

Tentamen för TDA143 PROGRAMMERADE SYSTEM

DAG: 15-03-17

TID: 14:00 – 19:00

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:	<table><tr><td>3:a</td><td>10 poäng på del 1 och 26 poäng på del 2</td></tr><tr><td>4:a</td><td>13 poäng på del 1 och 37 poäng på del 2</td></tr><tr><td>5:a</td><td>16 poäng på del 1 och 48 poäng på del 2</td></tr><tr><td>maxpoäng</td><td>20 poäng på del 1 och 60 poäng på del 2</td></tr></table>	3:a	10 poäng på del 1 och 26 poäng på del 2	4:a	13 poäng på del 1 och 37 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
3:a	10 poäng på del 1 och 26 poäng på del 2								
4:a	13 poäng på del 1 och 37 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:	Torsdag 9/4 kl 10-12 och fredag 10/4 kl 12-14, rum 6128 i EDIT-huset.								
Hjälpmedel:	En valfri lärobok i Java eller de på kursen utdelade föreläsningssanteckningarna (OH-bilderna) som behandlar programmeringsdelen av kursen. Förtydligande noteringar får finnas i boken resp. föreläsningssanteckningarna.								
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.

- 1) Om heltal lagras på två-komplementsform i 8 bits gäller att $10101010_2 + 01010101_2 = -1_{10}$.
- 2) I ett RAM-minne varierar åtkomsttiden beroende på vilken del av minnet som adresseras.
- 3) En skillnad mellan UDP och TCP, som båda är Internetprotokoll på transportlagret, är att protokollet UDP använder sig av en fast kommunikationsförbindelse.
- 4) Om du får ett email där det står att ditt datorkonto stängs av om du inte inom 24 timmar besvarar emaillet med lösenordet till ditt datorkonto, har du utsatts för *phishing*.
- 5) Handelsresande problemet (the traveling salesman problem) är ett s.k. NP-komplett problem, vilket betyder att inte finns några effektiva algoritmer för att hitta exakta lösningar till dem.
- 6) En algoritm som har tidskomplexiteten $\Theta(2^{\log n})$ är alltid snabbare än en algoritm som har tidskomplexiteten $\Theta(n^2)$.
- 7) MySQL är en dialekt av SQL.
- 8) Vattenfallsmodellen är en rigid utvecklingsprocess som lämpar sig dåligt för mjukvaruutveckling.
- 9) *Beta testning* är en form av *glass-box testning*.
- 10) Vid 3D-rendering byggs objekten upp av hexagoner.

(10 poäng)

Uppgift 2.

Vilken trend inom Software Engineering representeras av *The Two-Pizza Rule*? Förklara också betydelsen av denna trend.

(2 poäng)

Uppgift 3.

Många människor använder termerna Internet och world-wide web (WWW) som synonyma begrepp. Detta är dock inte riktigt. Vad är skillnaden mellan begreppen?

(2 poäng)

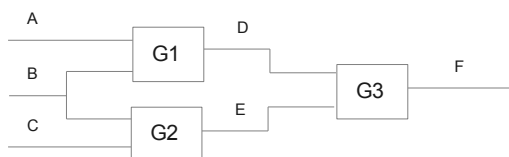
Uppgift 4.

Förklara vad ett binärt sökträd är och beskriv dess egenskaper. Ge exempel på hur det används för sökning och insättning.

(2 poäng)

Uppgift 5.

Nedan finns en krets angiven. Vilken typ av grind (AND, OR, NAND, NOR eller XOR) skall in på platserna G1, G2 respektive G3 för att kretsen skall överensstämma med angiven sanningstabell?



A	B	C	D	E	F
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	0	1

(2 poäng)

Uppgift 6.

Antag att du har följande databastabeller:

Orders:

Ordernumber	CustomerNum	Price	Product
37	102	1034	nails
43	93	2876	tiles
20	54	1999	cement
53	106	12345	bricks

Customers:

Number	Name	Address
54	Johan Svensson	Storgatan 12
106	Lisa Andersson	Drottninggatan 15
102	Anne Olsson	Lillgata 4
93	Erik Hansson	Kungsgata 23
99	Kalle Karlsson	Medelvägen 123

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

```
SELECT Name, Price  
FROM Orders, Customers  
WHERE Price > 2000 AND CustomerNum = Number
```

(2 poäng)

DEL 2: Programmeringsteknik

Uppgift 7.

a) Betrakta nedanstående program:

```
public class Uppgift7a {  
    public static void main(String[] args) {  
        int[] vekt = {5, 8, 3, 9, 4};  
        mystery(vekt);  
        System.out.println(java.util.Arrays.toString(vekt));  
    } //main  
  
    public static void mystery(int[] arr) {  
        for (int i = 0; i < arr.length - 1; i = i + 1) {  
            arr[i] = arr[i] + arr[i + 1];  
            arr[i + 1] = arr[i] - arr[i + 1];  
            arr[i] = arr[i] - arr[i + 1];  
        }  
    } //mystery  
} //Uppgift7a
```

