

# Programmeringsuppgift 1

## Avsikt

Avsikten med denna programmeringsuppgift och kommande programmeringsuppgifter är att du ska träna på att programmera. Detta innebär att du på egen hand ska lösa samtliga programmeringsuppgifter. Naturligtvis bör du diskutera olika sätt att lösa uppgiften med dina studiekamrater. Men det är viktigt att du själv skriver all din kod och att du inte kopierar från någon kamrat. Det är inte tillåtet att kopiera kod från någon annan.

Programmeringsuppgiften består av ett antal metoder och en klass som ska skrivas. Du ska kommentera din kod avseende vad varje metod / klassen kan användas till och även skriva radkommentarer. Se nedan.

I zip-filen finns java-filerna *TestP1.java*, *Exercise1.java* och *PaintWindow.java*. Placera dessa i paketet **p1**. Bildfilen *gubbe.jpg* ska du placera i katalogen M:\bilder. Om du jobbar på egen dator måste du ändra i metoderna *exercise1e()* resp *exercise1f()* så att programmet finner bildfilen.

Programmet **TestP1** används för att testa dina lösningar. När du väljer att exekvera metoderna *exercise1e()* och *exercise1f()* kommer programmet att avslutas då du stänger PaintWindow-fönstret.

## Kommentarer i dina lösningar

```
package p1;
import javax.swing.*;
/*
 * Klassen innehåller lösningar på uppgifterna 1a - 1f
 * @author Sten Stensson den 1/10-2014
 */
public class Exercise1 {
    /*
     * Metoden beräknar totala biljettpriset och det genomsnittliga
     * biljettpriset för ett antal vuxna, barn och pensionärer.
     */
    public void exercise1a() {
        // Deklaration av variabler som används i programmet
        int childs, adults, pensioners, totalPrice;
        String res;

        // Användaren matar in antalet vuxna. Lagras i 'adults'
        adults = Integer.parseInt( JOptionPane.showInputDialog( "Ange " ) );
        :
        JOptionPane.showMessageDialog( null, res ); // Meddelande till ...
    }
    :
}
```

## Lösning, granskning och redovisning

### Lösning

Din lösning av uppgiften ska lämnas in via *It's learning* senast kl 09.00 torsdagen den 2/10. Du ska placera *samtliga källkodsfiler i paketet p1* i en zip-fil. I zip-filen ska du också placera aktivitetsdiagrammet i uppgift 1c.

Zip-filen ska du ge namnet AAABBBP1.zip där AAA är de tre första bokstäverna i ditt efternamn och BBB är de tre första bokstäverna i ditt förnamn. Använd endast tecknen a-z när du namnger filen.

- Om Rolf Axelsson ska lämna in sina lösningar ska filen heta AxeRolP1.zip.
- Om Örjan Märta ska lämna in sina lösningar ska filen heta MarOrjP1.zip.
- Är ditt förnamn eller efternamn kortare än tre bokstäver så ta med de bokstäver som är i namnet: Janet Ek lämnar in filen EkJanP1.zip

## Granskning

Ca kl 14.00 den 2/10 kommer en kamrats lösning finnas i din inlämning på It's learning. Din uppgift är att granska kamratens lösningar avseende:

- funktion – hur väl uppfyller lösningen kraven? Är körresultatet korrekt? Hur skulle du lösa eventuella felaktigheter? Har du löst uppgiften på ett annat sätt?
- indentering mm, metodnamn, variabelnamn – hur välskriven är källkoden? Var måste koden rättas till?
- kommentarer – är källkoden väl kommenterad? Var saknas kommentarer?

Din granskning ska omfatta 1-2 A4-sidor. Granskningen ska vara som pdf-dokument och *lämnas in via It's Learning innan du redovisar fredagen den 3/10.* Namnet på granskningen ska vara samma som zip-filen, dvs. AAABBBP1.pdf.

