

Examination in

## PROGRAMMERINGSTEKNIK F1 TIN211

DAY: FRIDAY    DATE: 2012-04-13    TIME: 8.30-13.30 (OBS 5 timmar)    ROOM: M

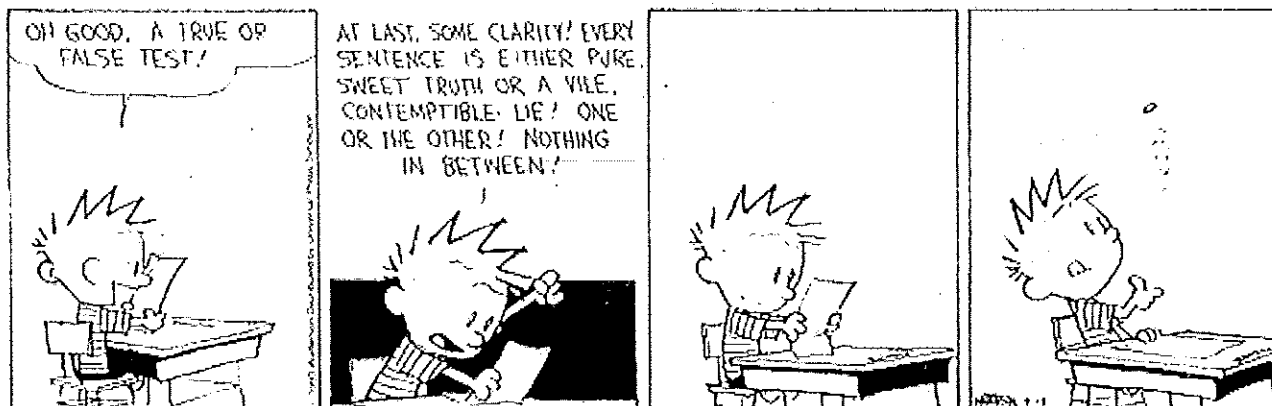
---

Responsible teacher: Erland Holmström tel. 1007, home 0708-710600  
Results: Are sent by mail from Ladok.  
Solutions: Are eventually posted on homepage.  
Inspection of grading: The exam can be found in our study expedition after posting of results.  
Time for complaints about grading are announced on homepage after the result are published or mail me and we find a time.  
Grade limits: CTH: 3=26p, 4=36p, 5= 46p, max 60p  
Aids on the exam: **Bravaco, Simonson Java Programming From the Grounf Up**  
or Horstmann Java Concepts and A print of Chapter 18.  
or Skansholm Java direkt.

### Observe:

- Start by reading through all questions so you can ask questions when I come. I usually will come after appr. 2 hours.
- All answers must be motivated when applicable.
- Write legible! Draw figures. Solutions that are difficult to read are not evaluated!
- Answer concisely and to the point.
- The advice and directions given during course must be followed.
- Programs should be written in Java, indent properly, use comments and so on.
- Start every new problem on a new sheet of paper.

### Good Luck!



Problem 1. Enkla frågor samt Sant eller falskt? Förklara dina svar. (förklaringen är vanligen viktigare än svaret)

a) Vad är svagheten med följande konstruktion?

```
catch ( Exception e ) {...}
catch ( ArithmeticException e ) {...}
```

b) Antag att vi har metoden

```
int aMethod(int x, int y)
```

Vilka av följande överlagringar (overloading) är inte ok?

```
float aMethod(int x, double y) {...}
int aMethod(int x, double y) {...}
float aMethod(int x, int y) {...}
int aMethod(int x, int y, int z) {...}
```

c) När ett fält deklarerats så kan man inte

ändra fältreferensen, ändra ett fältvärde, ändra fältets storlek?

d) En exception är ett objekt.

e) En JFrame är en Frame som är en Container som är en Component.

2+1+1+1+1 (6p)

Problem 2. Testar: strängar, tecken, loopar, slumpstal, listor, metoder, exceptions, ArrayList, inläsning med scanner.

Den sex-ställiga kod som används för bilnummer är naturligtvis unik – ingen annan bil har samma kod som din bil. Vi skall skriva ett litet program som genererar liknande koder men där Hamming avståndet (hd) alltid är 3.

