Processing 101

I det følgende vil jeg kort komme ind på de forskellige emner, som er blevet præsenteret i undervisningen.

# Variabler

##### Variabler kan inddeles i to grupper. **Primitive og Composite.**

Primitive er variabler der indeholder data af en bestemt type, eksempelvis:

Int (Heltal) eksempelvis: 1,2,4.

Float (kommatal): 2.5 , 4.5 , 10.555.

Boolean (sandheds operator): true , false.

Variabler som disse skal deklareres før man kan bruge dem. Det betyder at man skal afsætte et variabel navn og eventuelt en start værdi.

Eksempel:

int x; // her oprettes en variabel med navn x af typen int.

Int x = 0; // lige som ovenfor men har sættes startværdien til 0

Globale variabler er når man deklarere variablerne i toppen af sit program. De kan så bruges af alle funktioner og classes i programmet.

Lokale variabler er når man deklarere sine variabler inden i en funktion eller class. Disse variabler kan kun bruges af funktionen eller den givende class.

Composite variabler er avanceret datatyper som ofte er sammen satte af flere elementer eller primitive.

Eksempel:

Int[] x; // Et array som er en indekseret liste fra 0 og frem med navn x af typen int.

Int[] x = new int[8]; // som ovenfor men med længden 8.

Car x; // Et objekt af typen Car, objekter defineres i class af samme navn, her class Car.

# Funktioner

En funktion er et stykke kode der kan kaldes fra et vilkårligt sted i koden eller fra en anden funktion. Dette kaldes også et funktions kald, da men kalder funktionen fra et andet sted.

Funktioner har følgende syntax.

Void minFunktion(){

Koden som funktionen skal udføre skal stå her

}

Hvis man så vil kalde sin funktion gør man det på følgende måde.

minFunktion();

##### Ovenstående funktion udførere bare sin kode, men det er muligt at give en funktion parameter med. Det vil sige at man kan give funktionen et input med når man kalder den.

Eksempel :

Void minFunktion(int x, int y){

Koden som funktionen skal udføre skal stå her

}

Så vil funktions kaldet se ud på følgende måde.

minFunktion(10, 5);

I de to ovenstående eksempler står der void foran funktionen hvilket betyder at funktionen ikke returnerer en værdi. Hvis man vil have funktionen til at gøre dette er syntaxen.

Int minFunktion(){ // denne funktion returnere en int værdi

Koden som funktionen skal udføre skal stå her

Return x; // her returneres en int værdi

}

Hvis en funktion returnere en værdi kan man bruge den til at sætte værdien på eksempelvis en anden variable.

Eksempel:

Int tal = minFunktion();

# Class

En class er en måde at lave nye objekter på, som kan have indbyggede funktioner og indeholde mange forskellige variabler samlet i selve objektet.

En class består af Class name, construktor og metoder

Class name :

class Car{

her skal lokale variabler , construktoren, metoder skrives

}

## Construktor:

Når der laves et nyt objekt køres koden i constuktoren en gang, det svare lidt til setup. Construktoren skal være den første funktion der står i class og den skal have samme navn som class.

Car(){ // vigtigt der må ikke stå void eller noget andet foran men, der må gerne sendes parameter med

Koden som funktionen skal udføre skal stå her

}

## Metoder:

En metode er en indbygget funktion i et objekt, se funktion.

Samlet opsætning af class

class Car{

Variabler her

Car(){ // construktor

}

void minFunktion(){ // Metode

}

}

Eksempel:

Hvis man skal bruge opjektet skal den først deklareres lige som en variable.

car minBil = new car(); // her skabes objektet og construktoren køres en gang.

hvis man vil bruge en metode på objektet kalder man det på følgende måde

minBil.minFunktion(); // dette kalder den indbyggede funktion i class car.