### Granskning av Programmeringsuppgift 1

Lösning inlämnad av Eva Lind

Granskare: Einar Bok

Datum: 2/10-2014

#### Funktion:

exercise1a:

...

exercise1b:

...

...

#### Indentering, metodnamn, variabelnamn mm:

...

#### Kommentarer:

...

## Redovisning

Redovisning sker *fredagen den 3/10*. Redovisningstid publiceras på It's learning under torsdagen den 2/10. Kom väl förberedd till redovisningen. Kom i god tid till redovisningen så du är beredd då det är din tur. Se till att du är inloggad på en dator (eller har egen dator), att eclipse är igång på datorn och att det går att exekvera dina lösningar. Din granskning ska också vara öppen på datorn.

Redovisning sker i grupper om 4-6 studenter.

En redovisning sker genom att:

- Studentens lösningar körs med **TestP1**.
- Granskaren redogör för sina bedömningar. Ta gärna med en utskrift av din granskning!
- Studenten svarar för sina lösningar
- Labhandledaren ställer kompletterande frågor
- De studenter i gruppen som inte redovisar är åhörare.

Godkänd uppgift ska registreras på It's learning av labhandledaren. Om lösningen ej blir godkänd ska bristerna i lösningen uttryckas på It's learning .

Om labhandledaren anser att det endast krävs *mindre komplettering för att lösningen ska godkännas* kan denna komplettering äga rum direkt efter redovisningen. Labhandledaren granskar kompletterad lösning då tiden medger.

Om labhandledaren anser att det endast krävs *mindre komplettering för att granskningen ska godkännas* kan denna komplettering äga rum direkt efter redovisningen. Labhandledaren granskar kompletterad granskning då tiden medger.

I övriga fall måste du lämna in lösning och göra en granskning vid kommande redovisningstillfälle.

## Uppgift 1a

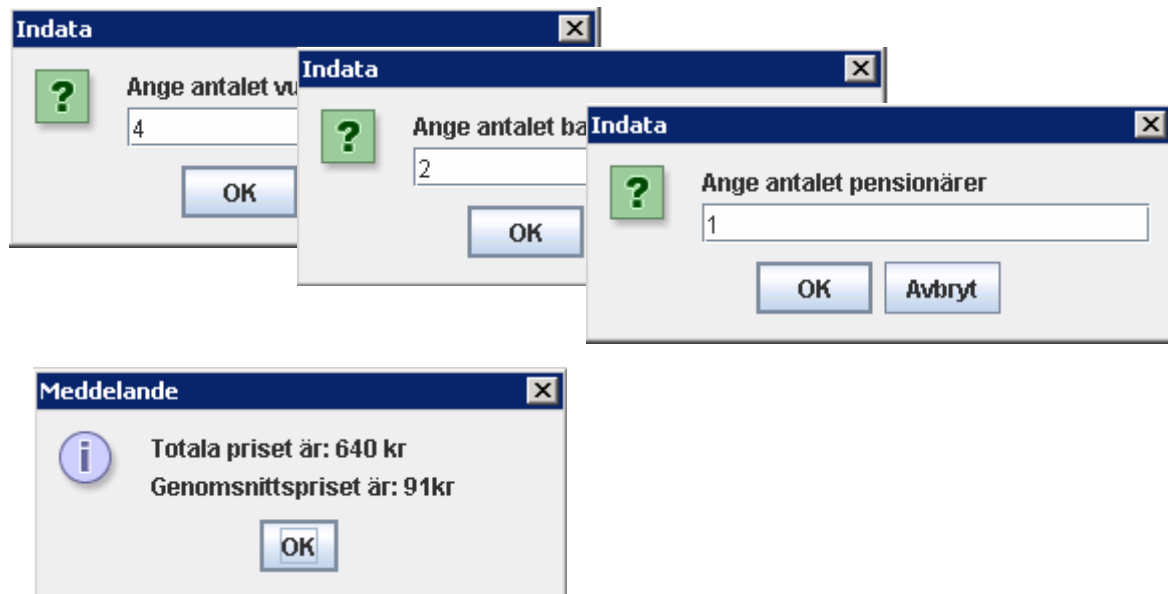
Cirkus Flott har följande biljettpriser:

