

Programmering

30.08.2021
Mandag

↳ public static void main (String [] args)
 ↳ {x} ≈ start til slutt.

↳ System.out.println (x)
 ↳ showMessageDialogue

Først null i den (heter parameter) også
komma og tilslutt strengen vi skal ha
ut.

↳ Også får å få ut alt er
det showMessageDialogue -
(null, «navn på streng»)

↳ "\n" ≈ enter
↳ /* */ ≈ skjule
↳ // ignorere

Variabler 1

int → Heltall (0, 1, 2 etc) [for integer]
string → tekst

String fornavn;
String etternavn;

→ string fornavn, etternavn;

Tilordningsinstruksjon:

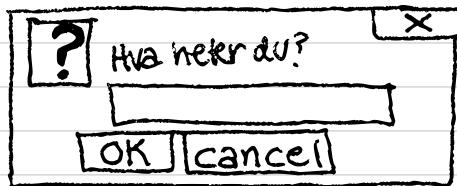
[variabel = verdi;]

NB! likhetsteget her brukes som tilordningsoperatoren

Eks:

fornavn: "Tone"; // variabelen fornavn får verdien "Tone"
etternavn: "Veli"; // variabelen etternavn får verdien
 "Veli"

InputDialog →



Da må man skrive

`JOptionPane.showInputDialog ("Hva heter du?");`

Det som blir lest inn blir av programmet alltid tolket som en tekststrenge. Dersom vi skal ha nytte av noe som er lest inn må vi lagre det i en variabel.

`String navn; // deklarerer variabel som skal brukes til lagring.`

`navn = JOptionPane.showInputDialog("Hva heter du?");
// leser inn og lagrer det som er skrevet`

HUSKEREGLER:

```
{ public static void main (String [] args)  
    PSVM (s [ ja]) }
```

Merke

Hvis vi skriver: `import static javax.swing.JOptionPane.*;`
istedet for: `import javax.swing.JOptionPane;`
slipper vi å skrive `JOptionPane` foran `showInputDialog`
etc.

MERK:

I dette kompediet er første { skrevet i en blokk på linjen under. dette er ikke lengre standard i Java. Standarden nå er å skrive følgende { på samme linje.

Feks: `public class Hilsen4.`

Tall:

- Heftall int (integer)
- Desimaltall double

Man kan gjøre koder kortere med:

import static javax.swing.JOptionPane;

sammensatte ord i variabler, metoder etc.:

- Klasser skal ha stor forbokstav
- Metoder og variabler skal ha liten forbokstav
- Hvert nye ord etter skal ha stor forbokstav
- Bruk navn som begynner etter indikator noe

Regnrekkefølge:

- venstre → høyre
- parenteser
- multiplikasjon
- divisjon
- addisjon
- subtraksjon

P(E)
M
D
A
S } PEMDAS

Casting → Konvertering mellom typer (int og double)
f.eks:

varierer
if ($x == y$) {
 System.out.println / showMessage("x")
}
int tall5 = 23;
int tall6 = 3;
double tall7 = (double) tall5; $\rightarrow 23.0$

Oppgave ⑧ aritmetikk:

$$\text{tall1} = 1$$

$$\text{tall2} = 2$$

$$\begin{aligned} &= \text{tall1} \cdot 4 + (\text{tall2} \cdot 2 + \text{tall1}) / (\text{tall2} - \text{tall1} \cdot 3); \\ &= 4 + (5) / -1 \\ &= \underline{\underline{-1}} \end{aligned}$$

Oppgave ⑨ aritmetikk:

$$\text{tall1} = 5$$

$$\text{tall2} = 4$$

$\text{tall1} == \text{tall2}$:	false
$\text{tall1} != \text{tall2}$:	true
$\text{tall1} < \text{tall2}$:	true
$\text{tall1} \geq \text{tall2}$:	false
$\text{tall1} < \text{tall2}$:	true
$\text{tall1} > \text{tall2}$:	false

Klasser og Objekter

19.09.21

```
public class Halloverden {
```

```
    public static void main (String[] args) {
```

```
        System.out.println ("Hallo verden!");
```

```
}
```

```
}
```

Her er "Hello verden" er klasse og main er en metode inni denne klassen. "out" er en klasse og "println" er en metode i denne klassen.

