

## Syfte

Syftet med denna laboration är dels att göra dej bekant med de verktyg som kan vara aktuella i programmeringsarbetet, dels ge en första inblick i att skriva enkla Javaprogram och få övning i att använda if-, while- och for-satserna.

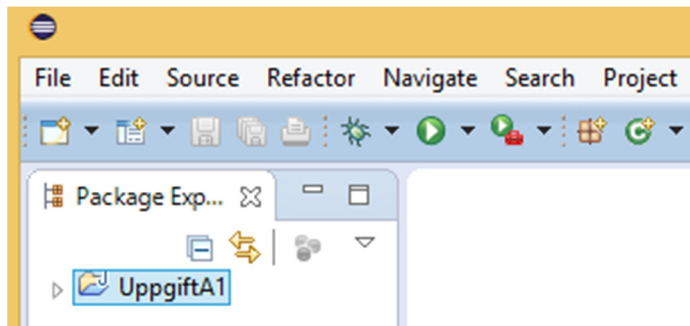
## Del A: Förberedande övningar

Lösningarna på uppgifterna i del A behöver inte redovisas.

### Uppgift A1 - Att kompilera och köra ett Java program

1. Logga in.
2. Starta Eclipse
3. Välj en lämplig workspace-katalog  
(en ny katalog i din hemkatalog, z:\ws\ till exempel på Chalmers Windows-datorer)
4. Skapa ett projekt (file -> new -> java project), döp det till UppgiftA1
5. Stäng välkomst-fönstret om det är öppet.

Har allt gått rätt ska det se ut ungefär så här i övre högra hörnet av Eclipse:



6. Gå till kursens hemsida och ladda ner filen Uppgift1.java, som du hittar på sidans dokumentdel, och gör sedan endera af följande.
  - a) Skapa en ny Java-fil i ditt projekt (new -> Class) som heter Uppgift1, och kopiera innehållet från Uppgift1.java från hemsidan (öppna den i en text-editor)
  - b) Importera Uppgift1.java (file -> import -> General/File System -> Ställ in katalog, välj fil)  
Se till att "Into folder" visar "Uppgift1A/src".
7. Kompilera och kör programmet genom att trycka på den grön-vita kör-symbolen



Fungerar allt som det ska får du upp rutor som frågar efter olika saker, får du inte det så felsök/be om hjälp.

8. Studera programkoden och försök förstå varför programmet gör vad det gör.



## Uppgift A2 - Kompileringsfel

När man skriver ett program kan man naturligtvis göra fel. En typ av fel är s.k. kompileringsfel, vilka uppstår då man inte följer de språkregler som finns för det programspråk man använder. Denna typ av fel upptäcks av kompilatorn när man försöker kompilera programmet och man erhåller felmeddelanden från kompilatorn. Dessa felmeddelanden kan, särskilt för den ovane, vara svåra att tyda. Denna uppgift går ut på att lära sig tyda felutskrifter från kompilatorn för att lokalisera och korrigera felen i ett program.

Inför nedanstående fel i programmet Uppgift1.java . Kompilera om programmet efter varje fel som införs och se vilka felutskrifter som erhålls. Återställ programmet till sitt ursprungliga utseende inför varje förändring (och kontrollera att det kompilerar utan fel).

- Ändra Uppgift1 på rad 5 till uppgift1.
- Ta bort void på rad 6.
- Ta bort { på rad 6.
- Ändra String på rad 7 till string.
- Ta bort ; på rad 7.
- Ändra == på rad 14 till =.
- Ta bort } på rad 22.
- Kommentera bort rad 11 (lägg // i början av raden)
- Kommentera bort en av import-raderna i början av programmet

Tips: I felutskriften anger kompilatorn på vilken rad i programmet den upptäckt felet. För att få radnumren utskrivna i programmet i jGrasp går du in i menyn View och markerar LineNumbers (jGrasp kan också automatiskt markera raden för ett fel om man klickar på felutskriften)

## Uppgift A3 – Mer kompileringsfel

Nedanstående Javaprogram som innehåller ett antal fel finns på kursens hemsida (ExFel.java).

```
public class TExFel {  
    public static void main(String[] arg) {  
        int tal1, tal2;  
        double tal3  
        indata = JOptionPane.showInputDialog("Ange första talet");  
        tal1 = parseInt(indata);  
        indata = JOptionPane.showInputDialog("Ange andra talet");  
        Tal2 = Integer.parseInt(indata);  
        tal3 = tal1 / tal2;  
        JOptionPane.showMessageDialog(null, "Resultatet blev"+tal3);  
    } //main  
} // ExFel
```

Skapa ett nytt projekt (UppgiftA3). Hämta ExFel.java från kursens hemsida och placera den i projektet. Kompilera programmet. Försök förstå de felutskrifter/markeringar som erhålls och försök rätta felen. Konsultera handledare om ni får problem.

När ni lyckas kompilera programmet, kör programmet och ge värdena 2 och 5 som indata.

Varför blir resultatet 0? Borde det inte vara 0.4?

