

Minitentamen 1 på kursen DA339, Objektorienterad programmering

Lördagen den 4 oktober 2014, 10.15-13.15

Hjälpmedel: Inga hjälpmedel är tillåtna

Tentamen:

Uppgifterna 1a – 1d och 4a – 4c besvaras genom att du anger de alternativ som du anser är korrekta (använd bokstäverna)

Om du anger samtliga korrekta alternativ på en uppgift får du 2 p. Om du anger något felaktigt alternativ ger det poängavdrag. På samma sätt får du poängavdrag om du missar att ange något korrekt alternativ. Du får dock aldrig mindre än 0 poäng på en uppgift.

Instruktioner:

- Varje uppgift (7 st) ska du börja på ny sida
- Skriv endast på ena sidan av pappret
- Lämna in uppgifterna sorterade från 1 till 7
- **OBS! Denna instruktion skall INTE lämnas in!**
Alla svar ska ges på separat svarspapper.

Bedömning:

Maximal poäng på tentamen är 48 p. För godkänt resultat krävs 24 p, varav minst 7 p på Del 1 (Uppgifterna 1-3) och 7 p på Del 2 (Uppgifterna 4-7).

Del 1

Uppgift 1

1a Vilka av följande påståenden är korrekta? (2p)

- A Ett subsystem är en del av ett system som kan fungera som ett eget system.
- B Ett subsystem kan inte bestå av andra subsystem
- C I vattenfallsmodellen startar implementation (kodning) först när designfasen är klar.
- D Vattenfallsmodellen fungerar bra då kraven är tydliga och inte ändras under utvecklingstiden.
- E Vattenfallsmodellen är en agil modell.
- F Man måste använda ett informationssystem i lösningen som byggs med vattenfallsmodellen

1b Vilka av följande påståenden om objekt är korrekta? (2p)

- A Alla objekt representerar abstraktioner av något konkret.
- B Alla objekt måste ha någon operation som utför en matematisk beräkning.
- C Alla objekt har ett tillstånd.
- D Användbara frågor för att identifiera objekt är: Vem är jag, Vad vet jag och Vad kan jag göra?
- E Alla objekt har en unik identitet.
- F Inkapsling betyder att ett attribut inte kan ändra värde efter det att objektet skapats.

1c Vilka påståenden om modeller nedan är korrekta? (2p)

- A En modell kan inte användas för simuleringar
- B En modell måste kunna sammanfattas i ett enda diagram.
- C En användbar modell måste erbjuda rätt detaljdjup.
- D En modell kan synliggöra arbetsflöden i en organisation.
- E En modell är inte enklare/snabbare att bygga än det verkliga systemet.
- F En modell bör vara korrekt, konsistent och komplett (för någon detaljnivå)

1d Vilka påståenden nedan är korrekta? (2p)

- A Man kan inte använda pseudokod för att testa om en algoritm fungerar som man tänkt.
- B I pseudokod är det viktigt att vara konsekvent med hur man skriver.
- C Algoritmer går att återanvända mellan olika program.
- D Algoritmer kan beskrivas i olika detaljeringsgrad. Ett steg i en algoritm (beskriven på hög nivå), kan detaljeras i en annan algoritm.
- E Olika algoritmer kan användas för att beskriva olika lösningar på samma problem.
- F En algoritm är unik för just det programspråk (tex Java) som man implementerar algoritmen i.

1e Visa att: (3p)

$$!(a \mid b \mid c) = !a \& !b \& !c$$

1f Förklara skillnaden mellan analys och design (och beskriv vad som bör ingå i analysaktiviteten respektive designaktiviteten). (3p)

Uppgift 2 (5p)

Skapa ett aktivitetsdiagram som visar det som finns beskrivet i systembeskrivningen nedan

Systembeskrivning: Trouble ticket

När ett problem kommer in, så skapar man först ett ärende. Sedan försöker man återskapa felet. Om felet inte kan återskapas begär man mer information och uppdaterar ärendet. Om problemet kan verifieras men problemet och dess lösning är kända behöver man inte hitta någon ny lösning. Om problemet kan verifieras och problemet är ett duplikat av ett redan känt problem, räcker det att verifiera lösningen för det kända problemet på den nya situationen. Om problemet kan verifieras och är tidigare okänt, skall man först hitta en lösning, sedan verifiera denna. Om lösningen inte kan verifieras, skall ny lösning hittas. Då verifierad lösning finns, skall denna kommuniceras ut och statistiken över alla ärenden uppdateras. Sedan skall ärendet stängas.