Når noe er private i en klasse kan det ikke
brukes i main klassen, public kan.
(attributter)

Math.PI → på 3,14

Math. POW → før å oppheve noe. Danner math (x, y).

Dersom jeg skal koble i på med tallstifter:

int a = 1234;

int b = 0b 100100010;

int c = 0137;

int d = 0xAB7D

// starter med 0b

// starter med 0

// heksadeimal starter med 0.

[DecimalFormat df2 = new DecimalFormat();]

switch (noest skal sette ner) {

case "+":

resultat = tall1 + tall2;

break;

osv — " —

}

Konstruktører er en metode som heter det samme som klassen selv. Konstruktør-metoden utføres når et objekt instansieres (opprettes, utføres new). Konstruktører selv kan ikke returnere noe data. Selv om konstruktører ikke kan returnere data skal det ikke stå void mellom public og metodenavnet.

Som oftest lages konstruktøren til å initialisere attributtene i klassen fordi man da blir fritt til å sette verdier allerede når objektet krysses. Det betyr at når jeg skriver "Class" "navn" = new "navn" ("Rød"); "Class". vistarge();

putter inn
verdi

Det er fordi konstruktøren i klassen kan

```
public "klassenavn" (String farge) {  
    this.farge = farge;  
}
```

Ved å sette attributtene i en klasse til "private" er de ikke tilgjengelige for andre enn metoder i selve klassen. Da må man krysse get/set metoder for å hente ut data og lagre data i disse. Dette er fordi man kan sette inn info i parametere og sikre at det førs ikke stikkes bokstaver i nummer tall skal være eller motsatt.

Tre logiske operatører (booleske uttrykk)

$$\begin{aligned} \&f &= \text{og} \\ \&\| &= \text{eller} \\ \&! &= \text{i ikke/vk} \end{aligned}$$

Disse bruktes hovedsaklig i if-setninger og while-løkker.

Switch

```
switch ("varer") {
    case "x":
        resultat = tall1 * tall2;
        break;
    case "y":
        resultat = tall1 + tall2;
        break;
    case "z":
        resultat = tall1 % tall2;
        break;
    default:
        resultat = 0;
        break;
}
```

Det finnes en innebygget Java-klasse "SecureRandom" som har en metode nextInt() som leverer tilfeldige tall.
(annet valg Math.Random())

Terningkast eksempel:

```
for(int i=0; i<antallkast; i++){
    kast = tilfeldigtau. nextInt(6);
    kast++;
    if (kast == 1) {
        terning1++;
    }
    else if (kast == 2) {
        terning2++;
    }
    osv. .... - - -
    else if {
        terning6++;
    }
}
```

Samme koden på switch form blir

```
switch (kast) {  
    case 1:  
        terning1++;  
        break;  
    case 2:  
        terning2++;  
        break;  
    case ...  
    default:  
        terning6++;  
        break;  
}
```

If!

```
if (bedingelse)  
    instruksjon1  
else  
    instruksjon2
```

Ordet Algoritme brukes generelt som betegnelse på en fast bestemt framgangsmåte for å oppnå et bestemt resultat.

$$\begin{array}{ll} x - = y & \rightarrow \\ x \cdot = y & \rightarrow \\ x / = y & \rightarrow \\ x \% = y & \rightarrow \end{array} \quad \begin{array}{l} x = x - y \\ x = x \cdot y \\ x = x / y \\ x = x \% y \end{array}$$

Økninger og minskinger av variabler:

```
int antall = 0;  
antall = antall + 1  
antall += 1;  
antall++;  
antall = antall - 1  
antall -= 1  
antall --
```

samme

samme

eller:

$$\begin{aligned} \text{antall} &= \text{antall} + 2 && \left. \right\} \text{samme} \\ \text{antall} &+= 2 && \\ \text{antall} &= \text{antall} \cdot 2 && \left. \right\} \text{samme} \\ \text{antall} &= 2 && \end{aligned}$$

ctrlarrays

Array er liste av heltall, stringer, double. Array er selv et objekt.

Kodecs

```

int [ ] liste = { 23, 2, -4, 5 };
string ut = "Her er tallene i listen: ";
for ( int i=0; i < liste.length(); i++ ) {
    ut += liste [ i ] + " ";
}

```

```
System.out.println(at);
```

```
package array.obj;
import static javax.swing.JOptionPane.*;
class film {
    private String tittel;
    private double pris;
    public
```

Når man har en Array liste, f.eks:

double[] liste1 = {1,2,3,5,-3,0,..., osv})

og man bruker:

Array.sort(liste1);

då vil hele lista bli sortert.