Vuxen: 110 kr  
Barn: 60 kr  
Pensionär: 80 kr

Komplettera metoden *exercise1a()* i klassen **Exercise1** med kod. Metoden ska

- fråga efter antalet vuxna, barn respektive pensionärer
- beräkna priset för samtliga personer
- beräkna den genomsnittliga kostnaden per person (som heltal, använd heltalsdivision).
- meddela användaren det totala priset och genomsnittspriset

**Körexempel** vid anrop av metoden *price* (fönsterna visas efter varandra):



**Exempel på frågor som kan ställas vid redovisningen:**

- Var är instruktionen där användaren matar in antalet pensionärer?
- Hur låter man användaren mata in ett heltal?
- Till vad använder man metoden `Integer.parseInt`?
- Hur visar man en inmatningsdialog?
- Hur visar man en meddelandedialog?
- Hur/var räknar programmet ut totala biljettpriset?
- Vilka lokala variabler finns i metoden?

## Uppgift 1b

Komplettera metoden *exercise1b()* i klassen **Exercise1**. Metoden ska skriva ut samtliga udda tal i intervallet 1-100. En *for-loop* ska användas för att göra utskriften.

1 3 5 7 9 11 13 15 ... 97 99

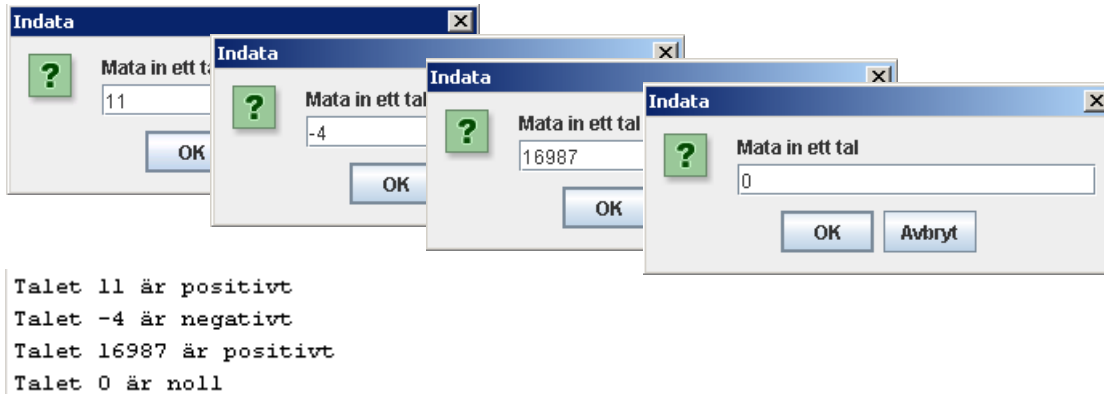
## Uppgift 1c

Du ska skriva metoden **exercise1c()** i klassen **Exercise1**. Metoden ska:

1. fråga användaren efter ett heltal.
2. skriva ut om talet är positivt, negativt eller noll. Denna utskrift ska ske med `System.out.println` som du ser i exemplet nedan.

Metoden ska upprepa ovanstående 4 gånger. Använd en for-loop för detta ändamål.

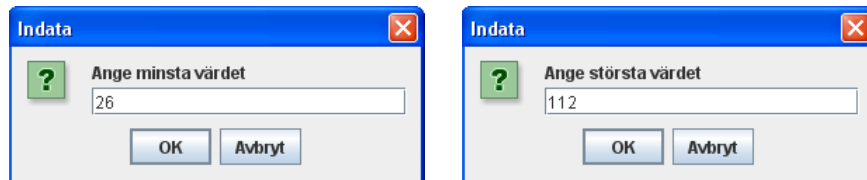
När du löser uppgiften ska du börja med att göra ett **aktivitetsdiagram** över instruktionerna i metoden. Därefter ska du implementera metoden **exercise1c** i klassen **Exercise1**.