- i) Vad blir utskriften när programmet exekveras? (2 poäng)
- ii) Förklara med ord (en mening) vad metoden `mystery` gör (= vilket syfte metoden har). (2 poäng)
- iii) Skriv om implementationen av metoden `mystery` så att koden blir enklare att förstå. (2 poäng)

b) Vad blir utskriften när nedanstående `main`-metod exekvers

```
import java.util.*;  
public class Uppgift7c {  
    public static void main(String[] args) {  
        ArrayList<Integer> list = new ArrayList<Integer>();  
        for (int i = 0; i < 10; i++) {  
            if (i % 3 != 0)  
                list.add(new Integer(i));  
        }  
        System.out.println(list);  
    }  
} //Uppgift7c
```

(2 poäng)

c) Filmtokiga Stina håller på och skriver ett Javaprogram för att kunna lagra information om de filmer hon sett. Hon har betygssatt filmerna med en femgradig skala som heltal mellan 0 och 4. Nu vill hon dock, av någon anledning, även kunna uttrycka sin betygssättning med omdömena "Urdålig", "Dålig.", "Godkänd.", "Bra." och "Jättebra.". Hon har därför skrivit nedanstående metod:

```
public static String omvandla(int betyg) {  
    if (betyg == 0)  
        return "Urdålig!";  
    else if (betyg == 1)  
        return "Dålig!";  
    else if (betyg == 2)  
        return "Godkänd!";  
    else if (betyg == 3)  
        return "Bra!";  
    else if (betyg == 4)  
        return "Jättebra!";  
} //omvandla
```

Det finns emellertid ett problem eftersom hon vid kompilering av metoden erhåller följande kompileringsfel:

missing return statement

Förklara varför felet uppkommer och skriv om metoden så att den blir korrekt.

(2 poäng)

Uppgift 8

En andragradsekvation har formen $ax^2 + bx + c = 0$, där $a \neq 0$. Lösningsformeln för en andragradsekvation är

$$x = \frac{-b}{2a} \pm \sqrt{\frac{b^2}{(2a)^2} - \frac{c}{a}}$$

Antalet rötter som en andragradsekvation har beror på ekvationens *diskriminant*, D , vilken är uttrycket under kvadratrotsymbolen:

$$D = \frac{b^2}{(2a)^2} - \frac{c}{a}$$

Om $D = 0$ så har ekvationen endast en rot: $x = \frac{-b}{2a}$

Om $D > 0$ så har ekvationen två reella rötter: $x_1 = \frac{-b}{2a} + \sqrt{\frac{b^2}{(2a)^2} - \frac{c}{a}}$ och $x_2 = \frac{-b}{2a} - \sqrt{\frac{b^2}{(2a)^2} - \frac{c}{a}}$

Om $D < 0$ så har ekvationen två imaginära rötter: $x_1 = \frac{-b}{2a} + i\sqrt{\left|\frac{b^2}{(2a)^2} - \frac{c}{a}\right|}$ och $x_2 = \frac{-b}{2a} - i\sqrt{\left|\frac{b^2}{(2a)^2} - \frac{c}{a}\right|}$

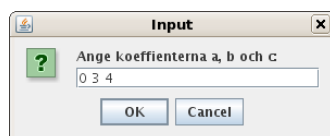
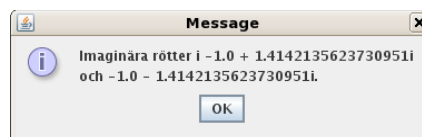
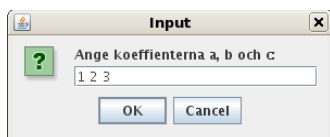
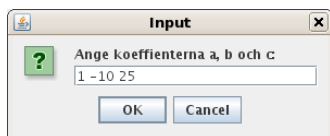
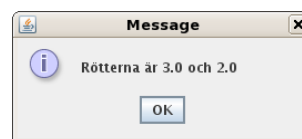
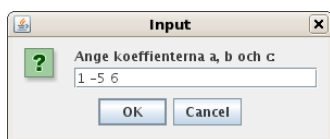
Skriv ett program som använder dialogrutor (enligt figuren nedan) för att upprepade gånger läsa in koefficienterna till en andragradsekvation och skriva ut lösningen till ekvationen. Exekveringen av programmet avbryts genom att användaren tryckar på Cancel-knappen.

För att få full poäng på uppgiften:

- skall koefficienterna med en inläsningssats (dvs ett **Scanner**-objekt skall användas)
- skall programmet ge en felutskrift om värdet 0 ges till koefficienten a.
- skall programmet innehålla en metod

public static void computeRoots(double a, double b, double c)

som beräknar och skriver ut rötterna till andragradsekvationen $ax^2 + bx + c = 0$.



(10 poäng)

Uppgift 9.

