Tentamen för TDA143 PROGRAMMERADE SYSTEM

DAG: 16-08-19 TID: 8:30 – 13:30

Ansvarig: Christer Carlsson, ankn 1038

Förfrågningar: Christer Carlsson

Resultat: erhålls via Ladok

Allmän info: tentamen är uppdelad i två delar: del 1 omfattar översikt av datateknik och del 2 omfattar programmering

Betygsgränser:

3:a	10 poäng på del 1 och 26 poäng på del 2
4:a	13 poäng på del 1 och 36 poäng på del 2
5:a	16 poäng på del 1 och 48 poäng på del 2
maxpoäng	20 poäng på del 1 och 60 poäng på del 2

Siffror inom parentes: anger maximal poäng på uppgiften.

Granskning: Onsdag 14/9 kl 12-13 och onsdag 21/9 kl 12-13, rum 6128 i EDIT-huset.

Hjälpmedel: En valfri lärobok i Java eller de på kursen utdelade föreläsningsanteckningarna (OH-bilderna) som behandlar **programmeringsdelen** av kursen.

Förtydligande noteringar får finnas i boken resp. föreläsningsanteckningarna.

Var vänlig och: Skriv tydligt och disponera papperet på lämpligt sätt.

Börja varje uppgift på nytt blad. Skriv ej på baksidan av papperet.

Observera: Uppgifterna är ej ordnade efter svårighetsgrad. Titta därför igenom hela tentamen innan du börjar skriva.

Alla program skall vara väl strukturerade, lätta att överskåda samt enkla att förstå.

Vid rättning av uppgifter där programkod ingår bedöms principiella fel allvarligare än smärre språkfel.

På de programmeringsuppgifter som ingår i tentamenstesen kan vissa poäng erhållas även om ett fullständigt program inte redovisas, detta kräver dock att en algoritm som löser problemet presenteras.

LYCKA TILL!!!!

Uppgift 1.

Denna uppgift består av 10 påståenden. Frågorna skall besvaras med ”sant” eller ”falskt”.

Ett korrekt svar ger 1 poäng, ett felaktigt svar ger -0.5 poäng och ett utelämnat svar ger 0 poäng. Totalt på uppgiften kan aldrig minuspoäng erhållas.

a) $500_{10} = 1F4_{16} = 111110100_2$.

b) Applikationslagret förbereder meddelanden och skickar dessa sedan till länklagret.

c) En *router* är en nätverksnod som kopplar ihop två olika nätverk som använder sig av olika nätverksprotokoll.

d) En stack är en datastruktur som kännetecknas av ett LIFO-beteende (last-in-first-out).

e) I ett binärt sökträd finns alltid det minsta värdet i ett löv.

f) Open-source development är när man släpper en tidig version av en programvara till en liten grupp, för att få viktig kritik för den fortsatta utvecklingen av programmet.

g) SQL är ett deklarativt programspråk.

h) Utvecklingen under de senaste åren inom mjukvaruindustrin gör det allt viktigare att ha större och mer strukturerade utvecklingsteam för att kunna säkerställa att kvalitén på slutprodukten blir så hög som möjligt.

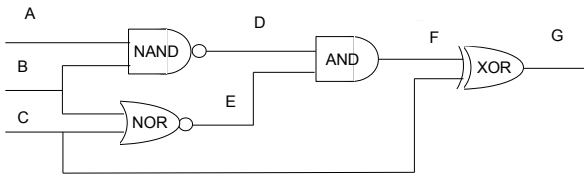
i) Cirka 60-70% av alla funktioner i ett typiskt mjukvarusystem används sällan eller aldrig.

j) Skuggor är ett viktigt inslag för att en datorgenererad bild skall se realistisk ut. Att skapa en hård skugga kräver betydligt mer beräkningskapacitet än att skapa en mjuk skugga.

(10 poäng)

Uppgift 2.

Skriv ut sanningstabellen för nedanstående krets. Inför nödvändiga beteckningar.



(2 poäng)

Uppgift 3.

Redogör kort för Moore’s lag och dess effekter.

(2 poäng)

Uppgift 4.

I media brukar alla former av skadliga programvara kallas *virus*. I verkliga fall är virus endast en typ av flera. Nämn och beskriv två exempel på sådana skadliga programvaror.

(2 poäng)

Uppgift 5.

a) Vad är tidskomplexiteten för följande kodavsnitt? (1 poäng)

```
int sum = 0;
for (int i =1; i <= n;  i = i + 1)
   sum = sum + i*i;
for (int i =1; i <= n;  i = i + 1)
   for (int j =1; j <= n;  j = j + 1)
      sum = sum + i*j;
```

b) Vad är tidskomplexiteten för följande kodavsnitt? (1 poäng)

```
int sum = 0;
for (int i =1; i <= n;  i = i + 1)
   for (int j =i; j <= n;  j = j + 1)
      sum = sum + i*j;
```

Uppgift 6.

