

# Enkel oversikt over terminologi og kodesnutter i INF1000 uke 1 og 2

Gard Inge Rosvold

September 5, 2016

## Contents

<b>1</b>	<b>Uke01</b>	<b>1</b>
1.1	Template . . . . .	1
1.2	Skriv til terminal . . . . .	2
1.3	Variabler . . . . .	2
1.4	Scanner og innlesning fra terminal/bruker . . . . .	2
1.5	Uttrykkstest if/else if/else . . . . .	3
1.6	Type konvertering (String til int) . . . . .	3
1.7	void metoder uten parametre . . . . .	3
1.8	Avsluttende komplett eksempel uke01 . . . . .	4
<b>2</b>	<b>Uke02</b>	<b>5</b>
2.1	Metoder med parametre . . . . .	5
2.2	Utskrift.java som ble laget i datastuen mandag 5. september .	6

## 1 Uke01

### 1.1 Template

Template for filen TestKlasse.java

```
public class TestKlasse {  
    public static void main(String[] args) {  
        // din kode her  
    }  
}
```

```
}  
}
```

- public sikrer at du har samme navn på klassen som filnavnet
- TestKlasse er navnet på klassen – og skal alltid ha stor forbokstav og være lik filnavnet før filtype endelsen (.java)
- public static void main(String[] args) er den magiske metoden hvor programmet "starter". Legg merke til at den er indentert 4 spaces AKA 1 tab, og at den avsluttende krøllparentesen har samme indentering
- De to skråstrekene // er kommentartegn og vil kun vises i kildekodefila – veldig greit for å forklare en mindre selvforklarende kodedel

## 1.2 Skriv til terminal

For å skrive ut til terminal, bruk:

```
System.out.println("Det du vil skrive ut");
```

## 1.3 Variabler

- En variabel *deklarerer* med *type* slik: int myndighetsalder; eller String brukersNavn; der hhv. int og String er typer, mens myndighetsalder og brukersNavn er variabelnavn.
- En variabel får verdi gjennom *initiering*: myndighetsalder = 18; & brukersNavn = "Gard";. Dette kan skje samtidig som *deklarerer*: int myndighetsalder = 18; & String brukersNavn = Gard;.

## 1.4 Scanner og innlesning fra terminal/bruker

- For å lese inn fra bruker trenger du **Scanner**, og da er det tre ting som trengs:
  1. Man må legge til følgende linje *i toppen* av fila:

```
import java.util.Scanner;
```

2. Det må deklarereres og initieres en variabel av type `Scanner`, eksempel:  
`Scanner tastatur = new Scanner(System.in);`, kun variabelnavn (her `tastatur`) er valgfritt
3. Hente neste input/linje fra bruker som `String` med `tastatur.nextLine();`, eksempel:  
`String fraBruker = tastatur.nextLine();`  
(**OBS:** Husk å gi beskjed til bruker med `System.out.println("Beskjeden");` om hva som forventes at skal skrives inn.)

## 1.5 Uttrykkstest if/else if/else

Uttrykkstestene `if`, `else if` og `else` må komme i denne rekkefølgen, og delene i `if` og `else if` skjer kun hvis uttrykket de tester er `true` (det motsatte er `false`, som er de to eneste `boolean` typene). På tester er det veldig viktig og lurt å indentere riktig for å være sikker på at krøllparentesene er korrekt, så ikke en `else` plutselig havner "inni" en `if`-test. Et større eksempel er på slutten av dette dokumentet.

## 1.6 Type konvertering (String til int)

- Hvis du har en verdi av type `String`, men vil ha den til `int` må du benytte `Integer.parseInt(«strengen»)`.
- Det vanligste eksempelet er alder: `int alder = Integer.parseInt("19");`. I dette eksempelet er verdien ("19") gitt direkte, men den kunne også vært lagret i en variabel først:

```
String tekstAlder = "19";
int alder = Integer.parseInt(tekstAlder);
```

## 1.7 void metoder uten parametre

Void metoder uten parametre har formen:

```
public static void metodeNavn() { // metodenavn har alltid liten forbokstav
    //din kode her indentert et hakk
}
```

- Der man velger et passende selvvalgt metodeNavn som representerer hva metoden gjør.  
**Husk:** Alltid liten forbokstav på metodenavn og en indentering mellom krøllparentes start/slutt.
- Du kan ikke sette metoder inni andre metoder, men de kan være under eller over andre metoder - se eksempelkode.

