




Seminar 1

python → Java



Hvem er vi ?

Marlen Jarholt

- Går første året på master i Programmering og systemarkitektur
- Er gruppelærer i IN1010 for tredje gang
 - Har også vært gruppelærer i IN1000
 - Til sammen 5 semester som gruppelærer
- Kan kontaktes på mattermost: marleja eller på mail: marleja@ifi.uio.no
- Funfact: Jeg over gjennomsnitt glad i å pusle puslespill

Johanna Haarseth

- Går første året på master i Programmering og systemarkitektur
- Er gruppelærer for fjerde gang
 - in1020, in1030 og in1000 tidligere
- Kan kontaktes på mattermost: johannph eller på mail: johannph@ifi.uio.no
- Funfact: Elsker alt av reality fra *Der ingen skulle tru at nokon kunne bu* til *KUWTK*

Kort: Praktisk informasjon

- Seminar vil inneholde en kort repitisjon av ukensstoff + oppgaver
 - Oppgavene vil bli lagt ut på forhånd på [semestersiden](#) (semesterssiden → Grupper → oppgaver gruppetimer)
 - All koden som blir koden i timen + løsningsforslag til oppgaven legges ut [her](#) (semesterssiden → Grupper → Gruppe 3)
- Lab tilbudet: 10 minutter med en gruppelærer en til en.
 - Mer info om det på [semestersiden](#).
 - Lag drive (G Suite) bruker allerede nå :D
- Mattermost
- Emnesiden

Kontrakt med hverandre

- Se på forelesningen før du kommer (aka vær forberedt)
 - Da vil du få mest ut av timen
- Hver aktive, det er det dere lærer mest av
- Alle spørsmål er gode spørsmål, lurer du på det er det garantert en til som lurer på det samme
 - Still gjerne spørsmål i chat (hvis du vil være anonym kan du sende oss en melding direkte), eller rekk opp hånda, så får du ordet :D
- Delta aktivt i samarbeids oppgavene man lærer masse av hverandre, kanskje får man også et nytt bekjentskap
- Vi skal gjøre hverandre bedre :)

Bli kjent med hverandre

- Start med å finne en blå gjenstand i rommet
- Når du presenterer deg selv skal du:
 - Vise den blå gjenstanden din
 - Navnet ditt
 - Linjen du går på
 - Og gjerne en funfact

Tips og triks

- Jobb mye de første ukene med å bare lære deg java, du vil takke deg selv siden :D
- Følg progresjonen i forelesningene, det går fort i IN1010
- Gjør masse oppgaver, bruk triks
- Diskuter med andre som tar faget
 - Delta på gruppearbeid, f.eks. i gruppetime
- Les feilmeldingene dine
 - Hvis du ikke forstår dem kan du google dem! Google er din venn ❤️
- Bruk gruppeundervisning og lab
- Begynn med obliquen med en gang den kommer ut

Send Johanna en direktemelding!

Marlen skal først ta litt repetisjon først og så gjør vi oppgaver etterpå!
Send Johanna en direktemelding med svar på spørsmålene under

1. Er du komfortabel med å gjøre noen oppgaver sammen med noen andre i zoom?
Deler dere inn etter hvordan dere sier dere ligger ann!
2. Har du sett gårsdagens forelesning?
3. Hvordan går java?
4. Hvor godt føler du at du forstår stoffet fra uke 1 på en skala fra 1(lite) – 6(veldig godt)?
5. Noen spesielle du gjerne vil samarbeide med?

Java: Kort om syntaks

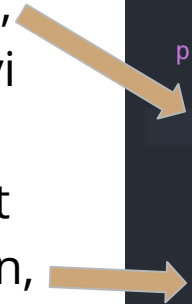
- I Java må vi ha semikolon (;) etter hver linje med kode
- For å dele inn i kodeblokker bruker vi krøllparenteser ({ })
- I if-tester og while og for-løkken må vi bruke parenteser (()) rundt uttrykkene

```
class A {  
  
    public static void main(String[] args) {  
  
        if (2 == 3) {  
            System.out.println("TRUE");  
        } else {  
            System.out.println("FALSE");  
        }  
    }  
}
```