Antag att du har följande databastabeller:

Games:

Name	Gene	AgeLimit
Grand Theft Auto	Action	18
Halo 3	Action	15
Rock Band	Music	7
Farmville	Action	13
Mass Effect	Roll Playing Game	15
Oblivion	Roll Playing Game	11

Inventory:

Item	Quantity
Grand Theft Auto	5
Halo 3	4
Rock Band	8
Farmville	8
Mass Effect	2
Oblivion	7

Hur ser tabellen ut som erhålls från nedanstående SQL-query?

```
SELECT Name, Quantity
FROM Games, Inventory
WHERE (Gene = 'Action' OR Gene = 'Roll Playing Game') AND AgeLimit < 18 AND Name = Item
```

(2 poäng)

DEL 2: Programmeringsteknik

Uppgift 7.

a) Vad blir utskriften när nedanstående program exekveras?

```
public class Uppgift7A {
  public static void mystery (int[] a) {
      for (int i = 1; i < a.length - 1; i++) {
          a[i] = (a[i - 1] + a[i + 1]) / 2;
      }
  }//mystery

  public static void main (String[] args) {
      int[] arr = {6, 13, 0, 3, 7};
      mystery(arr);
      System.out.println(java.util.Arrays.toString(arr));
  }//main
}//Uppgift7A
```

(2 poäng)

b) Betrakta nedanstående program:

```
public class  Uppgift7B {
  private int  biggest( int [] vekt) {
      int  max = vekt[0];
      for  (int  i = 1; i < vekt.length; i = i + 1)
          if  (vekt[i] > max)
              max = vekt[i];
      return  max;
  }//biggest

  public static void  main (String[] args) {
      int [] v = {4, 7, 3, 11, 7, 9};
      System.out.println("Det största talet i fältet är " + biggest(v));
  }//main
}//Uppgift7B
```

När programmet kompileras erhålls ett felmeddelande enligt:

non-static method cannot be referenced from a static context

Förklara vad som är felaktigt i programmet och ge (minst ett) förslag på hur felet kan rättas till. (2 poäng)

c) Betrakta nedanstående klasser:

```
public class Pair {
    private int x;
    private static int y;

    public Pair(int x, int y) {
        this.x = x;
        this.y = y;
    }//constructor

    public void changeValues(int x, int y) {
        this.x = x;
        this.y = y;
    }//changeValues

    public String toString() {
        return "x = " + x + ", y = " + y;
    }//toString
}//Pair
```

```
public class Main {
    public static void main(String[] arg) {
        Pair p1 = new Pair(5, 9);
        Pair p2 = new Pair(8, 2);
        p1.changeValues(1, 10);
        System.out.println(p1);
        System.out.println(p2);
    }//main
}//Main
```

Vilken utskrift fås när main-metoden i klassen Main exekveras? (2 poäng)

d) Vad blir utskriften när nedanstående program exekveras?

```
import java.util.Scanner;
public class Uppgift7D {
   public static void main(String[] args) {
        String str = "life is wonderful";
        System.out.println(mystery(str));
   } //main

   public static String mystery(String sentence) {
        String newStr = "";
        Scanner sc = new Scanner(sentence);
        while(sc.hasNext()) {
            newStr = sc.next() + " " + newStr;
        }
        return newStr;
   }//mystery
}//Uppgift7D
```

(2 poäng)

Uppgift 8

I ett program som administrerar studenters studieresultat finns en klass Course för att avbilda en kurs. Klassen tillhandahåller följande:

public Course(String code, String name, **double** credits)	konstruktor som tar kurskod, kursnamn och kurspoäng som argument
public String getCode()	returnerar kursens kod
public String getName()	returnerar kursens namn
public double getCredits()	returnerar kursens poäng
public String toString()	returnerar en strängrepresentation av kursen på fomen: TDA143 Programmerade system, 7.5 poäng

Din uppgift är att skriva en klass Student, som lagrar information om en student och dennas avklarade kurser. Klassen Student skall ha följande instansvariabler (inga andra instansvariabler får finnas):

private String name	studentens namn
private String programme	utbildningsprogrammet studenten går på
private ArrayList<Course> passed	en lista av de kurser studenten klarat av

Klassen skall innehålla:

public Student(String name, String programmme)	konstruktor som har studentens namn och utbildningsprogram som argument
public String getName()	returnerar studentens namn
public String getProgramme()	returnerar students utbildningsprogram
public void addCourse(Course c)	lägga in kursen c i listan av avklarade kurser
public int getNrPassed()	returnerar antalet avklarade kurser
public double getTotalCredits()	returnerar totalt antal avklarade poäng
public boolean hasPassed(String code)	returnerar **true** om studenten läst kursen med koden code, annars returneras **false**
public String toString()	returnerar en strängrepresentation av kursen på fomen: Adam Andersson, Industriell ekonomi TDA143 Programmerade System, 7.5 poäng MVE012 Inledande matematik, 7.5 poäng MVE016 Matematisk analys i en variabel, 7.5 poäng

(13 poäng)

Uppgift 9.