Förklaring: Värdet som skrivs ut beräknas i satsen  $\text{tal3} = \text{tal1} / \text{tal2};$

I denna sats är båda operanderna, tal1 och tal2, i divisionen heltal, vilket i Java innebär en heltalsdivision som betyder att resultatet av divisionen blir ett heltal!! I vårt aktuella fall där 2 divideras med 5 blir resultatet 0. Att utskriften blir 0.0 beror på att resultatet som erhöles från divisionen lagras i variabeln tal3 som är av typen double, vilket innebär att heltalet 0 konverteras till det reella talet 0.0 när det lagras i variabeln. För att erhålla ett reellt tal från en division måste minst en av operanderna vara ett reellt tal. För att göra om heltalet x till ett reellt tal skriver man i Java

(double) x.

- a) Ändra i programmet så att indatavärdena 2 och 5 ger resultatet 0.4.
- b) Utskriften av resultatet blev inte så snygg (texten och talet skrivs ihop). Rätta till!

### Uppgift A4

Hämta Triangel.java från kursenhemsidan, lägg den i ett nytt projekt. Kör programmet och försök förstå vad det gör.

Utgå sedan från programmet Triangel och skriv ett nytt program som via dialogrutor läser in värdena på de båda katetrarna i en rätvinklig triangel samt beräknar hypotenusan och skriver ut resultatet i en dialogruta. Döp ditt nya program till t.ex Triangel2.

### Uppgift A5

Hämta Cirkel.java från kursenhemsidan, lägg den i ett nytt projekt. Kör programmet och försök förstå vad det gör.

Utgå från programmet Cirkel.java och skriv ett nytt program som förutom ytan också beräknar och skriver ut cirkelns omkrets.

### Uppgift A6

Hämta Pris.java som från kursenhemsidan, lägg den i ett nytt projekt. Kör programmet och försök förstå vad det gör.

Utgå sedan från programmet Pris och skriva ett nytt program som ger 5 procents rabatt om totalpriset är större än 750 kronor, 10 procents rabatt om totalpriset är större än 1500 kronor och 15 procent om totalpriset är större än 3000 kronor. Ändra också utskriften på så sätt att uppgifter om bruttopris, rabatt och nettopris skrivs ut.

## Del B

Följande uppgifter skall redovisas för och godkännas av en kursassistent, och därefter ska källkoden lämnas in via kurshemsidan. Efter tredje föreläsningen skall ni kunna lösa samtliga dessa uppgifter. Se kurshemsidan inlämningsuppgift för sista datum för redovisning.

När en uppgift godkänns får ni en 7-siffrig kod av kursassistenten. När ni skickar in ska ni alltså ha 5 koder tillgängliga och skicka in dem efter instruktioner på inlämningssidan.

Det är tillåtet att redovisa alla uppgifter samtidigt, men det kan vara bättre att redovisa dem efter hand. Om man väntar för länge med att redovisa finns det risk att man inte hinner innan deadline.

## Uppgift 1

Skriv din lösning i en fil med namn Exchange.java

Skriv ett program som läser in dels den gällande växelkursen mellan Euro och svenska kronor, dels ett antal svenska kronor, samt beräknar och skriver ut hur många Euro detta motsvarar. Antalet erhållna Euro skall skrivas ut med exakt två decimaler.

## Uppgift 2

**2 a)** Skriv din lösning i en fil med namn Parabola.java

Skriv ett program som läser utgångshastigheten  $v$  (m/sek) och en kastvinkel  $a$  (i grader) och sedan beräknar banhöjden  $h$  och kastlängden  $d$  enligt nedanstående formler för kast utan luftmotstånd:

$$h = \frac{y^2 \cdot \sin^2 a}{2g} \quad d = \frac{y^2 \cdot \sin 2a}{g}$$

Där  $g = 9.81 \text{ m/s}^2$ .

Utskriften av banhöjd och kastlängd skall göras med ett lämpligt antal decimaler.

Tips: I Java finns en standardklass Math. På kurshemsidan finns en förteckning med länkar till standardklassernas dokumentation. Tänk på vilken enhet används i de trigonometriska funktionerna.

Testdata: En utgångshastighet på 20 m/sek och en kastvinkel på 45 grader ger en banhöjd på ca 10.19 meter och en kastlängd på ca 40.77 meter.

**2 b)** Skriv din lösning i en fil med namn Boat.java

För att en båt skall klassas som en 12:a får inte nedanstående uttryck avvika från värdet 12 med mer än 0.05.

$$\frac{2d + \sqrt{A} + L - f}{2,37}$$

I formeln betecknar  $d$  båtens omfång,  $A$  båtens segelyta,  $L$  båtens längd och  $f$  båtens fribordshöjd.

Skriv ett Javaprogram som läser in värden på  $d$ ,  $A$ ,  $L$  och  $f$  och avgör om båten uppfyller villkoret för att klassas som en 12:a.

Läs in samtliga indatavärden via en och samma dialogruta, dvs använd ett objekt av klassen Scanner för att avkoda de enskilda indatavärdena.