Wikipedia: "In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. Put another way, it measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other.

The Hamming distance between:

"toned" and "roses" is 3, 1011101 and 1001001 is 2, 2173896 and 2233796 is 3."

Koderna skall bestå av tre bokstäver ('A' till 'Ö') följda av tre siffror ('0' till '9'), och för att minska risken att något blir fel skall du se till att alla koder skiljer sig åt i minst *tre* av de sex tecken som ingår i koden (dvs  $hd \geq 3$ ), om man jämför tecken för tecken:

```
G M E 3 3 9
G D E 1 8 9
= ! = ! ! =
```

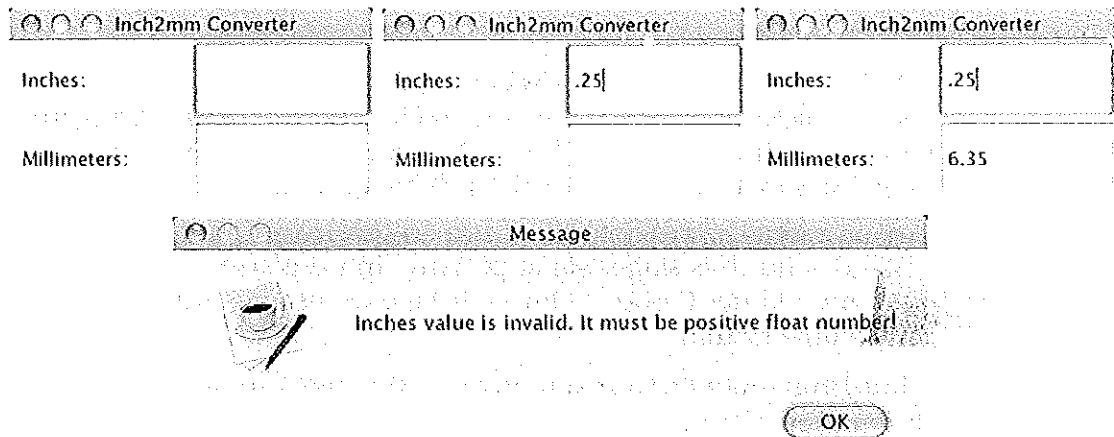
Om någon har koden GME339 så kan det alltså finnas någon med koden GDE189, men det finns ingen annan kod som exempelvis GDE389 eller MME039. Vi bryr oss här inte om omkastningar av tecken, så det kan finnas koder som MEG339 eller GEM933 .

Skriv ett program som genererar slumpmässiga koder enligt reglerna ovan. Om man kör programmet flera gånger skall man få nya koder varje gång (om man inte har en väldig tur/otur – du behöver inte kontrollera de koder vi fått vid tidigare körningar). (Du kan i en deluppgift alltid förutsätta att metoderna från andra deluppgifter finns tillgängliga även om du inte skrivit dom.)

a) Skriv först en metod som genererar slumpmässiga tecken.

*Problem 3. Testar: Swing, händelsehantering, exceptions*

Skriv en enkel konverterare mellan tum och mm och tvärtom. En tum är 25,4mm. Men vi fuskar lite och anger inte tum som man brukar (tex som 1/4", 5/8") utan i decimalform så 0.25 tum skall motsvara 6.35mm. När man startar programmet ser det ut som figuren till vänster. Skriver man i någon av textfälten och trycker return så fylls det andra fältet i. Om man matar in negativt tal eller något som inte är ett flyttal så skall en "felruta" komma upp med ett meddelande, se nedan. Du skall använda en gridlayout,



(14p)

*Problem 4. Testar: fält, sortering*

Skriv en metod `void columnSort(int[][] x)` som givet en matris `x` sorterar varje **kolumn**. Skriv också en metod `print(int[][] x)` som skriver ut en matris. Skriv också kod som skapar matrisen till vänster, anropar `columnSort` och sedan `print`.

