Programmeringsteknik för S1

Övningsgrupp 4

Johannes Hjorth
hjorth@nada.kth.se
Rum 4538 på plan 5 i D-huset
08 - 790 69 02

Kurshemsida:
http://www.nada.kth.se/kurser/kth/2D1311/

Ett litet exempel

Inga genvägar. Bästa sättet att lära sig programmering är att programmera!

```
import java.io.*;

public class Hello {

    public static void main(String[] args) {
        for(int i = 0; i < 4; i++) {
            System.out.println("Hello world!");
        }
    }
}
```

Om vi sparar ovanstående fil som Hello.java, kompilerar och kör...

```
>javac Hello.java
>java Hello
Hello world!
Hello world!
Hello world!
Hello world!
```

Vad bör man tänka på?

Vad är det vi vill åstadkomma med programmet?
Beräkna arean av en rektangel.
Vad behöver programmet hålla reda på?
Basen och höjden.
Vilka variabler behöver vi?
Två heltal b och h.
Hur skulle du räkna ut uppgiften?
Basen gånger höjden ger arean.
Kan vi formulera det som en algoritm?
Ja, se ovan.

Hur överför vi vår ide till ett program?

```
// Importera javas standardbibliotek för input/output
import java.io.*;

// Definera vår klass, innehåller variabler och metoder
public class RektangelArea {

    // Main metoden körs och utför vår algoritm
    public static void main(String[] args) throws IOException {

        // Deklarera variabler som programmet behöver
        String bas, höjd;
        int b, h, area;
        BufferedReader indata;

        indata = new BufferedReader(new InputStreamReader(System.in));

        // Fråga om basen och höjden
        System.out.print("Basen är: ");
        bas = indata.readLine();

        System.out.print("Höjden är: ");
        höjd = indata.readLine();

        // readLine() retunerar strängar, gör om dem till heltal
        b = Integer.parseInt(bas);
        h = Integer.parseInt(höjd);

        // Beräkna arean
        area = b * h;

        // Berätta resultatet för användaren!
        System.out.println("Arean är: " + area);
    }
}
```

Varför måste vi kompilera java filen?

Datorn kan inte förstå java i den form vi skriver. Genom att köra,

```
>javac RektangelArea.java
```

översätter vi vår programkod till java bytekod.

När vi sedan skriver,

```
>java RektangelArea
```

översätts bytekoden till maskinkod och exekveras.

Olika datorer har olika maskinkod, genom att spara programmet i bytekod kan det köras på många olika plattformar utan att behöva kompileras om.

Det är billigare att utveckla en version som fungerar på alla datorer, en av anledningarna till varför java har nått så stor framgång.

En annan fördel med java är den relativt enkla minneshanteringen. Mer om garbage collectorn senare.

Nu kör vi vårt program!

Observera att vi inte tar med .class i anropet

```
>java RektangelArea
Basen är: 7
Höjden är: 6
Arean är: 42
```

Det verkar fungera. Trevligt!

Ett till exempel...

Programmet frågar efter ett heltal och beräknar dess kvadrat. Detta upprepas tre gånger.

```
import java.io.*;

public class Kvadrat1{

    public static BufferedReader stdin = new BufferedReader
        (new InputStreamReader(System.in));

    public static void main(String[] args) throws IOException {
        System.out.print("Skriv ett heltal? ");
        int tal1 = Integer.parseInt(stdin.readLine());
        System.out.println("Kvadraten av " + tal1 + "=" + (tal1 * tal1));

        System.out.print("Skriv ett heltal? ");
        int tal2 = Integer.parseInt(stdin.readLine());
        System.out.println("Kvadraten av " + tal2 + "=" + (tal2 * tal2));

        System.out.print("Skriv ett heltal? ");
        int tal3 = Integer.parseInt(stdin.readLine());
        System.out.println("Kvadraten av " + tal3 + "=" + (tal3 * tal3));

    } // main()

} // class Kvadrat1
```

...som ser ut så här när det körs

I shell fönstret skriver vi,

```
>javac Kvadrat1.java
>java Kvadrat1
Skriv ett heltal? 12
Kvadraten av 12=144
Skriv ett heltal? 13
Kvadraten av 13=169
Skriv ett heltal? 15
Kvadraten av 15=225
```

Men...

Varför skriver vi om samma kod tre gånger?

```
System.out.print("Skriv ett heltal? ");
int tal1 = Integer.parseInt(stdin.readLine());
System.out.println("Kvadraten av " + tal1 + "=" + (tal1 * tal1));

System.out.print("Skriv ett heltal? ");
int tal2 = Integer.parseInt(stdin.readLine());
System.out.println("Kvadraten av " + tal2 + "=" + (tal2 * tal2));

System.out.print("Skriv ett heltal? ");
int tal3 = Integer.parseInt(stdin.readLine());
System.out.println("Kvadraten av " + tal3 + "=" + (tal3 * tal3));
```

Borde det inte gå att förenkla på något sätt?