## Uppgift 1d

Komplettera klassen **Exercise1** med metoden **exercise1d()**. Metoden ska fråga efter ett intervall, dvs. inmatning av två heltal, min resp max. Sedan ska metoden skriva ut en serie tal med början på min och sedan vart sjunde tal i intervallet. Det ska aldrig skrivas ut ett tal som är större än max.

Ett anrop av metoden kan ge följande körresultat:



1. Först inmatningar:
2. Sedan utskrift i meddelandefönstret:  
26, 33, 40, 47, 54, 61, 68, 75, 82, 89, 96, 103, 110

**Några exempel på körresultat:**

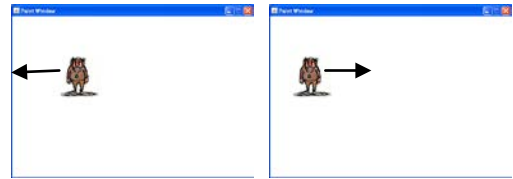
min	max	Körresultat
26	112	26, 33, 40, 47, 54, 61, 68, 75, 82, 89, 96, 103, 110
11	25	11, 18, 25
13	8	

## Uppgift 1e

Metoden *exercise1e()* (se klassen Exercise1) använder ett objekt av typen **PaintWindow** för att visa en gubbe som rör sig.

Exekvera metoden *exercise1e()* och se hur gubben rör på sig. Viktiga variabler i metoden är  $x$ ,  $y$ ,  $dx$  och  $dy$ . Testa med andra värden på  $dx$ , t.ex. -2, -1, 3 och 0 och kontrollera gubbens rörelse.

När gubben kommer till vänster kant ”studsar” gubben och rör sig till höger.



Din uppgift är att komplettera metoden med kod så att:

- gubben ”studsar” när gubben kommer till höger kant dvs så att gubben rör sig åt vänster. När du gjort denna förändring så ska gubben röra sig fram och tillbaka över fönstret, ända tills du stänger fönstret.

**Exempel på frågor som kan ställas vid redovisningen:**

- Hur skapar man ett PaintWindow-objekt?
- Vilken betydelse har variabeln  $x$  i metoden?
- Vilken betydelse har variabeln  $dx$  i metoden?
- Hur tilldelas variabeln  $y$  värde i metoden?

## Uppgift 1f

Uppgift 1f är en fortsättning på uppgift 1e. I uppgift 1f ska du skriva metoden *exercise1f()*. I denna uppgift ska gubben även röra sig i höjdlid och studsa mot överkant och underkant på fönstret. Börja med att kopiera metoden *exercise1e()* och ge kopian namnet *exercise1f()*.

Tilldela  $dx$  värdet 3 och  $dy$  värdet -2 och exekvera metoden *exercise1f()*. Som du ser rör sig gubben snett uppåt åt höger. Men försvinner i fönstrets överkant. Och på samma sätt försvinner gubben nedanför den undre kanten på fönstret om  $dy$  har ett värde större än 0.

Din uppgift är att komplettera metoden med kod så att:

- gubben ”studsar” mot den övre och den undre kanten på fönstret. Nu kommer gubben att röra sig inuti fönstret utan att lämna fönstret.
- se till att variabeln  $x$  tilldelas ett slumpvärde i intervallet 0 – 500, variabeln  $dx$  ett slumpvärde i intervallet -3 – 3 och variabeln  $dy$  ett slumpvärde i intervallet -3 – 3.

**Exempel på frågor som kan ställas vid redovisningen:**

- Som i Uppgift 1e
- Vilken betydelse har variabeln  $y$  i metoden?
- Vilken betydelse har variabeln  $dy$  i metoden?
- Hur fungerar metoden *nextInt* i klassen *Random*?