Livsmedelsverket har gjort följande klassificering när det gäller andelen kalorier som kommer från fett i våra livsmedel:

Rött: Över 40 procent av kalorierna från fett
Gult: Mellan 30 och 40 procent av kalorierna från fett
Grönt: Max 30 procent av kalorierna från fett

Skriv ett program som läser in antalet kalorier och hur många gram fett en förpackning livsmedel innehåller. Programmet skall beräkna och skriva ut hur stor procentuell andel av kalorierna som kommer från fettet, samt livsmedelsverkets klassificering av produkten. Det gäller att 1 gram fett innehåller 7,7 kalorier.

Du får själv välja om du vill göra in- och utmatning via dialogrutor eller använda System.in respektive System.out (se exemplen nedan).

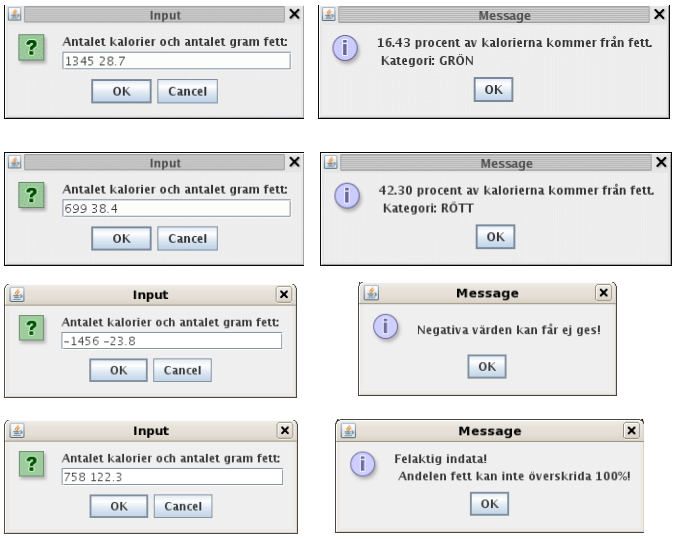
För att kunna erhålla full poäng på uppgiften:

- skall programmet utformas på så sätt att inläsningen upprepas tills användaren avbryter exekveringen (vid användning av dialogrutor genom att användaren trycker på Cancel-knappen och vid användning av System.in genom att användaren lämpligen ger ctrl z)
- skall antalet kalorier och antalet gram fett läsas med en inläsningssats (dvs ett Scanner-objekt skall användas).
- skall procenttalet i utskriften anges med *exakt* två decimaler
- skall programmet innehålla en metod

  public static double getCalories(**double** gram)

  som tar antalet gram fett och returnerar motsvarande antal kalorier
- skall kontroll göras på att indatavärdena inte är negativa
- skall kontroll göras på att andelen fett inte överskrider 100%

Med användning av dialogrutor **Med användning av System.in resp System.out**



Antalet kalorier och antalet gram fett: 1345 28.7
16.43 procent av kalorierna kommer från fett.
Kategori: GRÖN

Antalet kalorier och antalet gram fett: 699 38.4
42.30 procent av kalorierna kommer från fett.
Kategori: RÖTT

Antalet kalorier och antalet gram fett: -1456 -23.8
Negstiva värden får ej ges!

Antalet kalorier och antalet gram fett: 758 122.3
Felaktig indata!
Andelen fett kan inte överskrida 100%!

(11 poäng)

Uppgift 10.

Om man har två grupper av tal kan man ibland öka medelvärdet i båda grupperna genom att flytta ett tal från den ena gruppen till den andra. Skriv en metod

public static void increaseAverage(**int**[] a, **int**[] b)

som tar två grupper av tal representerade som heltalsfälten a och b. Om det är möjligt att öka medelvärdet i båda grupperna av tal skall metoden skriva ut vilket tal som skall flyttas vart, annars skall metoden skriva ut att att det inte är möjligt.

Om någon av fälten a eller b är **null** eller har längden 0 skall metoden kasta en IllegalArgumentException.

