

PROGRAMMERINGSTEKNIK, del A
TIN213

OBS! Det kan finnas kurser med samma eller liknande namn på olika utbildningslinjer. Denna tentamen gäller *endast* för den eller de utbildningslinjer som anges ovan. Kontrollera därför noga att denna tentamen gäller för den utbildningslinje du själv går på.

TID: 08:30 - 11:30 SAL: SB-huset

Ansvarig: Jan Skansholm, tel 0707 163230

Betygsgränser: Sammanlagt maximalt 30 poäng.
På tentamen ges graderade betyg:
3:a 15 poäng, 4:a 20 poäng, 5:a 25 poäng

Hjälpmedel: Skansholm, *Java direkt med Swing*, Studentlitteratur, eller
Bravaco, Simonson, *Java Programming: From the Ground up*
(Understrykningar och mindre anteckningar i boken är tillåtna.)

Inga kalkylatorer är tillåtna.

Tänk på:

- att skriva tydligt och disponera papperet på ett lämpligt sätt.
- att börja varje ny uppgift på nytt blad. Skriv endast på en sida av papperet
- Skriv den (anonyma) kod du fått av tentamensvakten på *alla* blad.

De råd och anvisningar som givits under kursen skall följas vid programkonstruktionerna. Det innebär bl.a. att onödigt komplicerade, långa och/eller ostrukturerade lösningar i värsta fall ej bedöms.

Uppgift 1) a) Vilket påstående är korrekt om `MemberList` på basis av vad som visas nedan?
För poäng krävs att du motiverar ditt svar!

```
import java.util.*;
public class MemberList {
    private LinkedList<Person> list;
    private String name;
    public MemberList(String name) { this.name = name; }
    public void add(Person p) { list.add(p); }
}
```

- a. Klassen går inte att kompilera.
- b. Det går inte att skapa objekt av klassen.
- c. Exekveringen riskerar att avbrytas om objekt av klassen används.
- d. Klassen är perfekt.

(2 p)

b) I ett program har man lagt följande rader:

```
String s1 = "HEJ";
String s2 = s1;
String s3 = "hej".toUpperCase();
if (s1==s2)
    System.out.println("lika");
else
    System.out.println("olika");
if (s1==s3)
    System.out.println("lika");
else
    System.out.println("olika");
```

Ange vilken utskrift programmet bör ge och *förklara varför!*

(1 p)

c) Konstruera en klass `Lag` som beskriver t.ex. fotbollslag som deltar i en serie. Klassen skall ha utifrån oåtkomliga instansvariabler för klubbnamnet, antal spelade matcher, totala antalet gjorda mål, totala antalet insläppta mål och antal poäng. Vinst ger 3 poäng, oavgjort 1 poäng och förlust 0 poäng. Klassen skall ha en konstruktor med vilken man kan initiera ett lag med ett visst namn. Dessutom skall det finnas en metod med namnet `nyMatch` som man anropar för att registrera ett matchresultat. Metoden skall ha två parametrar: antal mål som laget gjort i matchen respektive antalet mål som man släppt in. Slutligen skall det finnas en metod `toString` som ger en text som motsvarar en rad i en tabell (namn, antal spelade matcher, antal gjorda mål, antal insläppta mål samt antal poäng), t.ex.

```
"GAIS      3      3      2      4"
```

(6 poäng)

Uppgift 2) Två positiva heltal i och j kallas för *relativa primtal* om det inte finns något heltal större än 1 som de båda är delbara med. (Talen 16 och 21 är t.ex. relativa primtal, men däremot inte talen 18 och 21 eftersom de båda är delbara med 3.) Skriv en klassmetod `relPrim` som undersöker om två positiva heltal är relativa primtal. Metoden skall som resultat ge ett värde av typen `boolean`.

Skriv sedan ett fullständigt program som gång på gång läser in ett par heltal från en dialogruta (Obs! inte två dialogrutor) och som undersöker om de två talen är relativa primtal eller inte. Programmet *måste* använda sig av funktionen `relPrim`. Resultatet skall meddelas i en annan dialogruta. När användaren klickar på knappen *Avbryt* i den första dialogrutan skall programmet avslutas.

