



# TENTAMEN

Luleå tekniska universitet

<b>Kurskod:</b> D0007N
<b>Kursnamn:</b> Objektorienterad Programutveckling
<b>Tentamensdatum:</b> 2016-08-24
<b>Skrivtid:</b> 5 tim
<b>Tillåtna hjälpmedel:</b> Paul J. Deitel, Harvey M. Deitel, Java How to Program (eller) en utskrift av motsvarande e-bok (och) Svensk-Engelsk Lexikon

<b>Jourhavande lärare m fullständigt telefonnr:</b> Ingemar Andersson 0920-491397 (070-3117550)	<b>Jourhavande lärare m fullständigt telefonnr:</b>
<b>Jourhavande lärare m fullständigt telefonnr:</b>	<b>Jourhavande lärare m fullständigt telefonnr:</b>

<b>Betygsgränser:</b>	U/G/VG G $\geq$ 60% - VG $\geq$ 80%
<b>Totalt antal uppgifter och poäng:</b>	4/18
<b>Övriga upplysningar:</b>	

## Allmänna anvisningar

Kontrollera att du fått samtliga uppgifter.  
Besvara endast en uppgift per lösningsblad.  
Skriv tydligt.

## Efter tentamen

Tentamensresultatet syns "Mina sidor" på Studentwebben.  
Examinationsresultat ska meddelas inom 15 arbetsdagar efter. För kurser med fler än 60 examinerande samt för distansutbildningar får resultatet meddelas senast än 20 arbetsdagar efter.

## Uppgifter till tryckeriet för tentor campus Luleå

<b>Projektnummer SRT:</b> 341980	<b>Hur många ex:</b> 5
<b>Hur många sidor:</b> 2 (inklusive försättsblad)	<b>Dubbel eller enkelsidigt:</b> dubbel

1. Skriv ett program där användaren först får frågan om hur många tal som ska hanteras och därefter får möjlighet att mata in ett värde för varje tal. Talen ska hanteras av en array:  

```
int[] talen = new int[n];
```

Programmet ska sedan skriva ut dessa tal i omvänd ordning. (3p)
2. Skriv ett program som slumpar 100 tärningskast och skriver ut frekvenserna för respektive resultat (d v s antal 1:or, antal 2:or, o s v). Till din hjälp behöver du en array med plats för 6 heltal, frekvenserna. (3p)  
Du slumpar ett tal i intervallet 1-6 med  

```
int x;
```

```
x = (int) (Math.random()*6 + 1);
```
3. Skapa ett program som innehåller klassen `Account` och klassen `ListOfAccounts` som håller reda på kontoobjekt. Lägg till följande funktionalitet i klassen `ListOfAccounts`:
  - a) En metod, `getNoOfAccounts`, som returnerar antalet objekt som för tillfället finns i listan. (2p)
  - b) En metod, `getAccount`, som tar ett index som argument och returnerar objektet på denna position, utan att ta bort det från listan. (2p)
  - c) En metod, `remove`, som givet ett index tar bort, och returnerar, objektet från listan. Denna metod ska se till att listan inte blir fragmenterad när objektet tas bort och att antalet objekt i listan uppdateras. (2p)
  - d) Skriv sedan en `main`-metod där du skapar ett `ListOfAccounts`-objekt, lägger till kontobjekt och testat de nya metoderna i klassen. (2p)
4. Förklara hur följande webbapplikation (se figur 4.1 och 4.2) byggd med ramverket JavaServer Faces (JSF) fungerar och vad som krävs för att spara meddelandet ex. "spara detta i databas" i en relationsdatabas när användaren trycker på knappen "Spara meddelande". Exemplifiera gärna med kod. (4p)



**Figur 4.1:** savemessage.xhtml - användargränssnittet för en JSF webbapplikation som sparar meddelanden i databas

```
<h:form>
  <h:outputLabel for="message" value="Meddelande:"/>
  <h:inputText id="message" value="#{messageBean.msg}"/>
  <h:commandButton id="save" value="Spara meddelande"
    action="#{messageBean.saveMessage()}" />
</h:form>
```

**Figur 4.2:** savemessage.xhtml – utdrag från koden för att generera användargränssnittet presenterad i figur 4.1