Vi inför en ny metod!

```
import java.io.*;

public class Kvadrat2{

    public static BufferedReader stdin = new BufferedReader
        (new InputStreamReader(System.in));

    public static void main(String[] args) throws IOException {

        //Anrop av metoden
        beräknaKvadrat();

        //Anrop av metoden
        beräknaKvadrat();

        //Anrop av metoden
        beräknaKvadrat();

    } // main()

    //Metod som frågar om ett heltal och räknar ut kvadraten
    public static void beräknaKvadrat() throws IOException{

        System.out.print("Skriv ett heltal? ");
        int tal = Integer.parseInt(stdin.readLine());
        System.out.println("Kvadraten av " + tal + "=" + (tal * tal));

        /*
          Kvadraten kan även räknas ut med pow-metoden som finns i
          klassen Math.
          double kvadrat = Math.pow(tal, 2);
        */

    } // kvadrat()

} // class Kvadrat2
```

Det går att förenkla ytterligare...

Vi kan ersätta de tre anropen,

```
//Anrop av metoden
beräknaKvadrat();

//Anrop av metoden
beräknaKvadrat();

//Anrop av metoden
beräknaKvadrat();
```

med en liten for-slinga,

```
for(int i = 0; i < 3; i++)
    beräknaKvadrat();
```

där i startar på 0 och ökar med ett i varje iteration. Så länge i är mindre än tre anropas metoden beräknaKvadrat. Loopen görs totalt tre gånger.

Stegvis utveckling av program

När vi gör större program kan det vara bra att bara skapa skalet på de olika metoderna först.

```
import java.io.*;

public class Kvadrat3{

    public static BufferedReader stdin = new BufferedReader
        (new InputStreamReader(System.in));

    public static void main(String[] args) throws IOException {

        //Anrop av metoden
        beräknaKvadrat();

        //Anrop av metoden
        beräknaKvadrat();

        //Anrop av metoden
        beräknaKvadrat();

    } // main()

    //Metod som frågar om ett heltal och räknar ut kvadraten
    public static void beräknaKvadrat() throws IOException{

        System.out.print("Skriv ett komplext tal? ");
        String tal=stdin.readLine();
        System.out.println("Funkar inte ännu!!!");
        /*
          Jag måste komma på hur man beräknar kvadraten av komplexa tal.
        */

    } // kvadrat()

} // class Kvadrat3
```

Inför Lab2

I lab2 ska ni skriva ett program som frågar användaren om längd och vikt och sedan beräknar BMI med hjälp av en funktion.

Några frågor kan vara nyttiga att veta svaret på är,

Hur läser man in en sträng?

Hur konverterar man en sträng till ett heltal?

Hur skriver man en for slinga?

Hur skriver man en if-else sats?

Hur anropar man en metod?

Hur läser man in från tangentbordet?

Hur konverterar man en sträng till ett tal?

Så här kan man göra...

Inläsning av sträng och konvertering till heltal,

```
System.out.print("Talet är: ");
String tal = stdin.readLine();
int t = Integer.parseInt(tal);
```

En for slinga skrivs på följande sätt,

```
for(int i = 0; i < 10; i++) {
    // Det som skrivs här kommer upprepas tio gånger
    // då i = 0, i = 1, ... i = 8 och i = 9
}
```

En if/else skrivs enklast,

```
if(a < b) {
    System.out.println("a är mindre än b");
} else {
    System.out.println("a är större eller lika med b");
}
```

Exempel på hur man definerar en metod;

```
public static double sum(double a, double b) {
    return a+b;
}
```

Obs! Detta ska ligga innanför klassens måsvingar!

Vi behöver också veta hur man anropar dessa metoder från till exempel `main`-metoden,

```
double x = sum(val1,val2);
```

Tänk på att returtypen och variabeln du sparar returvärdet i måste matcha. Om metoden retunerar en sträng kan du inte spara resultatet i en int.

Ett miniexempel!

Kort exempel på hur ett program som gör ett metodanrop kan se ut

```
import java.io.*;

public class MetodAnrop {

    public static void main(String[] inparametrar) {
        int x = 4, y = 7;
        int summa = sum(x,y); // Metodanrop!

        System.out.println(x + " + " + y + " = " + summa);
    }

    public static int sum(int a, int b) {
        return a + b;
    }

}
```

Kort exempel på inlästning av en sträng samt konvertering till heltal, hämtat från `Kvadrat1`.

```
System.out.print("Skriv ett heltal? ");
int tal1 = Integer.parseInt(stdin.readLine());
System.out.println("Kvadraten av " + tal1 + "=" + (tal1 * tal1));
```