Sommelier Winnie Cooler, behöver ett dataprogram för att hålla reda på buteljerna som hon har i sin välfyllda vinkällare. Din uppgift är att hjälpa henne att skapa en klass **Wine**. För varje vin som Winnie har i lager vill hon hålla reda på följande uppgifter:

namn	namnet på vinet
typ	anger om vinet är vitt, rött eller rose
årtal	vinets årgång
kommentar	anger kort beskrivning av vinet
antal	antal flaskor i lager

För att kunna ange vilken typ ett vin har, skall klassen ha tre *publika klasskonstanter* **WHITE**, **RED** och **ROSE** av typen **int**.

Klassen skall innehålla *en* konstruktörer som initierar samtliga instansvariabler, instansmetoder för att avläsa samtliga tillstånd, instansmetoder för att ändra antalet och kommentaren, samt en metod **toString** för att enkelt kunna få en utskrift på formen:

Namn: Hectare Feteasca Neagra
Typ: Rött
Årtal: 1997
Kommentar: Elegant, moget och nyanserat. Har toner av vanilj, tobak och choklad.
Antal i lager: 12

(10 poäng)

Uppgift 10.

a) Skriv en metod

```
public static ArrayList<Wine> allRoseFromYear(ArrayList<Wine> bottles, int year)
```

som tar en ArrayList **bottles** som innehåller objekt av klassen **Wine** (se uppgift 3 ovan) och ett heltal **year** som anger ett årtal. Metoden skall returnera en ny ArrayList som innehåller de objekt i **bottles** som är roseviner från året **year**.

(6 poäng)

b) Ett palindrom är ett ord eller fras som har samma utseende oavsett om man läser den framifrån eller bakifrån, t.ex "madam", "Nej, leta i ateljen!" och "Was it a car or a cat I saw?".

Skriv en metod

```
public static boolean isPalindrom(String phrase)
```

som tar en sträng **phrase** och returnerar värdet **true** om **phrase** är ett palindrom, annars returneras värdet **false**.

Som framgår av exemplen ovan, är det endast bokstäverna som ingår i en fras som avgör om frasen är ett palindrom eller inte, och ingen skillnad görs mellan versaler och gemener.

Tips 1:

Skriv en hjälpmetod:

```
private static String removeNonLetters(String str)    som returnerar sträng som är en kopia av str  
                                                       förutom att alla icke-bokstäver är borttagna
```

Tips 2:

Vad finns det för några användbara metoder i klasserna **String** och **Character**?

(6 poäng)

- c) *Vinjettering* är ett optiskt fenomen som är karaktäristiskt för gamla fotografier. Fenomenet orsakas av att kameraobjektivet inte klarar av att belysa hela fotot jämnt - utan ljusstyrkan avtar ut mot kanterna – vilket innebär att fotografiet blir mörkare i kanterna än i mitten.

Din uppgift är att skriva en metod

```
public static int[][][] vignetting(int[][][] sample) {
```

som tar en digital färgbild **samples** och returnerar en ny digital färgbild som är en *vinjetterad* kopia av bilden **samples**.

En digital färgbild representeras som ett tvådimensionellt fält av bildpunkter, där varje bildpunkt utgörs av *tre* heltalsvärden i intervallet 0-255. De enskilda värdena i en bildpunkt representerar intensiteten av färgerna rött, grönt och blått.

Vinjetteringen åstadkoms genom att multiplicera komponenterna i varje bildpunkt i den ursprungliga bilden med värdet:

$$1 - (\text{distance to center})^2 / (\text{half of image diagonal})^2$$



Original bild

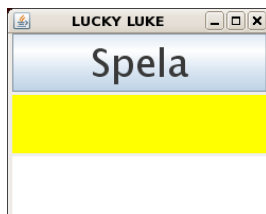


Vinjetterad bild

(7 poäng)

Uppgift 11.

Skriv ett program som fungerar som en enkel form av en spelmaskin av typen "enarmad-bandit". Användargränssnittet skall vara enligt figurerna nedan:



Fönstret skall ha titeln LUCKY LUKE och består av tre huvuddelar. Överst finns en knapp med texten **Spela**. Därunder en panel med plats för tre **JLabel**-komponenter, vilka under spelets gång kommer att innehålla siffror mellan 0 och 9, men som från början är blanka. Underst finns en **JLabel**-komponent som innehåller information om hur en spelomgång gick.

När man klickar på knappen **Spela** genomförs en spelomgång, som består i att tre heltal mellan 0 och 9 slumpas fram och visas i de tre **JLabel**-komponenterna i mittersta fältet i fönstret. Om de tre talen är olika visas texten **Ingen vinst** i det nedersta fält i fönstret, om två av talen är lika visas texten **Vinst 2 kr** och om samtliga tre tal är lika visas texten **Vinst 50 kr**.

För att erhålla full poäng på uppgiften skall gränssnittet ha lämpliga färger, teckenstorlekar och fonter, samt att programmet skall kunna avslutas genom att trycka på stängningsrutan i fönstrets ram.

(11 poäng)