Java variabler og if-tester

- Variablene må vite hvilken type (int, double, String osv) variabelen når vi definerer den
- If-tester er nesten helt like (bortsett fra syntaks endringen) som i python, men i stedet for elif bruker vi else if

```
class A {  
  
    public static void main(String[] args) {  
  
        int tall = 7;  
  
        if (tall == 3) {  
            System.out.println("Lik tre");  
        } else if (tall > 3) {  
            System.out.println("Storre enn tre");  
        } else {  
            System.out.println("Mindre enn tre");  
        }  
  
    }  
  
}
```



Java for- og while-løkker

- For-løkker

- Enten med teller
- Eller for hvert element i f.eks. en liste (for each løkke)
- Vi bruker for-løkke når vi vet hvor mange ganger noe skal skje

```
1  for (int i = 0; i <= 10; i = i + 2) {
2      System.out.println(i);
3  }
4  String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
5  for (String i : cars) {
6      System.out.println(i);
7  }
```

- While-løkke

- Vi bruker while-løkke når vi ikke vet hvor mange ganger noe skal skje
- For eksempel hvis det avhenger av input
 - Lager et nytt Scanner objekt for å ta input fra bruker (Husk import!)

```
15  Scanner myScanner = new Scanner(System.in);
16  System.out.println("Gjett navnet mitt");
17  String inp = myScanner.nextLine();
18  while (!inp.equals("Kari")){
19      System.out.println("Feil");
20      inp = myScanner.nextLine();
21  }
22  System.out.println("Riktig");
```

Array og HashMap (ordbøker)

- Når vi lager liste (array) i Java må vi først si hva skal putte i den, f.eks. String eller int
 - Man kan ikke endre lengden på en array, så f.eks. myNum kan ikke ha flere enn fire elementer. Statisk
- For ordbøker (heter HashMap) må man også si hva man skal putte i dem (f.eks. int eller String)
 - Husk import!
 - hashmap.put(key, value)
 - hashmap.get(key)

```
1 String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
2 int[] myNum = {10, 20, 30, 40};
3 System.out.println(cars[0]); // >> "Volvo"
```

```
1 // Husk å importere!
2 import java.util.HashMap;
3
4 public class Main {
5     public static void main(String[] args) {
6         // Lager et HashMap object som jeg kaller capitalCities
7         HashMap<String, String> capitalCities = new HashMap<String, String>();
8
9         // Legger til key og value (Land, By)
10        capitalCities.put("England", "London");
11        capitalCities.put("Germany", "Berlin");
12        capitalCities.put("Norway", "Oslo");
13        System.out.println(capitalCities.get("England"));
14    }
15 }
```

Java klasser og objekter

- Hvis klassen har en konstruktør må konstruktøren ha AKKURAT de samme navnet som klassen
- Hovedprogrammet skal alltid ha en main metode
 - Det er denne som blir kjørt når man kjører koden
- En *Static* metode er en metode som "eies" av klassen og ikke til et objekt
 - Derfor kan man kalle på static metoder uten å ha lagd en instans av klassen
- Public/private
 - Når noe er public blir det en del av interfacet til objektet eller klassen
 - Når noe er private er det kun objektet som kan bruke variablene / metodene

```
class Katt {  
    private String navn;  
    private int alder;  
  
    public Katt (String navn, int alder){  
        this.navn = navn;  
        this.alder = alder;  
    }  
  
    public static String hentEnFunFactOmKatter(){  
        return "Katter svetter gjennom potene";  
    }  
  
    private void ookAlder(){  
        this.alder += 1;  
    }  
  
    public void haBursdag(){  
        ookAlder();  
    }  
  
    public void skrivUtInfo(){  
        System.out.println("Navn: " + navn);  
        System.out.println("Alder: " + alder);  
    }  
}  
  
class Hovedprogram{  
    public static void main(String[] args) {  
        String funFact = Katt.hentEnFunFactOmKatter();  
        System.out.println(funFact);  
  
        Katt pus = new Katt("Pus", 3);  
        pus.haBursdag();  
        pus.skrivUtInfo();  
  
        pus.ookAlder();  
    }  
}
```

!!!kke lov!!

