

# INF1000 - Obligatorisk innlevering 2

Temaer denne uka: *Feilmeldinger, kontrollflyt og void-metoder.*

## Oppgave 2.1)

**Tema:** *Void-metoder.*

**Filnavn:** `Metoder.java`

- a) Lag et program som kommuniserer med brukeren slik at det tar inn et navn og et bosted fra terminalen. Eksempel på hvordan en kjøring av programmet skal se ut:

```
Skriv inn navn:  
Espen Askeladd  
Skriv inn bosted:  
Oslo  
Hei, Espen Askeladd! Du er fra Oslo.
```

- b) Legg koden som leser inn informasjon og skriver ut en hilsen i en egen metode. Kall denne metoden 3 ganger slik at du får lest inn og skrevet ut informasjon om 3 personer. Kjøringen av programmet bør ligne på den i oppgave a).

Synes du denne oppgaven var vanskelig? [Se øvingsoppgaver 2.1.1.](#)

## Oppgave 2.2)

**Tema:** *Vise kodeflyt.*

**Filnavn:** `Kodeflyt`

Dette er en penn- og papiroppgave. Oppgaven skal scannes inn eller tas bilde av og leveres på vanlig måte i Devilry.

Forklar programflyten til følgende program. Vi antar at brukeren først taster 1 og deretter 2. Tegn flyten i programmet med piler og ta utgangspunkt i forelesningsfoilene.

```

import java.util.Scanner;

public class PersonRegister {
    public static void main(String[] args) {
        Scanner innleser = new Scanner(System.in);
        String navn = "Ikke satt";
        String bosted = "Ikke satt";
        int valg = 0;
        printMeny();
        valg = Integer.parseInt(innleser.nextLine());
        if (valg == 1) {
            registrerPerson();
            navn = innleser.nextLine();
            bosted = innleser.nextLine();
        }
        if (valg == 2) {
            printPerson(navn, bosted);
        }
        printMeny();
        valg = Integer.parseInt(innleser.nextLine());
        if (valg == 1) {
            registrerPerson();
            navn = innleser.nextLine();
            bosted = innleser.nextLine();
        }
        if (valg == 2) {
            printPerson(navn, bosted);
        }
    }

    // metode for aa gjore valg fra brukeren
    static void printMeny() {
        System.out.println("1. Registrer en person");
        System.out.println("2. Print personen");
    }

    // metode for aa skrive ut beskjed om aa registrere en person
    static void registrerPerson() {
        System.out.println("Skriv inn navnet, trykk enter, og skriv saa inn bostedet og trykk
    }

    // metode for å printe en person
    static void printPerson(String navnForUskrift, String bostedForUskrift) {
        System.out.println("Du har registrert personen: " + navnForUskrift + " fra " + bostedFo
    }
}

```

Synes du denne oppgaven var vanskelig? [Se øvingsoppgaver 2.2.1.](#)

### Oppgave 2.3)

**Tema:** *Metoder med parametre*

**Filnavn:** EnkelKalkulator.java

Vi skal nå lage en enkel kalkulator som skal ha funksjonalitetene addere, subtrahere og multiplisere.

- a) Lag en metode som tar i mot to heltall som parametre og printer summen av de to tallene.
- b) Lag en metode som tar i mot to heltall som parametre og printer differansen mellom de to tallene.
- c) Lag en metode som tar i mot to heltall som parametre og printer produktet av de to tallene.
- d) Lag et program som leser inn to heltall fra brukeren og som deretter kaller alle de tre metodene slik at brukeren får vite summen, differansen og produktet.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 2.3.1.

Synes du denne oppgaven var lett? Se utfordringsoppgaver 2.3.3 og 2.3.4.

### Oppgave 2.4)

Denne oppgaven skal leveres som en .txt-fil hvor du svarer på spørsmålene under

**Tema:** *Feilmeldinger.*

**Filnavn:** Feilmeldinger.txt

- a) Forklar forskjellen mellom kjørefeil og kompileringsfeil.

b)

```
Class Feilmeldinger{
    public static void main(String[] args) {
        system.out.println("Velkommen til feilmeldinger!")
    }
}
```

Koden over gir følgende feilmeldinger når vi prøver å kompilere programmet:

```
$ javac Feilmeldinger.java
Feilmeldinger.java:1: error: class, interface, or enum expected
Class Feilmeldinger{
^
Feilmeldinger.java:2: error: class, interface, or enum expected
    public static void main(String[] args) {
                        ^
2 errors
```

1. Forklar hvordan du vil gå fram for å rette feilen.

2. Kompileringen sier at det er to feil, men i virkeligheten er den andre feilmeldingen ikke en feilmelding. Forklar hvorfor.

Etter å ha rettet feilen over får vi en ny feilmelding.

3. Forklar fremgangsmåten for å finne feilen og hva som er feil.

```
$ javac Feilmeldinger.java
Feilmeldinger.java:3: error: ';' expected
    system.out.println("Velkommen til feilmeldinger!")
                        ^
1 error
```

Etter å ha rettet denne feilen får vi nok en feilmelding.

```
$ javac Feilmeldinger.java
Feilmeldinger.java:3: error: package system does not exist
    system.out.println("Velkommen til feilmeldinger!");
                        ^
1 error
```

4. Forklar fremgangsmåten for å finne feilen og hva som er feil.

Synes du denne oppgaven var vanskelig? [Se øvingsoppgaver 2.4.1, 2.4.2 og 2.4.3.](#)

## Oppgave 2.5)

**Tema:** *Egen oppgave.*

**Filnavn:** MinOppgave2.java

Lag en egen oppgave som handler om metoder. Både oppgaveteksten og besvarelsen skal leveres inn.

## Frengangsmåte for innleveringer i INF1000

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
  - Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
  - Hvor lang tid (ca) brukte du på innleveringen?
  - Var det noen oppgaver du ikke fikk til? Hvis ja:
    - Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
    - Hvorfor tror du at oppgaven ikke fungerer?
    - Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på [Devilry](#).
3. Lever de 5 .java-filene samt README.txt i *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

Den obligatoriske innleveringen er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken [her](#) og flere utfordringsoppgaver [her](#).