## 1.8 Avsluttende komplett eksempel uke01

```
import java.util.Scanner;

public class TestKlasse {

    public static void main(String[] args) {
        System.out.println("Velkommen til INF1000");

        // Oppsett av variabler
        int brukersAlder;
        int myndighetsAlder = 18;
        String brukersNavn;
        Scanner tastatur = new Scanner(System.in);

        // Hent informasjon fra bruker
        // Foerst navn
        System.out.println("Skriv inn navnet ditt");
        brukersNavn = tastatur.nextLine();

        // Deretter alder
        System.out.println("Skriv inn din alder");
        String brukersAlderSomString = tastatur.nextLine();
        // Konverterer til typen 'int'
        brukersAlder =
        Integer.parseInt(brukersAlderSomString);

        // Skriver ut beskjed til bruker - navnet foerst og er
        likt uansett
        System.out.println("Hei " + brukersNavn + "!");
    }
}
```

```

        // Deretter om han er myndig eller ei, som er resultat
        av alderen bruker skrev
        if (brukersAlder < myndighetsAlder) {
            System.out.println("Med alder paa " +
brukersAlder + " er du nok ikke myndig (enda)");
        } else if (brukersAlder == myndighetsAlder) {
            System.out.println("Joess - du er akkurat
myndig i aar med " + brukersAlder + " som alder!");
        } else {
            // Kan regne med 'int' typene, men ikke med
String
            int aarSidenMyndig = brukersAlder -
myndighetsAlder;
            System.out.println("Det er " + aarSidenMyndig
+ " aar siden du var myndig" );
        }
    }

    public static void linjeskift() {
        // Metoden min 'linjeskift' kaller paa en tom println
        for aa kun faa et linjeskift
        System.out.println();
    }
}

```

## 2 Uke02

### 2.1 Metoder med parametre

- Alltid angi hvilken type (String, int, double) man forventer at hver parameter skal ha
- Husk å indenter riktig her også

- Etter at metoden er blitt "ferdig" vil variablene i hovedmetoden fortsatt ha samme verdi som før

## 2.2 Utskrift.java som ble laget i datastuen mandag 5. september

```
import java.util.Scanner;

public class Utskrift {

    public static void main(String[] args) {
        // Her kan man kommentere inn/ut den delen man vil
        kjoere
        //tekstTesten();
        tallTesten();
    }

    public static void tekstTesten() {
        // Foerst lage en Scanner ved variabelnavn tastatur
        for aa hente fra bruker
        Scanner tastatur = new Scanner(System.in);
        // Bruker utskrift metoden vi skrev lenger nede for aa
        skrive ut hva vi vil bruker skal skrive
        utskrift("Gi meg et navn");
        String navn = tastatur.nextLine();
        // Skriver ut en velkomst
        velkomst("Hei du", navn);
    }

    public static void tallTesten() {
        // Bruker utskrift metoden for aa be bruker om tall
        utskrift("Gi meg et tall");
        // Lager Scanner - merk at det er annerledes
        "rekkefoelge" fra over,
        // for det er ikke ved deklarerer/initialisering vi
        henter bruker, det er nextLine() delen.
        Scanner tastatur = new Scanner(System.in);
```

```

        // Henter tallet som tekst, eller String
        String somTekst = tastatur.nextLine();
        // Konverterer String til int med linjen under,
        parseInt(..) metoden tar 1 parameter av type String
        int somTall = Integer.parseInt(somTekst);
        // Spoer bruker hva den vil gjoer - bedre informasjon
        er oensket paa deres program(!)
        utskrift("Vil du gjoer 1, 2, eller 3");
        // Kan _gjenbruke_ variabel fra tidligere, siden
        programmet naa er ferdig med _verdien_ som var lagret
        // i somTekst variabelen.
        // Deklarerer den ikke paa nytt, gir den bare en ny
        verdi - som denne gangen er hva brukeren vil "gjoer"
        somTekst = tastatur.nextLine();
        if (somTekst.equals("1")) {
            // 1 er oek med fem
            oekMedFem(somTall);
        } else if (somTekst.equals("2")) {
            // 2 er oek med 10
            oekMedTi(somTall);
        } else if (somTekst.equals("3")) {
            // 3 er trekk fra 10
            System.out.println("Tallet foer metoden: " +
somTall);
            trekkFraTi(somTall);
            System.out.println("Tallet etter metoden: " +
somTall); // Vil vaere det samme tallet
        } else {
            // Alt annet er ugyldig, og brukeren kan faa
            beskjed om det
            // == OBS == dette er nytt etter timen
            utskrift("Ugyldig kommando");
        }
    }

    public static void utskrift(String tekst) {
        // Denne metoden skriver ut String objektet 'tekst'
        som kommer med som parameter
    }

```

```

        // De som var igjen etter timen husker at tekst kan
        vaere hva som helst, vi kalte den
        // Ogsaa for alpha
        System.out.println(alpha);
    }

    public static void velkomst(String velkomstSetning, String
        navn) {
        // Skriver ut en velkomstsetning med navnet, legg
        merke til at det er 2 krevde argumenter/parametre
        System.out.println( velkomstSetning + " " + navn);
    }

    public static void oekMedTi(int somTall) {
        System.out.println(somTall);
    }

    public static void oekMedFem(int somTall) {
        System.out.println(somTall+5);
    }

    public static void trekkFraTi(int somTall) {
        // Utvidet her for aa vise at selv om variabelen
        endres i metoden, endres den ikke i
        // de opprinnelige metoden
        somTall = somTall - 10;
        System.out.println(somTall);
    }

}

```