

Omtentamen för TDA540

Objektorienterad Programmering

Institutionen för Datavetenskap
CTH HT-15, TDA540

Dag: 2016-04-09, *Tid:* 14.00-18.00

Ansvarig:	Alex Gerdes
Examinator:	Joachim von Hacht och Christer Carlsson
Förfrågningar:	Alex Gerdes (alexg@chalmers.se, 0317726154)
Resultat:	Erhålls via Ladok
	3:a 24 poäng
Betygsgränser:	4:a 36 poäng
	5:a 48 poäng
	max 60 poäng
Siffror inom parentes:	Anger maximal poäng på uppgiften
Granskning:	Tentamen kan granskas på den 25:e april 2016 (12:00 – 13:00) och 2:e maj 2016 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

(3 poäng)

Betrakta metoderna nedan:

```
public static void run() {
    int[] xs = {5, 8, 3, 9, 4};
    mystery(xs);
    System.out.println(java.util.Arrays.toString(xs));
}

public static void mystery(int[] xs) {
    int l = xs.length;
    for (int i = 0; i < l/2; i++) {
        int t = xs[i];
        xs[i] = xs[l - i - 1];
        xs[l - i - 1] = t;
    }
}
```

Vad blir utskriften när metoden run exekveras? Förklara vad metoden mystery gör.

Uppgift 2

(4 poäng)

Vilken utskrift fås då nedanstående program körs?

```
public class P {
    protected String name;
    protected int number;

    public P(String name, int n) {
        this.name = name;
        number = n;
    }

    @Override public String toString() {
        return name + " / " + number;
    }
}

public class SV extends P {
    private int tel;

    public SV(String name, int n, int m) {
        super(name, m);
        tel = n;
    }

    @Override public String toString() {
        return name + " / " + number + " / " + tel;
    }
}

public class MainClass {
    public static void main(String[] args) {
        P a = new SV("Johan", 7, 13);
        SV b = new SV("Anna", 5, 11);
        System.out.println(a);
        System.out.println(b);
    }
}
```

Uppgift 3

(3 poäng)

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

```
public static String storlek(int n) {
    if (n < 0)
        return "Negative";
    else if (n < 100)
        return "Small";
    else if (n < 1000)
        return "Large";
}
```

Uppgift 4

(6 poäng)

En paritetsbit är en binär siffra som anger huruvida antalet bitar med värdet 1 i en given mängd bitar är jämnt (eller udda). Att lägga in paritetsbitar är ett enkelt sätt för feldetektering när man till exempel överför data. Om paritetsbiten inte stämmer med siffergruppen, då vet man att data är felaktigt. Till exempel:

7 databitar	byte med jämn paritetsbit
0000000	00000000
1010001	11010001
1101001	01101001
1111111	11111111

Skriv en metod som givet ett positivt heltal n med $n \leq 127$ returnerar ett byte (ett heltalsfält med ettor och nollor) med en jämn paritetsbit som finns på index 0. Till exempel, ett metदानrop med talet 42 ger {1,0,1,0,1,0,1,0} som svar. Binär representation av 42 i 7 databitar är 0101010 och antalet ettor är udda och därför lägger man till en 1 på index 0: 10101010. Antar att det finns en metod `int[] bin(int n, int d)` som översätter ett heltal n till binär med d databitar.

Uppgift 5

(6 poäng)

Din uppgift är att skriva ett program som upprepade gånger läser in ett binär tal, som representeras med en sträng ettor och nollor, och skriver ut om paritetsbiten stämmer med resten av siffergruppen. 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 (till exempel ett värde som är inte binär)

Uppgift 6

(7 poäng)

Skriv en metod

```
public static List<Integer> difference(List<Integer> xs, List<Integer> ys)
```

som tar två osorterade listor och returnerar en ny lista som tar bort den första förekomsten (om den finns) av varje element i `ys` från `xs`.