Java kompilering og kjøring

- Man må alltid kompilere filene før man kjører dem
 - For å kompilere bruker man `javac *.java`
- For å kjøre et java program skriver du følgende: `java <navn på klassen>`
 - class filen blir lagd under kompilering

Lagre --> kompiler --> kjør

```

def hovedprogram():
    a = 3
    b = 3.3
    c = 'Hei' # Ingen forskjell på " og '
    d = [1, 2, 3] # liste
    e = [a, b, c] # liste
    f = {'a': 3, 'b': 4}

    for i in range(10):
        print("Tallet er", i)

    for j in d:
        print(j)

    while a < 10:
        print(a)
        a += 1

    inp = input("Hva heter du? ")
    int_inp = int(inp)
    float_inp = float(inp)

hovedprogram()

class A:

    # Konstruktør
    def __init__(self, arg1, arg2):
        self._instansvar1 = arg1
        self._instansvar2 = arg2

    def foo(self, bar):
        return bar

```

Java vs Python

```

import java.util.Scanner;

class Hovedprogram {
    public static void main(String[] args) {
        int a = 3;
        double b = 3.3;
        String c = "Hei";
        char d = 'd';
        int[] e = {1, 2, 3};

        for (int i = 0; i < 10; i++) {
            System.out.println("Tallet er " + i);
        }
        for (int j : e) {
            System.out.println(j);
        }
        while (a < 10) {
            System.out.println(a);
            a++;
        }

        Scanner scanner = new Scanner(System.in);
        String inp = scanner.next();
        int int_inp = Integer.parseInt(inp);
        double double_inp = Double.parseDouble(inp);
    }
}

class A {
    private String instansvar1;
    private int instansvar2;

    public A(String arg1, int arg2) {
        instansvar1 = arg1;
        instansvar2 = arg2;
    }

    public String foo(String bar) {
        return bar;
    }
}

```

```

def hovedprogram():
    a = 3
    b = 3.3
    c = 'Hei' # Ingen forskjell på " og '
    d = [1, 2, 3] # liste
    e = [a, b, c] # liste
    f = {'a': 3, 'b': 4}

    for i in range(10):
        print("Tallet er", i)

    for j in d:
        print(i)

    while a < 4:
        print(a)
        a += 1;

    inp = input("Hva heter du? ")
    int_inp = int(inp)
    float_inp = float(inp)

hovedprogram()

class A:

    # Konstruktør
    def __init__(self, arg1, arg2):
        self._instansvar1 = arg1
        self._instansvar2 = arg2

    def foo(self, bar):
        return bar

```

```

import java.util.Scanner;

class Hovedprogram {
    public static void main(String[] args) {
        int a = 3;
        double b = 3.3;
        String c = "Hei";
        char d = 'd';
        int[] e = {1, 2, 3};

        for (int i = 0; i < 10; i++) {
            System.out.println("Tallet er " + i);
        }
        for (int j : e) {
            System.out.println(j);
        }

        while (a < 4) {
            System.out.println(a);
            a++;
        }

        Scanner scanner = new Scanner(System.in);
        String inp = scanner.next();
        int int_inp = Integer.parseInt(inp);
        double double_inp = Double.parseDouble(inp);
    }
}

class A {
    private String instansvar1;
    private int instansvar2;

    public A(String arg1, int arg2) {
        instansvar1 = arg1;
        instansvar2 = arg2;
    }

    public String foo(String bar) {
        return bar;
    }
}

```

Breakoutrooms

1. Slå på kamera og ha en presentasjonsrunde
2. Diskuter disse spørsmålene:
 - a. Hvordan går Java-programmeringen så langt?
 - b. Hvilke forventninger har du til in1010?
 - c. Hvordan er det å starte semesteret hjemmefra?
3. Jobb sammen med oppgavene dere finner på mattermost
 - a. Enten ved at en deler skjerm
 - b. Eller ved å bruke www.gus.jp (velg Java øverst til venstre)
 - c. Eller jdoodle.com
4. Bruk “ask for help”-knappen for å få hjelp 😊
5. Vi møtes her igjen for å gå gjennom oppgavene til slutt (dere bestemmer hvilke)

Ris, ros, forslag ?

<https://nettskjema.no/a/180345>