(9 p)

Uppgift 3) I Linux finns ett kommando som heter `wc` (word count). Som argument på kommandoraden skall kommandot ha namnet på en fil. Som resultat skriver programmet ut antalet rader, antalet ord samt antalet tecken i filen. Dessutom skrivs filnamnet ut sist. Uppgiften är att skriva en egen version i Java av programmet `wc`. När man kör programmet kan det se ut på följande sätt:

```
> java Wc nisse.txt
      25      204      1298      nisse.txt
```

Programmet skall kontrollera att exakt ett argument givits på kommandoraden. Om så inte är fallet skall en felutskrift ges. Om den angivna filen inte existerar skall programmet skriva ut en felutskrift på formen: "Cannot open xxx", där xxx är det namn som angavs på kommandoraden. Ditt program får bara läsa igenom filen en gång.

Tänk på att efter varje rad i en textfil finns en radslutsmarkör `\n` som man måste ta hänsyn till när man beräknar antalet tecken. I Linux består en radslutsmarkör av ett tecken.

(12 p)

Lösningar till tentamen, del A 2017-12-16

```
// Uppgift 1a
```

Rätt alternativ är c eftersom instansvariabeln list har värdet null.

```
// Uppgift 1b
```

```
    Utskrift:
    lika
    olika
    s1, s2 och s3 är referenser till String-objekt och det är referenserna
    som jämförs i
    if-satserna, inte det de refererar till. I det första fallet tilldelas
    referensen s1
    till referensen s2. Dessa referenser är således lika. I det andra
    fallet skapas först
    ett nytt String-objekt som refererar till texten "hej". Därefter skapas
    ytterligare ett
    nytt String-objekt som kommer att referera till texten "HEJ".
    Referensen till detta
    objekt tilldelas till s3, som alltså refererar till ett annat String-
    objekt än s1.
```

```
// Uppgift 1c
```

```
    public class Lag {
    private String namn;
    private int antal;
    private int plus;
    private int minus;
    private int poäng;

    public Lag(String namn) {
        this.namn=namn;
    }

    @Override
    public String toString() {
        return namn + "\t" + antal + "\t" + plus + "\t" + minus + "\t" +
poäng;
    }

    public void nyMatch(int gjorda, int insläppta) {
        antal++;
        if (gjorda > insläppta)
            poäng +=3;
        else if (gjorda == insläppta)
            poäng++;
        plus += gjorda;
        minus += insläppta;
    }
    }
}
```

```
// Uppgift 2
```

```
import javax.swing.*;
import java.util.*;
class RelPrimProg {

    public static boolean relPrim(int i, int j) {
```

```

    for (int k=2; k<=i && k<=j; k++)
        if (i%k == 0 && j%k == 0)
            return false;
    return true;
}

public static void main(String arg[]) {
    while(true) {
        String s = JOptionPane.showInputDialog("Ange två heltal");
        if (s == null)
            break;
        Scanner sc = new Scanner(s);
        int a = sc.nextInt(), b = sc.nextInt();
        if (relPrim(a,b))
            JOptionPane.showMessageDialog(null, "Relativa primtal");
        else
            JOptionPane.showMessageDialog(null, "Inte relativa primtal");
    }
}

// Uppgift 3
import java.io.*;
import java.util.*;
public class Wc {
    public static void main(String[] arg) throws IOException {
        if (arg.length != 1) {
            System.out.println("Illegal argument");
            System.exit(1);
        }
        BufferedReader f = null;
        try {
            f = new BufferedReader(new FileReader(arg[0]));
        }
        catch (FileNotFoundException e) {
            System.out.println("Cannot open " + arg[0]);
            System.exit(1);
        }
        int antTeck=0, antOrd=0, antRad=0;
        String rad;
        while ((rad = f.readLine()) != null) {
            antRad++;
            antTeck += rad.length() + 1; // Unix
            Scanner sc = new Scanner(rad);
            while (sc.hasNext()) {
                antOrd++;
                sc.next();
            }
        }
        f.close();
        System.out.println(antRad + " " + antOrd + " " + antTeck + " " +
arg[0]);
    }
}

```