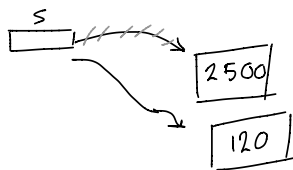


## Gym.java

```
public class Gym {
    public static void main (String[] arg) {
        String s = JOptionPane.showInputDialog("Pris för årskort?");
        double årskort = Double.parseDouble(s);
        s = JOptionPane.showInputDialog("Pris för biljett?");
        double biljett = Double.parseDouble(s);
        s = JOptionPane.showInputDialog("Antal besök?");
        int antal = Integer.parseInt(s);
        if (årskort < antal * biljett)
            JOptionPane.showMessageDialog(null, "Årskort är billigast!");
        else
            JOptionPane.showMessageDialog(null, "Biljetter är billigast,");
    }
}
```



## Mobil3.java

```
public class Mobil3 {
    public static void main (String[] arg) {
        String s = JOptionPane.showInputDialog("Antal minuter?");
        int min = Integer.parseInt(s);
        int svar = JOptionPane.showConfirmDialog(null, "Dagtid?");
        if (svar == 0)
            JOptionPane.showMessageDialog(null, "Kostnad: " + (0.6+min*2.50));
        else if (svar == 1)
            JOptionPane.showMessageDialog(null, "Kostnad: " + (0.6+min*0.55));
    }
}
```

## Studs.java (exempel på while-loop)

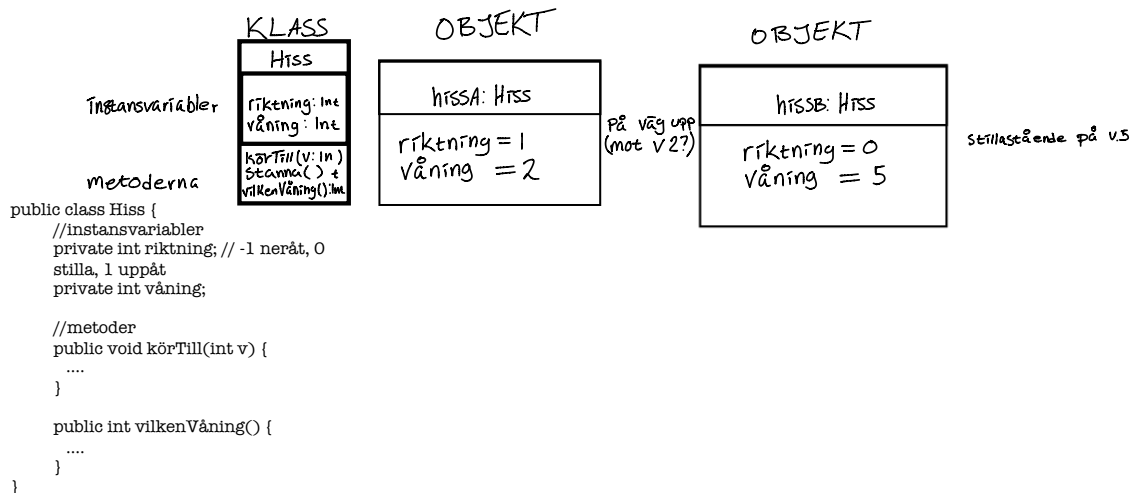
```
public class Studs {
    public static void main (String[] arg) {
        String s = JOptionPane.showInputDialog("Höjd?");
        double h = Double.parseDouble(s);

        int i = 0;
        while (h > 0.01) {
            h = h * 0.7;
            i = i + 1;
        }
        JOptionPane.showMessageDialog(null, "Den studsar " + i + " gånger.");
    }
}
```

## Summa.java (exempel på for-loop)

```
public class Summa {
    public static void main (String[] arg) {
        String s = JOptionPane.showInputDialog("n?");
        int n = Integer.parseInt(s);
        int summa, k;
        for (k=1, summa=0; k <= n; k=k+1)
            summa = summa + k;
        JOptionPane.showMessageDialog(null, "Summan blir " + summa);
    }
}
```

görs bara en gång  
villkoret för att loopen ska fortsätta  
vad som ska göras efter varje varv  
för användas när man vet hur många "n" man har



## Variabler

- Lokala variabler inne i en metod
  - Inne i en metod.
  - Kan bara användas lokalt i den metoden (endast kända i den metoden de är deklarerade)
  - Existerar bara så länge de exekveras
- Instansvariabler
  - Inte i metoden utan ligger i själva klassen (utanför metoden)
  - Man kommer åt dessa
- Klassvariabler
  - Också deklarerade utanför metoden (på samma sätt som instansvariabler) men deklarereras med "static".
- Parametrar till metoderna
  - De har samma egenskaper som de lokala variablerna

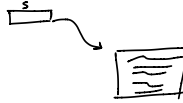
### Två kategorier

- Referensvariabler
- Enkla variabler

Enkel variabel

```
int i;  
i = 4;
```

Referensvariabel (Jämvägar på skrivbordet)



String s:

Stor bokstav, alltid en klass, då är det en ref-variabel

### Deklaration

Modifierare typ variabelnamn ...

Inbyggda numeriska typer			
typ	storlek	minsta värde	största värde
byte	8 bitar	-128	127
short	16 bitar	-32 768	32 767
int	32 bitar	-2 147 483 648	2 147 483 647
long	64 bitar	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
float	32 bitar	Ungefär $-3,4 \times 10^{38}$ 7 siffrors noggrannhet	Ungefär $3,4 \times 10^{38}$ 7 siffrors noggrannhet
double	64 bitar	Ungefär $-1,7 \times 10^{308}$ 15 siffrors noggrannhet	Ungefär $1,7 \times 10^{308}$ 15 siffrors noggrannhet

```
int i, k, m, n;  
i = 4; k = 7;  
m = i++ * k--; (m får värde 4 * 7 men i ökas efteråt, k minskas efteråt)
```