Exempel: om `x` är 5x7 matrisen till vänster nedan så skall metoden `columnSort` arrangera `x` matrisen som till höger

12	6	7	17	18	19	8		0	2	1	1	0	3	2
0	14	8	15	5	3	2		1	2	6	3	5	3	2
7	2	1	6	9	18	21		7	6	7	6	7	18	8
1	5	9	3	7	11	2		12	12	8	15	9	19	21
89	12	6	1	0	19	27		89	14	9	17	18	19	27

(10p)

*Problem 5. Testar: rekursion*

Tänk dig talen 1..n och sätt in ett '+' för addition eller ett '-' för subtraktion mellan varje siffra. Summera nu talen och se om du får noll. Skriv ett *rekursivt* program som finner alla sekvenser av längd `n` som ger summan 0.

Exempel: anropet `zeroSum(7)` skall tex ge utskriften

```

1 + 2 - 3 + 4 - 5 - 6 + 7 = 0
1 + 2 - 3 - 4 + 5 + 6 - 7 = 0
1 - 2 + 3 + 4 - 5 + 6 - 7 = 0
1 - 2 - 3 - 4 - 5 + 6 + 7 = 0

```

Tips: spara +/- tecknen i en char array. `zeroSum` ovan är bara en wrapper-metod.

(10p)

`char randomLetter()` - returnerar en slumpmässig bostav A..Ö  
(bokstäverna I,Q,V,Å,Ä,Ö ingår aldrig i ett bilnummer men vi kan tillåta detta)  
Du kan också i det följande anta att följande metod finns tillgänglig:

`char randomDigit()` - returnerar en slumpmässig siffra 0..9

b) Skriv en metod

```
String createRandomCode() {
    som skapar en 6 ställig kod med 3 bokstäver följt av 3 siffror.
```

c) Skriv en metod

```
int hDistance(String s1, String s2)
    som beräknar Hamming avståndet mellan de lika långa strängarna s1 och s2.
    Om strängarna inte är lika långa skall man kasta en exception,
    IllegalArgumentException, med en lämplig ledtext.
```

d) Skriv en metod som kontrollerar om en given kod (code) har Hamming avstånd < hd (dvs skiljer sig åt på färre än hd platser) till någon av koderna i listan med koder (codes). Om så är fallet returneras true. Observera att codes kanske inte är full.

```
boolean codeOK(String code, String[] codes, int hd)...
```

e) Skriv nu en metod

```
String[] generateCodes(int n, int hd)
```

som genererar en lista med n st bilnummer-lik koder enligt beskrivningen och kraven ovan.

f) Skriv nu en klass som frågar användaren hur många koder som önskas samt skriver ut så många koder på skärmen, 10st per rad med ett "space" mellan.

g) Transportstyrelsen (som utfärdar bilnummer) undantar en del ordkombinationer av olika anledningar.

- Ord eller begrepp som kan användas nedsättande om personer, t.ex. DUM, FUL eller LAT.
- Ord eller begrepp med sexuell eller annan kroppslig innebörd, t.ex. LEM, NYP, GAY, FAG eller RAP.
- Ord eller begrepp med drogrelaterad innebörd, t.ex. GHB, LSD eller XTC.
- Förkortningar om vissa organisationer, t.ex. ETA, KKK eller BSS.
- Ord som är siffror, t.ex. TRE, FEM eller SJU.

(Man kan läsa mer om detta på [http://sv.wikipedia.org/wiki/Registrerings skyltar\\_i\\_Sverige](http://sv.wikipedia.org/wiki/Registrerings skyltar_i_Sverige))

Antag nu att vi sparat dessa undantag i en fil "undantag.txt" med ett ord per rad. Skriv en till metod i klassen

```
public ArrayList readUndantag() {
    som läser filen och sparar orden i listan som returneras. Programmet skall sedan kolla även mot dessa ord innan det godkänner en kod. Du redovisar readUndantag och de ändringar som behövs i resten av klassen.
```

Man läser en fil på samma sätt som man läser från tangentbordet men anger filens namn med (`new File ("undantag.txt")`) till Scannern istället för `System.in`. Glöm inte fånga `FileNotFoundException` som skall ge utskriften `File "Undantag.txt" not found`.

(ungefärlig fördelning: 2+2+2+2+2+5+5) (20p)