

Tentamen för TDA540

Objektorienterad Programmering

Institutionen för Datavetenskap
CTH HT-15, TDA540

Dag: 2016-01-16, *Tid:* 14.00-18.00

Ansvarig:	Alex Gerdes
Examinator:	Joachim von Hacht och Christer Carlsson
Förfrågningar:	Alex Gerdes (alexg@chalmers.se, 0705677747)
Resultat:	Erhålls via Ladok
	3:a 24 poäng
	4:a 36 poäng
Betygsgränser:	5:a 48 poäng
	max 60 poäng
Siffror inom parentes:	Anger maximal poäng på uppgiften
Granskning:	Tentamen kan granskas på den 5:e februari 2016 (12:00 – 13:00) i EDIT 6128. Vi eventuella åsikter om rättningen ange noggrant vad du anser är fel.
Hjälpmedel:	Cay Horstmann: <i>Java for everyone</i> eller Jan Skansholm: <i>Java direkt med Swing</i> Understrykningar och smärre förtydligande noteringar får finnas.
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. Indentera programkoden! Vid rättning av uppgifter där programkod ingår bedöms principiella fel allvarigare än smärre språkfel.

Lycka till!

Uppgift 1

(2 poäng)

Betrakta metoden from nedan:

```
public static void from (int base, int value) {
    if (value > 0) {
        from(base, value / base);
        System.out.printf("%d", value % base);
    }
}
```

Vad skriver metoden from ut om base är 2 och value är 42? Förklara vad metoden gör.

Tips: vad skriver metoden ut med base 10?

Uppgift 2

(2 poäng)

Vilken utskrift fås då nedanstående program körs?

```
public class AClass {
    private int x, y;
    private static int z;

    public AClass(int a, int b, int c) {
        x = a;
        y = b;
        z = c;
    }

    @Override
    public String toString() {
        return x + ", " + y + ", " + z;
    }
}

public class TestClass {
    public static void main(String[] args) {
        AClass a = new AClass(1, 2, 3);
        AClass b = new AClass(4, 5, 6);
        System.out.println(a);
        System.out.println(b);
    }
}
```

Uppgift 3

(1 poäng)

Nedanstående kodavsnitt ger ett kompileringsfel. Förklara vad som är fel och rätta till.

```
do {
    boolean b = Math.random() < 0.5;
} while (b);
```

Uppgift 4

(8 poäng)

För att avgöra om ett n -siffrigt tal N är ett Keithtal skapar man en Fibonacci-liknande talföljd som börjar med de n siffrorna i N med den mest signifikanta siffran först. Man fortsätter sedan talföljden med termer som var och en är summan av de n föregående

termerna. N är ett Keithtal om N ingår i den på detta sätt konstruerade talföljden och är större än 9.

Betrakta exempelvis ett tresiffrigt tal $N = 197$. Talföljden blir då:

1, 9, 7, 17, 33, 57, 107, 197, 361, ...

Eftersom 197 ingår i talföljden så är det ett Keithtal. Om $N = 123$ då blir talföljden:

1, 2, 3, 6, 11, 20, 37, 68, 125, ...

och kan vi dra slutsatsen att 123 är inget Keithtal, för 123 är inte med i talföljden.

Skriv en metod som givet ett heltal n returnerar om talet är ett Keithtal eller inte.

Uppgift 5

(5 poäng)

Din uppgift är att skriva ett program som upprepade gånger läser in ett heltal samt beräknar om talet är ett Keithtal eller inte. Du får själv välja om du vill göra in- och utmatning via dialogrutor eller använda `System.in` respektive `System.out`.

För att 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 eller ctrl-c),
- skall talet läsas i en inläsningssats (dvs ett Scanner-objekt skall användas)
- skall programmet ge felutskrift om ett ogiltigt indata ges (dvs värden mindre än 10)

Uppgift 6

(6 poäng)

Skriv en metod

```
public static int[] getIndexOfMinValues(int[] vect)
```

som tar ett sorterat heltalsfält `vect` och returnerar ett nytt fält som innehåller index för samtliga förekomster av det minsta värdet i `vect`.

