### 3.1 Matematiske funktioner

De matematiske funktioner som sinus, cosinus, kvadratrod osv. kaldes i Java med Math.sin(x), Math.cos(x), Math.sqrt(x) osv., hvor x er en variabel, et fast tal eller et beregningsudtryk. Prøv i editoren at skrive Math. så kan du se hvilke metoder og konstanter du kan kalde via Math biblioteket. F.eks. er ligger pi som en konstant i Math biblioteket. Pi kaldes ved at skrive Math.PI. Husk vinkler regnes i radianer ligesom i Mathcad.

Vi kan f.eks. lave en tabel over værdierne af kvadratrod-funktionen Math.sqrt( ) for x=0 til x=10 med programmet (senere, i afsnittet om løkker, vil vi se en smartere måde).

**public class** Kvadratrod

{

**public static** **void** main(String[] args)

{

System.out.println(”kvadratroden af 0 er ” + Math.sqrt(0));

System.out.println(”kvadratroden af 1 er ” + Math.sqrt(1));

System.out.println(”kvadratroden af 2 er ” + Math.sqrt(2));

System.out.println(”kvadratroden af 3 er ” + Math.sqrt(3));

System.out.println(”kvadratroden af 4 er ” + Math.sqrt(4));

System.out.println(”kvadratroden af 5 er ” + Math.sqrt(5));

System.out.println(”kvadratroden af 6 er ” + Math.sqrt(6));

System.out.println(”kvadratroden af 7 er ” + Math.sqrt(7));

System.out.println(”kvadratroden af 8 er ” + Math.sqrt(8));

System.out.println(”kvadratroden af 9 er ” + Math.sqrt(9));

System.out.println(”kvadratroden af 10 er ” + Math.sqrt(10));

}

}

Programudskriften bliver:

kvadratroden af 0 er 0.0

kvadratroden af 1 er 1.0

kvadratroden af 2 er 1.4142135623730951

kvadratroden af 3 er 1.7320508075688772

kvadratroden af 4 er 2.0

kvadratroden af 5 er 2.23606797749979

kvadratroden af 6 er 2.449489742783178

kvadratroden af 7 er 2.6457513110645907

kvadratroden af 8 er 2.8284271247461903

kvadratroden af 9 er 3.0

kvadratroden af 10 er 3.1622776601683795

På næste side ser du et program, der udregner længden af den skrå side (hypotenusen) af en retvinklet trekant ud fra længden af dens to lige sider (kateder): kvadratroden af a²+b², hvor a og b er længderne af de to sider:

**public class** Trekant

{

**public static void** main(String[] args)

{

**double** a, b, hypotenuse;

a = 3;

b = 4;

hypotenuse = Math.sqrt(a\*a + b\*b);

System.out.println(”En ligesidet trekant med sider ”+a+” og ” +b);

System.out.println(”har hypotenuse ”+hypotenuse);

}

}

Programudskriften bliver:

En ligesidet trekant med sider 3.0 og 4.0

har hypotenuse 5.0

Her er et tilsvarende program, der udregner hypotenusen ud fra længden af en af de andre sider og den modstående vinkel. Det gøres ud fra formlen a/sin(v), hvor a er længden af siden, og v er vinklen (i radianer):

**public class** Trekant2

{

**public static void** main(String[] args)

{

**double** a, v, hypotenuse;

a = 10;

v = 0.3; // svarer til ca. 34 grader.

hypotenuse = a/Math.sin(v);

System.out.println(”Hypotenusen har længden: ”+hypotenuse);

}

}

Programudskriften bliver:

Hypotenusen har længden: 33.838633618241225

Ud over de almindelige matematiske funktioner findes også Math.random( ), der giver et tilfældigt tal mellem 0 og 0.999999...

### 3.2 Kald af metoder

Math.sqrt( ), Math.sin( ) og de andre matematiske funktioner og andre kommandoer, f.eks. System.out.println( ), kaldes under et *metoder*.

En metode er en navngiven programstump, der kan gøre et eller andet eller beregne en værdi. F.eks. *gør* System.out.println() det, at den skriver tekst på skærmen, og Math.sqrt() *beregner* en kvadratrod. Når en metode nævnes i teksten, skriver vi altid ”()” bagefter, så man kan se, at det er en metode.

Nedenstående linie indeholder et *metodekald*:

hypotenuse = a/Math.sin(v);

Math.sin er navnet på metoden, og man kalder det v, der står inde i ”( )”, for argumentet eller parameteren.

Et metodekald er en nævnelse af en metodes navn efterfulgt af de rigtige parametre. Parametrene er omgivet af parenteser.

Ved et kald uden parametre skal man stadig have parenteserne med. Math.random( ) skal kaldes uden parametre. Her er et eksempel på et metodekald af Math.random( ):

**double** tilfaeldigtTal;

tilfaeldigtTal = Math.random();

***Ved et metodekald kan man som parameter indsætte alt, der har et resultat af den rigtige type: Konstanter, variabler, regneudtryk og resultatet af et andet metodekald***

Følgende er altså tilladt:

double v,x;

x = Math.sin(0.1); // konstant som parameter

x = Math.sin(v); // variabel som parameter

x = Math.sin(Math.sqrt(0.3)); // værdi af andet metodekald som

*// parameter*

Vi vil senere se, hvad der sker, når computeren udfører et metodekald, samt lære, hvordan man kan lave sine egne metoder.

### 3.3 Logiske variabler

En boolsk variabel, også kaldet en logisk variabel, kan kun indeholde værdierne sand eller falsk. Den bruges oftest til at huske, om noget er sandt eller ej, men kan også bruges til at repræsentere noget, der kun har to tilstande, f.eks. om en lampe er tændt eller slukket.

Variabeltypen hedder boolean, og den erklæres med f.eks.:

**boolean** detErForSent;

En boolesk variabel kan kun sættes til værdierne true eller false. F.eks.:

detErForSent = false;

På højre side af lighedstegnet kan stå et logisk udtryk, dvs. et udsagn, der enten er sandt eller falsk, f.eks. "klokken er over 8" (her forestiller vi os, at vi har variablen klokken)

detErForSent = klokken > 8;

Udtrykket klokken > 8 undersøges af Java ved at indsætte værdien af variablen i regneudtrykket og derefter afgøre om udsagnet er sandt. Hvis f.eks. klokken=7, står der 7>8, det er ikke sandt, og detErForSent får værdien false. Hvis klokken=10, står der 10>8, det er sandt, og detErForSent får værdien true.

**Opgave 3.1** Skriv et program, som bestemmer og udskriver skæringen med x – aksen på

følgende parabel y = x2 + 2x - 9

Dvs. find evt. nulpunkter for følgende ligning

x2 + 2x – 9 = 0

**Hint:** **Generelt**

Anden gradsligning

Diskriminant

Hvis fås løsning

Opgaven afleveres på Fronter.  
Du skal kun aflevere filen: *filnavn.java*

som findes under f.eks. D:\NetBeansProjects\JavaApplication1\src alt afhængig af hvor I har gemt jeres projekter. Filerne ligger altså i SRC-mappen (source mappen, kildemappen).

God arbejdslyst

Just