Uppgift 3 (5p)

Skriv pseudokod för följande problem:

Programmet skall skriva ut numren 1 till 100. För nummer som är delbara med 3, skall programmet skriva "Fizz" istället för numret och för nummer som är delbara med 5, skall programmet skriva "Buzz" istället för numret. För nummer som är delbara med både 3 och 5, skall programmet skriva "FizzBuzz" istället för numret.

Källa:

Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

Del 2

Uppgift 4

4a Vilka påståenden är korrekta för java?

(2p)

- A Java är plattformsoberoende
- B Programmet `javac.exe` kan översätta källkodsfiler till bytekodsfiler
- C Java används endast i skolor och på universitet/högskolor
- D Kompilatorn kan exekvera class-filer
- E Java Virtual Machine används för att editera källkodsfiler
- F Programmet `java.exe` används för kontroll av indentering i källkoden

Följande program är givet. Raderna inleds med radnummer vilka inte tillhör programmet:

```
1  /*
2   * Programmet används i uppgift 4b.
3   */
4  public class Program4b {
5      public void action() {
6          int nbr1 = 10;
7          double nbr2;
8
9          nbr2 = 2 + nbr1;
10         nbr1 = 2 * nbr1;
11
12         System.out.println(nbr1 + "," + nbr2); // utskrift
13     }
14     public static void main(String[] args) {
15
16         Program4b p4b = new Program4b();
17         p4b.action();
18     }
19 }
```

4b Vilka påståenden är korrekta om programmet ovan?

(2p)

- A `nbr1` är en variabel i metoden `action`
- B På rad 7 dubblas innehållet i `nbr2`
- C Instruktionerna exekveras i följande ordning: 6, 7, 8, 9, 11, 14, 15
- D Klassen heter *Program4b*
- E `p4b` är en *referensvariabel*
- F På rad 15 anropas metoden `action`.

4c Vilka av följande påståenden är korrekta?

(2p)

- A Klassen *Random* kan användas då man behöver slumpital i ett program
- B Klassen *Calendar* kan användas till att ta reda på dagens datum
- C Ett objekt av typen *String* kan innehålla 0 tecken
- D Med metoden *Double.parseDouble* kan man avrunda ett flyttal till 2 decimaler
- E Med metoden *JOptionPane.showMessageDialog* kan man visa ett textmeddelande i ett dialogfönster
- F Med metoden *Math.max* kan man erhålla det största tänkbara heltalet

Uppgift 5

5a Vilka utskrifter är möjliga då nedanstående kod exekverats? Motivera ditt svar. (2p)

```
String str;  
int nbr;  
do {  
    str = JOptionPane.showInputDialog("Mata in ett heltal" );  
    nbr = Integer.parseInt( str );  
} while( nbr<1 || nbr >6);  
System.out.println( nbr );
```

5b För vilka värden på nbr är följande villkor *true*? (2p)

A(nbr >= 20) && (nbr < 30)

B(nbr < 4) || (nbr > 2)

5c Vilket blir körresultatet då nedanstående kod exekveras? Motivera ditt svar. (2p)

```
int a = 10, b = 14, sum = 0;  
while(b>a) {  
    sum = sum + a;  
    System.out.println( sum + " " + a + " " + b);  
    a--;  
    b-=2;  
}
```

5d Vilket blir körresultatet då nedanstående kod exekveras? Motivera ditt svar. (2p)

```
for(int i = 2; i<5; i++) {  
    for(int j=5; j>=i; j--) {  
        System.out.print(j);  
    }  
    System.out.println();  
}
```

Uppgift 6

6a Skriv en for-loop som ger utskriften:

(2p)

10 8 6 4

I varje iteration ska exakt ett tal skrivas ut.

6b Skriv switch-satsen med if-else-sats

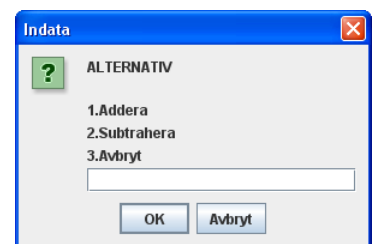
(2p)

```
switch( nbr ) {  
    case 4 :  
        System.out.println("fyra");  
        break;  
    case 6 : case 7 :  
        System.out.println("sex eller sju");  
        break;  
}
```

6c Du har fått i uppgift att komplettera med kod vilken låter användaren välja ett alternativ ur en meny. Valet ska ske med hjälp av en inmatningsdialog (se figur) och valet ska lagras i variabeln *choice*. Om användaren skriver ett alternativ som inte finns i menyn (mindre är 1 eller större än 3) så ska användaren få ange alternativ på nytt. Du kan utgå från att användaren alltid matar in text som kan konverteras till ett heltal.