Exempel:

Antag att följande deklarationer har gjorts::

```
int[] groupA = {1, 2, 3};
int[] groupB = {4, 3, 4, 5};
int[] groupC = {6, 6, 6};
```

Anropen

```
increaseAverage(groupA, groupB);
increaseAverage(groupB, groupA);
increaseAverage(groupA, groupC);
```

skall då ge utskrifterna

```
Move 3 from second group to first group!
Move 3 from first group to second group!
Impossible to increase average in both groups!
```

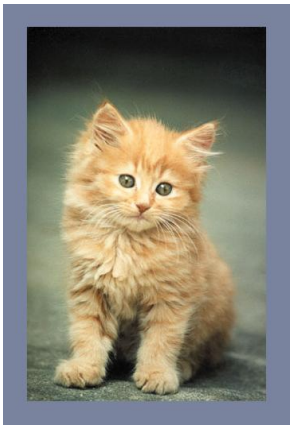
Tips: Det kan vara lämpligt att införa ett par hjälpmetoder. (8 poäng)

b) En digital bild kan representeras som ett tvådimensionellt fält av bildpunkter. I en digital färgbild utgörs varje bildpunkt av *tre* heltalsvärden i intervallet 0-255, där de enskilda värdena representerar intensiteten av färgerna rött, grönt och blått. En färgbild kan avbildas med ett tredimensionellt fält av typen **int**[][][], är den första dimensionen definierar bildens höjd, den andra dimensionen definierar bildens bredd och den tredje dimensionen representerar färgerna rött, grönt och blått.

Din uppgift är att skriva en metod

public static int[][][] withFrame(**int**[][][] samples, **int**[] color)

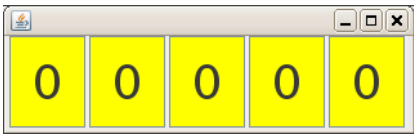
som tar en bild samples och returnerar en ny bild som är en kopia av samples men är omsluten av en ram som har färgen color (enligt figurerna nedan). Tjockleken på ramen skall vara 10% av den ursprungliga bildens bredd.



(8 poäng)

Uppgift 11.

Skriv ett program som visar upp ett fönster, enligt figuren bredvid, för att avbilda ett kodlås som består av 5 komponenter, där varje komponent visar en siffra i intervallet [0, 9].



Siffran i varje komponent ändras genom att trycka på komponenten och när samtliga siffror är rätt inställda öppnas låset.

I ditt program skall du använda klassen NumberButton nedan, för att avbilda komponenterna i kodlåset.

```
import javax.swing.*;
import java.awt.*;
public class NumberButton  extends JButton {
   private int number;
   public NumberButton(int number) {
       if (number < 0 || number > 9)
           throw new IllegalArgumentException();
       this.number = number;
   }//constructor
   public void paintComponent(Graphics pen) {
       super.paintComponent(pen);
       setText("" + number);
   }//paintComponent
   public void add() {
       number = (number + 1) % 10;
       repaint();
   }//add
   public int getNumber() {
       return number;
   }//getNumber
}// NumberButton
```

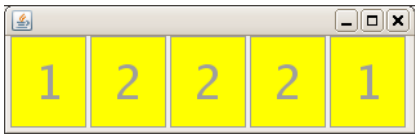
Klassen NumberButton utökar (**extends**) JButton, därför är en instans av klassen NumberButton tryckbar (och har tillgång till alla metoder som finns i klassen JButton). I klassen NumberButton finns en instansvariabel number, som håller reda på vilken siffra som skall visar på knappen. När en instans av klassen skapas ges knappen ett värde via parametern till konstruktorn. Om parametern inte är ett heltal i intervallet [0, 9] kastas ett IllegalArgumentException. För att ändra siffran finns instansmetoden add(), som ökar instansvariabeln number med 1 och ritar om komponenten. Om den aktuella siffran är 9 när metoden add() anropas kommer den nya siffran att bli 0. Klassen innehåller också instansmetoden getNumber() för att avläsa den aktuella siffran på knappen.

Implementera själva fönstret i en klass med namnet CodeLock. Denna klass skall ha en konstruktor

```
CodeLock(int theCode)
```

som har ett heltal som parameter, vilket utgör den *korrekta koden* för kodlåset. Om parametern är ett negativt tal eller ett tal större än 99999 skall ett undantag av typen IllegalArgumentException kastas (eftersom detta tal inte kan vara en giltig kod för kodlåset). Vilken inställning på siffrorna som visas upp initialt när en instans av klassen CodeLock skapas bestämmer du själv. Kan exempelvis, som i figuren ovan, sättas till 00000 eller till en kombination som beräknas slumpmässigt (med en slumptalsgenerator).

När den korrekta koden ställts in, och kodlåset öppnas, skall detta illustreras genom att göra sifferkomponenterna otryckbara.



För full poäng på uppgiften skall programmet naturligtvis skapa och visa upp ett fönster med kodlåset. Vidare krävs att gränssnittet har lämpliga färger, teckenstorlekar och fonter, samt att programmet skall kunna avslutas genom att trycka på stängningsrutan i fönsters ram.

(12 poäng)