Exempel

Antag att följande deklaration har gjorts

```
int[] arr = {12, 3, 50, 24, 3, 51, 32, 50, 16};
```

anropet `getIndexOfMinValues(arr)` returnerar då fältet `{1, 4}`, eftersom 3 är det minsta värdet och detta värde finns i fältet på index 1 och 4. *Tips: Det kan vara lämpligt att införa ett par hjälpmetoder.*

Uppgift 7

(8 poäng)

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[][][]`. 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[][][] blur(int[][][] samples, int n)
```

som tar en bild `samples` och returnerar en ny bild med effekten att bilden blir rörelseoskärpa. Detta görs genom att beräkna färgerna i punkten (x, y) i den nya bilden med hjälp av ett så kallade faltningsfilter. Vid användning av faltningsfiltret beräknas ett nytt värde på en bildpunkt genom att, förutom att beakta bildpunkten själv, även beakta närliggande bildpunkter. Ett faltningsfilter kan således anses som en matris. Faltningsfiltret för att skapa en rörelseoskärpa-effekt är en $n \times n$ diagonal matris med diagonala faktorna $\frac{1}{n}$. Till exempel en matris med storlek 5:

$$\begin{bmatrix} \frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{5} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} \end{bmatrix}$$

Filtret anger att det nya värdet i en bildpunkt beräknas genom att multiplicera bildpunktens gamla värde med faktorn i mitten av matrisen (dvs med $\frac{1}{5}$) och addera, beroende på faktorna, några närliggande bildpunkter. Metoden `blur` tar matrisens storlek (`n`) som en argument. *Tips: återigen är det lämpligt att införa hjälpmetoder. Till exempel en metod för att skapa ett filter och en metod för att applicera filtret.*

Uppgift 8

(6 poäng)

Skriv en metod som givet en sträng `s` och ett heltal `n` genererar en ny sträng med längden `n`, bestående av slumpvis valda tecken ur `s`. Om `s` är den tomma strängen sätts resultatet till den tomma strängen. Du får använda `String`-metoderna som anges vid nästa frågan.

Uppgift 9

(8 poäng)

Vi vill översätta svenska meningar till "I-språket". Detta sker genom att alla vokaler ersätts med ett 'i'. Till exempel:

```
"Hur är läget?" => "Hir ir ligit?"
```

Skapa en klass `ILanguage` med följande metoder:

1. `private boolean isVowel(char ch)` ger sant om `ch` är en vokal annars falskt.
2. `public String toI(String swe)` givet en sträng på svenska returneras översättningen till I-språket. Använd metoden från 1 i din lösning.

För att lösa uppgifterna är det tillåtet att använda följande metoder från klassen `String`:

- `char charAt(int i)` ger tecknet vid index `i`
- `int indexOf(char ch)` ger index för tecknet `ch` eller `-1` om tecknet saknas
- `int length` ger längden av strängen
- `String substring(int start, int end)` ger en delsträng från `start` till `end - 1`
- `String substring(int start)` ger en delsträng från `start` till strängens slut
- `String[] split(String str)` delar upp en sträng i ett fält av delsträngar utifrån ett visst tecken, till exempel:
`"aaa:bb:cccc:dd".split(":") -> {"aaa", "bb", "cccc", "dd"}`

Uppgift 10

(6 poäng)

Om man har två grupper av tal kan man ibland öka medelvärdet i båda grupperna genom att flytta ett tal från ena gruppen till den andra. Skriv en metod som givet två talgrupper `A` och `B` (i form av `int[]`) skriver ut vilket (om något) tal som skall flyttas vart.

Exempel

Talgrupper:

```
A = {3, 1, 2};
B = {4, 3, 4, 5};
```

Utskrift:

```
Move 3 from B to A!
```

Vi antar att grupperna innehåller max 10 positiva heltal och att maxvärdet för talen är 100. Om ingen ökning av medelvärdet är möjligt skrivs "Impossible" ut.

Uppgift 11

(8 poäng)

Finns i sjön är ett enkelt kortspel för två eller fler deltagare. Spelet går ut på att samla på fyrtal, det vill säga grupper om fyra kort med samma valör. Deltagarna får sju kort var, och resterande kort sprids ut på bordet med baksidan uppåt och utgör "sjön".

Den spelare som är i tur frågar en valfri motspelare efter kort av en viss valör, och motspelaren måste då lämna ifrån sig alla sina kort av den begärda valören. Saknas sådana kort, svarar motspelaren med frasen "finns i sjön", varpå den som frågat får ta upp ett kort från sjön; därefter går turen vidare. Vinnare är den spelare som vid spelets slut har samlat ihop flest fyrtal.

Skapa en objektmodell för kortspelet:

1. Ange 4 lämpliga klasser
2. Hitta något lämpligt attribut (instansvariabel) för varje klass
3. Skapa konstruktörer för varje klass och visa hur du med hjälp av dessa kan koppla ihop hela modellen. Skapa en metod `buildGame` som utför arbetet och returnerar hela modellen