

**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: Maskinsalar

---

Ansvarig:              Robin Adams, tel. 031 772 63 48

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) Konstruera en klass `Nummerlappar` som kan användas för att dela ut nummerlappar, t.ex. i en butik. Klassen skall ha fyra metoder:

- `taLapp`, anropas av en kund, ger nästa könummer, varje dag skall den första kunden som kommer få nummer 1,
- `betjÄnaKund`, anropas av en expedient, ger numret för nästa kund som skall betjÄnas, om kundkÖn är tom skall värdet -1 returneras,
- `betjÄnasNu`, ger numret för den senaste kunden som fick betjÄning,
- `init`, anropas av expedient vid dagens början, återstÄller systemet

(6 p)

b) Klassen `Int` ser ut på följande sÄtt:

```
public class Int {
    private int x;
    public Int() { x = 0; }
    public void set( int x ) { this.x = x; }
    public int get() { return x; }
}
```

Vilken utskrift ger kodavsnittet nedan?

```
List<Int> list = new ArrayList<>();
Int n = new Int();
for ( int j = 0; j < 3; j++ )
    list.add( n );
int k = 1;
for ( Int x : list )
    x.set( k++ );
for ( Int x : list )
    System.out.println( x.get() );
```

(2 p)

Uppgift 2) Ett primtal är ett tal som endast är jämnt delbart med 1 och med sig själv. En teknik att finna primtal går under namnet *Eratosthenes sÄll*. Den går till på följande sÄtt: Man använder ett fÄlt där elementen har typen `boolean`. Från början sÄtts element nummer 0 och 1 till `false` och alla Övriga element i fÄltet till `true`. Därefter löper man igenom fÄltet med början på index 2. När man stÖter på ett element (låt oss sÄga att det har nummer *i*) som har värdet `true` gör man följande. LÖp igenom resten av fÄltet (fr.o.m. *i*+1 till fÄltets slut) och sÄtt varje element som har den egenskapen att dess index är en jämn multipel av *i* till `false`. När man t.ex. kommer till element nr. 5 så skall elementen 10, 15, 20 etc. i fÄltet ges värdet `false`.

När denna process är avslutad visar de element i fÄltet som fortfarande har värdet `true` vilka index som är primtal. Konstruera ett program som använder denna teknik för att beräkna och skriva ut alla primtal som är mindre än eller lika med 1000. Utskriften skall ske i ett kommandofönster.

(10 p)

Uppgift 3) Vissa envisa användare av amerikanskt tangentbord brukade förr i tiden göra livet surt för sina vänner genom att inte skriva vanliga å, ä och ö i sina elektroniska brev. I stället skrev de konstiga bokstavskombinationer. Man kunde spara ett sådant brev i en textfil och sedan köra ett program som översatte texten i filen så att å, ä och ö visades på normalt sätt. Din uppgift är att skriva ett program som läser en textfil och som skriver ut textfilens innehåll i kommandofönstret. Vid utskriften skall alla förekomster av bokstavskombinationerna aa, ae och oe översättas till å, ä respektive ö. (För enkelhets skull får du ändra den inlästa texten så att den bara innehåller små bokstäver innan du gör översättningen.)

Namnet på filen skall ges på *kommandoraden* när programmet startas. Programmet skall kontrollera att det finns exakt ett argument på kommandoraden. Om så inte är fallet skall felutskriften "Felaktigt antal argument" ges och programmet avslutas. Om antalet argument är korrekt skall programmet kontrollera att den angivna filen går att öppna. Om inte så skall en felutskrift med formen "Kan inte öppna filen xxx" ges. xxx skall vara filens namn.

(12 p)