Altstå med Array.sort(x); hvor x = liste2.

double[] liste2 = Arrays.copyOf(liste1, liste1.length);

definører ny liste som kopi av den første.

double[] liste3 = liste2;

Når er liste 2 og 3 identiske, av i strive:

boolean b = Arrays.equals(liste2, liste1);

Teksteller programmet med true/false verdier (booleske) om de er like eller ikke.

Når man søker:

int location = Arrays.binarySearch(liste1, 5, 6);

Så leter programmet gjennom listen og ser hvilke plass 5,6 ligger på.

JAVA KURS

04.11.19

Companies usually use Java 11 or Java 1.8 because they have long term support.

IntelliJ IDEA

En IDE er forenklet sagt "text editor with lots of features and facilities".

packages er med på å organisere dine klasser sammen.
Akkurat som en mappestruktur (sjekk ved å høyreklikke på koden og trykk på "reveal in finder" (på macbook)).
Alle klasser i java med stor bokstav velst.

Variablene char holder verdier som enkelt bokstav:
`char nameInitial = 'A';`

Reference type Variabler starter med stor forbokstav som f.eks String. Primitivs starter med liten som int, double osv...

En metode som gjør alt til store bokstaver er `toUpperCase()`. F.eks:

```
String navn = new String (original): "Anna");  
System.out.println(navn.toUpperCase());
```

Hvis jeg vil vite datoen idag:

```
LocalDate now = LocalDate.now();  
System.out.println(now);
```

Med engang jeg skriver . i sout til now kommer det opp verdig mange metoder.

Variabelnavn burde alltid starte med liten bokstav.

String er en klasse så når vi skriver String og et navn ved siden av så er det navnet et objekt av type string, og de objektene har mange egenskaper etterfulgt av punktum.

Søndag 07.11.2021

① "import java.util.ArrayList;" ? ✓

② For-each løkke?

JTextArea Start hittomme

TextField hite linje

```
int search (int [ ] kilde, int nøkke) {
    for (int i = 0; i < kilde.length; i++)
        if (kilde[i] == nøkkel)
            return i;
    return -1;
}
```

Hvis vi har en sortert liste, er binær søking bedre:
søkestrategi: søker først i midten av arrayet.
Hvis ingen koll skal det avgjøres om søkingen skal mot hoyre eller venstre.

Hvis jeg skal sjekke om to string eller variabler x, y er like:

```
System.out.println(x.equals(code));
```

slaget kommer i boolsk verdi.

(int) Math. pow (grunn tall, eksponent);

Kvadratrot: Math. Sqrt (tall);

Pi er Math.PI

Høgste operatører & & og
& eller

Dersom du bare har ett uttrykk som du vil få fram ved en if setning kan det også skrives i string:

int alder = 18;

String alderTest = alder >= 18 ? "Jeg kan ta lappen":
"Jeg er for ung".

switch

NB!

Dette burde ikke brukes da det handler om å bytte rundt om med en variabel (f.eks string) → else:

String gender = "DAME";

```
switch (gender) {  
    case "DAME":  
        System.out.println("Jeg er en dame");  
        break;  
    case "MANN":  
        System.out.println("Jeg er en mann");  
        break;  
    default:  
        System.out.println("Ukjent");  
}
```

Arrays

Hvis jeg koder en Array og skriver ut tom kommer default "0". String "null" og boolean "false". For å system.out.println en tom Array skriver man:

system.out.println (Arrays.toString (liste));

hvor liste er navnet på arrayet.

To for-løkker skrevet helt likt:

```
int [ ] numbers = { 1, 4, 8, 16 };
```

```
for (int i = 0; i < numbers.length; i++) {  
    system.out.println (numbers[i]);
```

```
}
```

eller

```
for (int numbers: numbers) {  
    System.out.print (numbers);  
}
```

Vilfritt navn

Skrive ut komplisert måte:

```
Arrays.stream (numbers). forEach (system.out.println);
```

Når man skriver static behøves det ikke å lage "new"
i main metoden.

Bare for Arrays: (utvidet) =

for hver noe på venstre
gjør det i parentesen på høyre.