(2p)

```
String menu = "ALTERNATIV\n\n1.Addera\n2.Subtrahera\n3.Avbrt";  
int choice;  
// lägg till kod som uträttar ovanstående
```



Uppgift 7 (4p)

Du ska komplettera klassen Uppgift7 (se nedan) med kod.

Programmet ska låta användaren mata in en sträng. Sedan ska den inmatade strängen användas för lite information som ges via en meddelandedialog:

- A. Antalet tecken i String-objektet ska skrivas ut
- B. Om String-objektet innehåller minst 4 tecken ska det fjärde tecknet skrivas ut.
- C. Tecknen i strängen ska skrivas ut baklänges



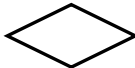
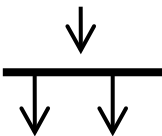
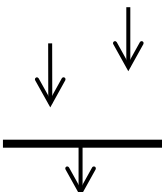


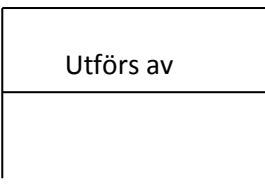
Du får inte ändra i main-metoden.

Så här kan en programkörning vara:



```
public class Uppgift7 {  
    public void action() {  
        // Komplettera med kod här  
    }  
  
    public static void main(String[] args) {  
        Uppgift7 u7 = new Uppgift7();  
        u7.action();  
    }  
}
```

Bilaga 1: Notation aktivitetsdiagram

aktivitet activity	
flöde control flow	
beslutsromb eller samlare av flöden decision node /decision diamond/merge node	
villkor guard condition	[villkor]
flödesdelare fork	
flödessamlare join	
stardnod initial node	
slutnod final node	
swimlane	

Bilaga 2: Lite kort om olika klasser

JOptionPane.showInputDialog(String message)

Visar *message* i ett dialogfönster. Returnerar inmatad sträng

JOptionPane.showMessageDialog(null, String message)

Visar *message* i ett dialogfönster

Integer.parseInt(String str)

Översätter *str* till ett *int*-värde och returnerar värdet

Long.parseLong(String str)

Översätter *str* till ett *long*-värde och returnerar värdet

Double.parseDouble(String str)

Översätter *str* till ett *double*-värde och returnerar värdet

Math.pow(bas,exp)

Beräknar och returnerar bas^{exp}

Math.sqrt(value)

Beräknar och returnerar *kvadratroten ur value*

Math.min(nbr1, nbr2)

Beräknar och returnerar det minsta värdet av *nbr1* och *nbr2*

Math.max(nbr1, nbr2)

Beräknar och returnerar det största värdet av *nbr1* och *nbr2*

Calendar cal = Calendar.getInstance();

Variabeln *cal* tilldelas ett Calendar-objekt

cal.get(Calendar.YEAR)

Returnerar året

cal.get(Calendar.MONTH)

Returnerar månaden (0-11)

cal.get(Calendar.DAY_OF_WEEK)

Returnerar dagens nummer i veckan (0-6).

cal.get(Calendar.DAY_OF_MONTH)

Returnerar dagens nummer i månaden (1-31)

cal.get(Calendar.DAY_OF_YEAR)

Returnerar dagens nummer i året. (1-365)

cal.get(Calendar.HOUR_OF_DAY)

Timme på dagen (0-23)

cal.get(Calendar.MINUTE)

Minut i timmen (0-59)

cal.get(Calendar.SECOND)

Sekund i minuten. (0-59)

Random rand = new Random();

Variabeln *rand* tilldelas referens till ett Random-objekt

rand.nextInt(int value)

Returnerar ett slumptal (heltal) som är minst 0 och högst ett mindre än *value*

rand.nextDouble()

Returnerar ett slumptal i intervallet 0 – 0.99999999...

String str = "Minitentamen";

Variabeln *str* tilldelas referens till ett String-objekt

str.length()

Returnerar antalet tecken i strängen *str*

str.charAt(int pos)

Returnerar tecknet i angiven position i strängen *str*

str.equals(String anotherStr)

Jämför strängarna *str* och *anotherStr*. Returnerar *true* om de innehåller samma teckensekvens och annars *false*.

str.substring(int start, int end)

Returnera en ny sträng som innehåller tecknen i *str*, från positionen *start* till positionen (*end*-1)

str.toLowerCase()

Returnerar en ny sträng där alla tecken i *str* är gemener

str.toUpperCase()

Returnerar en ny sträng där alla tecken i *str* är versaler