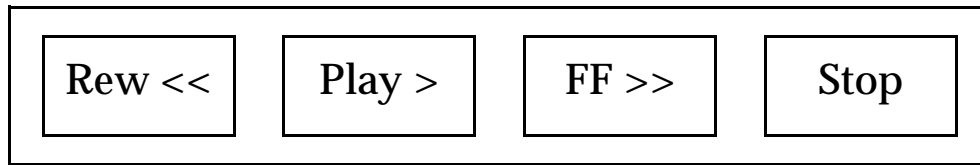
- a) Konstruera ett klassdiagram med företag och personer. Där:
  - ett företag har noll eller flera anställda
  - varje person jobbar på högst fyra olika företag
  - varje anställd har en chef som sin tur har en noll eller flera underställda.

Sätt också ut:

- lämpliga rollnamn
  - namn på eventuella associationer samt riktningar på dessa.
- b) Ett samarbetsdiagram (eng. collaboration diagram) som illustrerar följande enkla scenarie:

En person (P) ber om "audiens" hos ett företag via dess personalavdelning (PA).  
PA meddelar P om att han kan komma för intervju hos sin eventuellt blivande chef C kl K dagen D.  
Personen kommer till receptionen (R) och frågar efter C.  
Receptionen kontrollerar om C är inne.  
Är C inne så meddelar R C att P är på väg upp  
annars frågar R PA om ny tid och meddelar P den nya tiden.

9. (6) En bandspelare har följande knappar som kontrollerar motorn:



där Rew är Rewind och FF är Fast Forward.

Stop stoppar bandet mjukt. Om användaren byter riktning utan att först stoppa bandet (t ex om man trycker Play och sedan Rew) ska motorlogiken stoppa bandet automatiskt för att undvika bandbrott.

Play-Rew ska alltså utföras som Play-Stop-Rew, Rew-FF som Rew-Stop-FF osv. Däremot behöver inte bandet stoppas om man trycker Play-FF, eftersom spelriktningen då är oförändrad.

Antag att det finns ett objekt **panel** som skickar meddelandena *rew*, *play*, *ff* och *stop* vidare till ett objekt **control** som följd av att användaren trycker på knapparna. Objektet **control** i sin tur skickar meddelandena *forward*, *fast forward*, *fast backward* och *stop* till **motor** för att kontrollera bandets hastighet.

**a)** Konstruera ett *sekvensdiagram* som visar meddelandesändningen mellan de tre ovan nämnda objekten. Täck speciellt in följande sekvens av knapptryckningar på panelen: play, stop, ff, rew, play.

**b)** Konstruera ett *tillståndsdigram* som beskriver motorns olika tillstånd samt övergångar mellan dessa. Hantera dom tillfälliga stoppen mellan fram- och bakåtpolning med speciella stopptillstånd som tillfälligt stoppar bandet och efter tiden  $t$  automatiskt går över i nästa tillstånd. Tex fastforward till fast backward ger sekvensen: fast forward, tillfälligt stopp, efter tiden  $t$  övergång till fast backward.

**10. (6)** Antag att vi skall konstruera ett system för att hantera studenter, anställda, kurser, tentor och inrapportering till LADOK

Följande lista och kort-korta beskrivningar av ingående objekt har vi fått från den administrativa avdelningen. I övrigt är vi fria att designa systemet efter vår egen förmåga och införa nya objekt om vi anser att detta behövs. Fast ett sidokrav från den administrativa avdelningen är att varje deluppgift skall få plats på högst en A4 var. (I ett senare skede kan dom tänka sig att vi gör mer detaljerade beskrivningar.)

<b>Objekt</b>	<b>Beskrivning</b>
Student	<ul style="list-style-type: none"><li>• en person</li><li>• går kurser</li></ul>
Lärare	<ul style="list-style-type: none"><li>• en person</li><li>• håller kurser</li><li>• har lön</li><li>• rapporterar in resultat till LADOK</li></ul>
Kurs	<ul style="list-style-type: none"><li>• har deltagare</li><li>• har en kursledare/lärare</li><li>• har kurskod</li></ul>
Studievägledning	<ul style="list-style-type: none"><li>• ansvarar för tentamensbokning</li></ul>
Tentamen	<ul style="list-style-type: none"><li>• för kurs och moment</li><li>• anmälda tentander</li><li>• sal</li><li>• salsvakt, som bla skall ha lön</li></ul>
LADOK	<ul style="list-style-type: none"><li>• inrapportering av kursmoment</li></ul>

- Konstruera CRC-kort över problemet.
- Konstruera ett klassdiagram med lämpliga arvshierarkier, associationer och beroenden. Diagrammet skall vara på (konceptuell) analysnivå.