Exempel

Antag att följande deklaration har gjorts

```
public static List<Integer> toList(int[] xs) {
    List<Integer> list = new ArrayList<>();

    for (int x : xs) list.add(x);

    return list;
}

int[] xs = {1, 6, 2, 1, 4, 1, 2, 1, 8};
int[] ys = {1, 2, 1, 4};
```

anropet `difference(toList(xs), toList(ys))` returnerar en lista med följande element: `[6, 1, 2, 1, 8]`. Obs att två förekomster av 1 har tagits bort och att fortfarande en förekomst av 2 finns kvar.

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[][][] scale(int[][][] samples, int n)
```

som tar en bild `samples` och returnerar en ny bild som är `n` gånger större i varje dimension. Varje pixel blir alltså ersätt med $n \times n$ pixels av samma färg. Till exempel en bild med storlek 3 som blir 2 gånger större:

$$\begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} \xrightarrow{2\times} \begin{bmatrix} p_{11} & p_{11} & p_{21} & p_{21} & p_{31} & p_{31} \\ p_{11} & p_{11} & p_{21} & p_{21} & p_{31} & p_{31} \\ p_{12} & p_{12} & p_{22} & p_{22} & p_{32} & p_{32} \\ p_{12} & p_{12} & p_{22} & p_{22} & p_{32} & p_{32} \\ p_{13} & p_{13} & p_{23} & p_{23} & p_{33} & p_{33} \\ p_{13} & p_{13} & p_{23} & p_{23} & p_{33} & p_{33} \end{bmatrix}$$

Uppgift 8

(8 poäng)

I vissa fall måste man översätta en URL, till exempel när man använder reserverade tecken i en HTTP-request. Att översätta en URL sker genom att ersätta alla reserverade tecken med ett 'escape'-tecken som börjar med en procent tecken, till exempel:

```
"http://alex.nl/age?input=25" => "http%3A%2F%2Falex.nl%2Fage%3Finput%3D25"
```

Skapa en klass `Encode` med följande metoder:

1. `private static String escape(char ch)` som returnerar en sträng enligt nedanstående tabell:

<code>'.'</code>	<code>⇒</code>	<code>"\%3A"</code>
<code>'/'</code>	<code>⇒</code>	<code>"\%2F"</code>
<code>'?'</code>	<code>⇒</code>	<code>"\%3F"</code>
<code>'='</code>	<code>⇒</code>	<code>"\%3D"</code>

om tecknet inte finns med då returnerar metoden en sträng med enbart `ch`.

2. `public static String encodeURL(String url)` som översätter en URL så att alla reserverade tecken är översatt. Använd metoden från 1 i din lösning.

För att lösa uppgifterna är det tillåtet att använda metoder från klassen `String`, till exempel:

- `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 9

(7 poäng)

Skriv en metod `copy` tar en sträng och ett heltal `n` som argument och returnerar en sträng med `n` kopior av strängen. Använd `copy` metoden för att implementera en metod `julgran` som skriver ut följande om man anropar metoden så här `julgran(6)`:

```
.....*.....
....*-*....
...*-**...
..*-***..
.*-****.
*-*****
*-*-*-*-*
```

Parametern till metoden `julgran` bestämmer antal rader text. I ovanstående exempel är det 6, om man använder ett större tal då får man fler som blir också längre. Det skapas en triangel med stjärnor med streck emellan och inneslutet av punkter. Med lite fantasi liknar utskriften en julgran. Om parametern är 1 då skriver metoden bara ut en enstaka stjärna: `*`. Om parametern är mindre eller lika med 0 då skriver metoden ingenting ut.

Uppgift 10

(8 poäng)

Vi vill samla all din musik i ett musikbibliotek. Istället för att rota bland högar med CD-skivor, eller söka igenom digitala filer, vill vi använda ett program för att kolla om en låt, artist, etc. finns med i biblioteket. Musikbiblioteket borde innehålla följande information: album, artist, genre och låt. Skapa en objektmodell för musikbiblioteket:

1. Ange 4 lämpliga klasser
2. Hitta något (eller några) 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.