**Testdata:** Följande värden ger att båten är en 12:a:

$d = 6.3$ ,  $A = 63.7$ ,  $L = 8.85$  och  $f = 1.02$

**Obs!** En instans av klassen Scanner anropar metoden Local.getDefault() för att ta reda på vilka konventioner som används för att exempelvis ange decimaltal. Den dator ni kör programmet på kan alltså ha svenska konventioner, dvs decimalkomma används istället för decimalpunkt. För att en instans av klassen Scanner som med säkerhet använder decimalpunkt vid inläsning av decimaltal skall ni innan ni skapar instansen av klassen Scanner skriva satsen Locale.setDefault(Locale.UK); för att få engelska konventioner.

## Uppgift 3

Skriv din lösning i en fil med namn DateFormats.java

Skriv ett program som läser ett datum på formen yymmdd (t.ex 991012) och skriver ut datumet på den amerikanska formen mm/dd/yy (t.ex 10/12/99). Använd dialogrutor för inmatning och utskrifter.

Utforma programmet på så sätt att körningen upprepas tills man trycker på Cancel-knappen i

inmatningsrutan. Du får anta att den indata som ges till programmet är korrekta datum på formen yymmdd. Uppgiften skall lösas med användning av heltalsdivision och rest vid heltalsdivision.

**Tips:** Vad blir resultaten av uttrycken  $1234/100$  och  $1234\%100$ ?

**Testdata:** Resultatet av indatasträngen 010203 skall bli 02/03/01.

## Uppgift 4

**4 a)** Skriv din lösning i en fil med namn Leibniz1.java

Den berömde matematikern Gottfried Leibniz gav följande formel för  $\pi$ :

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots$$

Skriv ett program som beräknar  $\pi$  enligt denna formel och tar med de 500 första termerna. Metoden Math.pow får inte användas i lösningen.

**Tips:** Här är ett annat sätt att uttrycka samma samband:

$$\pi = \frac{4}{1} + \frac{-4}{3} + \frac{4}{5} + \frac{-4}{7} + \frac{4}{9} + \frac{-4}{11} + \dots$$

Uttryckt så blir det tydligare att varje steg kan utföras som en addition, och från ett steg till nästa så växlar täljaren tecken (+/-) och nämnaren ökar med 2.

**4 b)** Skriv din lösning i en fil med namn Leibniz2.java

Skriv ett program som beräknar  $\pi$  enligt denna formel och tar med så många termer att den sist medtagna termen är den första termen som till sitt absolutbelopp är mindre än 0.00001. Fundera på: Hur stor är noggrannheten i det värde du erhåller på  $\pi$  om den sist medtagna termen till absolutbelopp är mindre än 0.00001? Hur många decimaler bör du då skriva ut?

## Uppgift 5

**5 a)** Skriv din lösning i en fil med namn Easter.java

I Scientific American, Februari 1981, fanns följande algoritm för att beräkna vilket datum påskdagen infaller under åren mellan 1900 och 2099:

1. Kalla året för Y. Subtrahera 1900 från Y och kalla skillnaden för N.
2. Dividera N med 19. Kalla resten (återstoden) för A.
3. Dividera  $(7A+1)$  med 19. Kalla kvoten för B.
4. Dividera  $(11A+4-B)$  med 29. Kalla resten för M.
5. Dividera N med 4. Kalla kvoten för Q.
6. Dividera  $(N+Q+31-M)$  med 7. Kalla resten för W.
7. Beräkna  $25-M-W$ . Kalla resultatet D.
8. Om D är positivt infaller påskdagen i April och dagen är D, annars infaller påskdagen i Mars och dagen erhålls ur nedanstående tabell:

D	Dag
0	31
-1	30
...	...
-8	23
-9	22

Alla variabler i algoritmen är heltal och alla operationer är heltalsoperationer.

Skriv ett program som använder dialogrutor för att upprepade gånger läsa in ett årtal, samt (med ovan beskrivna algoritm) beräknar och skriver ut när påskdagen infaller för det inlästa årtalet. Exekveringen av programmet avbryts genom att användaren trycker på Cancel/Avbryt vid en inmatning. Programmet skall kontrollera att det inlästa årtalet ligger mellan 1900 och 2099. Om felaktigt årtal ges skall en felutskrift göras och ett nytt årtal begäras.

**Testdata:**

2011 inföll påskdagen 24 april

2000 inföll påskdagen 23 april

1964 inföll påskdagen 29 mars

1910 inföll påskdagen 27 mars

**5 b)** Skriv din lösning i en fil med namn MultiplicationTable.java

Skriv ett program som läser in ett heltal N och genererar en multiplikationstabell med N rader och N kolumner enligt exemplet nedan:

Ange gradtal: 4

1 \* 1 = 1

1 \* 2 = 2      2 \* 2 = 4

1 \* 3 = 3      2 \* 3 = 6      3 \* 3 = 9

1 \* 4 = 4      2 \* 4 = 8      3 \* 4 = 12      4 \* 4 = 16

**Tips 1:** Använd två nästlade for-satser - en yttre som styr antalet rader och en inre som styr antalet kolumner inom raderna.

**Tips 2:** Hur många kolumner skall den inre for-satsen skriva ut på den första raden? På den andra raden? På den tredje raden?