

## DM502 : De første øvelser

Øvelserne i de første to uger afholdes i instituttets terminalrum (lige over U49) og omhandler introduktion til Linux, text-editor, Java og debuggeren JSwat.

Vi forudsætter at du allerede har en IMADA konto. Hvis ikke, kontakt Peter Schneider-Kamp eller en af instruktorene så hurtigt som muligt.

### 1. Introduktion til IMADAs system

Instruktoren vil give en introduktion til IMADAs system, der består af computere udstyret med Ubuntu Linux. Introduktionen vil indeholde punkter som:

- Login
- Opsætning af forwardet email
- Emacs (eller anden editor)
- Shell, inklusiv de vigtigste kommandoer
- Introduktion til KDE/Gnome
- Kompilering og afvikling af Java-programmer
- Debuggeren JSwat (husk: `javac -g filnavn.java`)
- Start af browser
- Udskrift

Det er helt legalt at bruge lidt tid på lege rundt i systemet så man kan begynde at føle sig "hjemme". I det omfang der er tid til det, er det også til disse øvelser I skal spørge om alt det I ikke kan finde ud af på systemet. Efter de første øvelser forventes det at I kan klare jer på egen hånd i terminalrummet. Skulle I senere have et problem eller spørgsmål ang. brugen af computerne, så er der oftest en ældre studerende i terminalrummet man kan spørge (de svarer som regel gerne på spørgsmål).

## 2. Hello World

Lav opgave 2, 3 og 4 i ”Noter og opgaver”.

## 3. Brug af debuggeren JSwat

Indtast følgende Java-program i en fil `Gentag.java`. Bemærk at det er et krav at filnavnet svarer til klassenavnet (klassenavnet er det der står efter `public class`, i dette tilfælde `Gentag`):

```
public class Gentag {
    public static void main( String [] args ) {
        int antal = 11;

        int taeller = 0;
        while ( taeller < antal ) {
            System.out.println(" Hello World");
            taeller = taeller + 1;
        }
    }
}
```

Hvad forventer du der sker når programmet kører?

Oversæt programmet med kommandoen: `javac -g Gentag.java`

Kør programmet med kommandoen: `java Gentag`

Sker der det du forventer?

Nu skal du prøve at afvikle programmet en linie af gangen, og følge med i værdierne af programmets variable. Dette kan gøres med debuggeren JSwat. JSwat startes med kommandoen `jswat`.

Start JSwat.

Debuggeren vil - hvis du ikke har brugt den før - være ”tom”. Ellers vil den returnere til den tilstand du forlod den i sidste gang du brugte den. Vi skal først indlæse programteksten i debuggeren. Brug menuen **File** → **Open File...** Find filen `Gentag.java` og tryk **Ok**. Programteksten skulle nu gerne være at se i JSwat vinduet.

For at lære JSwat lidt bedre at kende, lader du først programmet køre ”normalt” — dvs. uden stop undervejs. Dette gøres ved at bruge menuen **Session** → **Start...**, den tilsvarende knap (grøn pil) eller tast **F5**. I vinduet, der dukker op, er du, indtil videre, interesseret i tekstfeltet **Class Name**, afkrydsningsfeltet **Resume immediately** samt fanebladet **Class Path**. I tekstfeltet skrives `Gentag`. Afkrydsningsfeltet *skal* være afkrydset. Skift til fanebladet **Class Path**. Tryk **Add Jar/Folder**. Find og marker folderen, der indeholder `Gentag.class`. Tryk **Open**. Kontroller at folderen er tilføjet i listen. Tryk **Ok**.

Hold øje med den nederste venstre del af JSwat vinduet. I fanebladet **Process Output** (aktivér det) ses programmets output. Dette skulle gerne være identisk med det output du så da programmet kørte i terminal-vinduet.

Herefter vil vi prøve at køre programmet en linie af gangen. Først skal du sætte et såkaldt "breakpoint". Klik én gang i margenen til venstre for linien der indeholder `int antal = 11;`, således at linien er markeret med en rød firkant. Du har nu fortalt debuggeren at programmet skal sættes i pause-tilstand når det når til denne linie. Start en session som før. Du skulle nu gerne observere at linien bliver markeret grøn, og at programmet ikke kører videre. Brug nu knappen **Step Over** eller tast **F8** til at køre programmet en linie af gangen. Læg mærke til hvordan programmet gentager indholdet af `while`-løkken. Nederst til højre i JSwat vinduet findes fanebladet **Variables**. Aktiver dette, tryk **F8** flere gange, og observer hvorledes variabelen `taeller` ændrer værdi undervejs. Når du bliver træt af dette kan du trykke på knappen **Continue**, og programmet vil køre færdig uden at stoppe.

Leg og eksperimenter gerne videre med debuggerens andre muligheder. Prøv f.eks. at sætte et "breakpoint" i en af linierne inde i `while`-løkken, starte programmet op og bruge **Continue** knappen et antal gange. Hvad sker der?

#### 4. Middelveerdi.java

Indtast følgende Java-program i en fil `Middelveerdi.java`.

```
import java.util.Scanner;

public class Middelveerdi {
    public static void main( String [] args ) {
        int antal, tal, sum;
        double resultat;
        Scanner tastatur = new Scanner( System.in );

        antal = 0;
        sum = 0;

        System.out.print( "Indtast et nyt tal: " );
        tal = tastatur.nextInt();
        while( tal > 0 ) {
            sum = sum + tal;
            antal = antal + 1;
            System.out.print( "Indtast et nyt tal: " );
            tal = tastatur.nextInt();
        }

        resultat = (double) sum / antal;
        System.out.println( "Middelveerdi: " + resultat );
    }
}
```

Oversæt (`javac Middelveerdi.java`) og kør (`java Middelveerdi`) programmet.

Anvend debuggeren JSwat på programmet og følg en kørsel af programmet trin for trin. Husk at for at anvende JSwat skal programmet være oversat med `-g` option (`javac -g Middelvaerdi.java`).

Derefter skal følgende spørgsmål overvejes:

- Hvad kan programmet?
- Hvilke mangler har programmet?

Forsøg derefter at forbedre programmet ud fra de mangler du har identificeret. Husk at hver gang programmet rettes skal det oversættes igen før det køres.