



TENTAMEN

Luleå tekniska universitet

Kurskod: D0007N
Kursnamn: Objektorienterad Programutveckling
Tentamensdatum: 2015-12-18
Skrivtid: 5 tim
Tillåtna hjälpmedel: Paul J. Deitel, Harvey M. Deitel, Java How to Program (eller) en utskrift av motsvarande e-bok (och) Svensk-Engelsk Lexikon

Jourhavande lärare m fullständigt telefonnr: Ingemar Andersson 0920-491397 (070-3117550)	Jourhavande lärare m fullständigt telefonnr:
Jourhavande lärare m fullständigt telefonnr:	Jourhavande lärare m fullständigt telefonnr:

Betygsgränser:	U/G/VG G \geq 60% - VG \geq 80%
Totalt antal uppgifter och poäng:	3/18
Övriga upplysningar:	

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

Projektnummer SRT: 341980	Hur många ex: 5
Hur många sidor: 3 (inklusive försättsblad)	Dubbel eller enkelsidigt: dubbel

1. Skriv den kod som saknas i main-metoden för att programmet nedan ska skriva ut följande: (3p)

```
x = 33
this.x = 1
SkuggTest.this.x = 0

public class SkuggTest {
    public int x = 0;
    class NivaEtt {
        public int x = 1;
        void metodNivaEtt(int x) {
            System.out.println("x = " + x);
            System.out.println("this.x = " + this.x);
            System.out.println("SkuggTest.this.x = " +
                SkuggTest.this.x);
        }
    }
    public static void main(String... args) {
        //Skriv din kod här
    }
}
```

2. Figur 2.2 illustrerar ett javaprogram (desktop) med en "JFrame" dialog som ger användaren möjlighet att skriva in ett ord i en textruta och när användaren klickar på knappen "Lägg till" ska programmet spara ordet ex. "java". Ordet som användaren skriver in i textrutan ska sparas i en instansvariabel som tillhör en separat klass med namnet "Ord". Varje gång användaren trycker på knappen "Lägg till" skapas en ny instans (objekt) av klassen Ord som håller reda på det ord som skrivs in i textrutan. För att hålla reda på alla instanser (objekt) av Ord som skapas skall dessa läsas in i en samling (ArrayList). Ett ord får bara vara med i samlingen om det är unikt dvs. det får inte finnas dubletter av ett ord i samlingen. Två ord som jämförs som lika av "equals"-metoden ska räknas som samma ord. Ordningsföljden på orden har ingen betydelse. Om användaren försöker lägga in fler än 5 ord ska ett eget definierat undantag "SamlingenFullException" kastas. Programmet behöver dock inte definiera denna "exception" bara visa hur den kan användas. När användaren klickar på knappen "Visa" ska programmet stega igenom (iterera) samlingen (ArrayList) av "Ord" objekt och skriva ut respektive ord med "System.out.println". När användaren klickar på knappen "Ta bort" ska objektet som håller ordet som användaren skrivit in i textrutan tas bort från samlingen (ArrayList). Om samlingen med ord är tom eller om objektet som håller ordet som ska tas bort inte finns i samlingen ska ett meddelande om detta visas i en "JOptionPane.showMessageDialog".

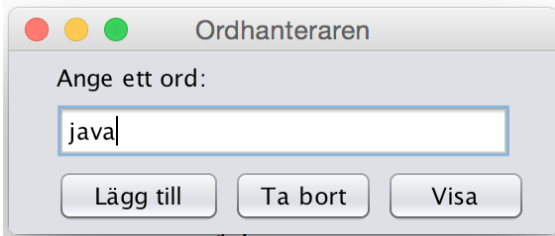
Skriv den kod som krävs för att respektive händelse (a-c) kopplad till knapparna i programmet ska fungera (figur 2.1 nedan visar exempel på start av händelse för knappen "Lägg till"). Observera att du inte behöver skriva den kod som genererar gränssnittet ex. `txtWord = new javax.swing.JTextField();` etc. endast koden för respektive händelse a-c.

```
private void btnAddWordActionPerformed(java.awt.event.ActionEvent evt) {
    // Din kod startar här
}
```

Figur 2.1: Händelse som exekveras när användaren trycker på knappen "Lägg till".

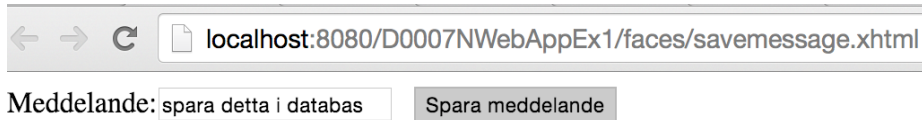
- Lägg till" (5p)
- "Ta bort" (3p)

c) "Visa" (2p)



Figur 2.2: Ordhanteraren - ett litet javaprogram för att hantera ord

3. Förklara hur denna webbapplikation (se figur 3.1 och 3.2) 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. (5p)



Figur 3.1: 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 3.2: savemessage.xhtml – utdrag från koden för att generera användargränssnittet presenterad i figur 3.1