

Fra Python til Java

Introduksjon til et nytt programmeringsspråk

Uke 1 (IN1010-tidsregning)

Forelesning 1 av 3 i uke 1 (mandag 11.1.21)

Time 2

Oversikt

- Introduksjon – om programmeringsspråk
- Python \Rightarrow Java
 - Hvordan skrive og kjøre et Java-program?
 - Noe er likt
 - Noe bare ser annerledes ut
 - Noe er helt forskjellig
- Kom i gang
- I morgen: Klasser ved Stein Gjessing
- Neste uke: 2 * 2 timer forelesning om Java

Studenters digitale spor i utvalgte IFI emner ved UiO

3 årig samarbeidsprosjekt mellom IFI, IPED, og USIT.

- Formål: Undersøke faktorer som bidrar til at studenter lykkes i utvalgte emner ved IFI.
- Analysere loggdata fra digitale studiesystemer som IFI bruker.
 - Devilry, Trix, nettkurs i logiske metoder, Vortex, Inspira og FS med mer.
 - Ingen sensitive opplysninger vil bli samlet inn
- Strategisk viktig for IFI og MN/UiO for å forbedre studietilbudet.
- Deltakelse er helt frivillig
 - Kan trekke dere når dere vil
 - Ingen negative konsekvenser
 - Faglærere får ikke tilgang til data på individnivå
- Samtykkeskjema vil bli sendt ut i nærmeste fremtid.

Hvor mange programmeringsspråk finnes det?

700?

900?

- Python
- Java
- bash
- Excel
- SQL
-
- Nye behov oppstår – anvendelser, omgivelser, verktøy
- Vi kan lage bedre språk i dag enn for 50 år siden
- Ulike preferanser

Hvorfor ble Python laget?

Python ble opprinnelig laget i 1989 av nederlenderen Guido van Rossum med filosofien:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Hvorfor ble Java laget?

Java ble laget i 1996 av James Gosling og andre i SUN Microsystems (nå Oracle) med mål:

- It must be “simple, object-oriented, and familiar”. It must be “robust and secure”.
- It must be “architecture-neutral and portable”. It must execute with “high performance”.
- It must be “interpreted, threaded, and dynamic”.

Typisk bruk av de to språkene

Python er godt egnet til

- nybegynneropplæring
- korte, enkle programmer du trenger nå
- kopling mot nyttige programpakker i kjemi, fysikk, matematikk, . . .

Java er godt egnet til

- objektorientert programmering
- store programmer
- trygge programmer
- raske programmer

Fra Python til Java

Gitt følgende Python program – hvordan skriver (og kjører) vi samme program i Java?

```
print("Dette er et Python-program")  
tall = 2+3  
print("Dette er et tall: " + str(tall))
```


(Nesten) helt likt

Uttrykk og tilordning

```
# Python  
lengde * bredde  
2 * 3.1416 * r
```

```
// Java  
lengde * bredde  
2 * 3.1416 * r
```

```
# Python  
areal = lengde * bredde  
omkrets = 2 * 3.1416 * r
```

```
// Java  
areal = lengde * bredde;  
omkrets = 2 * 3.1416 * r;
```

Egentlig helt likt

Blokker og programsetninger

- I Java brukes krøllparenteser ({ }) i stedet for innrykk.
- Alle Java-setninger avsluttes med semikolon (;).

```
# Python
if lengde < 0:
    lengde = -lengde
```

```
// Java
if (lengde < 0) {
    lengde = -lengde;
}
```

- Linjeskift og innrykk betyr altså ikke noe i et Java-program:

```
// Java
if (
    bredde < 0)    {bredde
= -bredde; }
```

if-tester kan ha flere alternativer

```
# Python
if x < 0:
    f = -1
elif x == 0:
    f = 0
else:
    f = 1
```

```
// Java
if (x < 0) {
    f = -1;
} else if (x == 0) {
    f = 0;
} else {
    f = 1;
}
```

Legg merke til:

- Java har parenteser rundt if-testen (i stedet for kolon etter)
- Java har ingen **elif** – i stedet er det mulig å skrive **else if**

Løkker

While-løkker

```
# Python
while x <= 255:
    total = total + x
    x = 2 * x
```

```
// Java
while (x <= 255) {
    total = total + x;
    x = 2 * x;
}
```

For-løkker med teller


```
# Python
for i in range(10):
    sum = sum + i
```

```
// Java
for (int i = 0; i < 10; i++) {
    sum = sum + i;
}
```

Utskrift

I java finnes to metoder for utskrift:

- `System.out.print` skriver ut parameteren.
 - Hvis det er flere elementer skjøtes de sammen med +
 - Eller man kan kalle `System.out.print` flere ganger
- `System.out.println` gjør det samme, men legger til linjeskift.



samme resultat!

```
System.out.print("Navn: " + "Donald Duck");  
System.out.println();
```

```
System.out.println("Navn: " + "Donald Duck");
```

Helt forskjellig

Angivelse av variabeltype

- Python har *dynamisk typing* der typen er knyttet til verdiene ; derfor kan en variabel tilordnes verdier av en hvilken som helst type.
- Java har *statisk typing* der både variablene og verdiene har type; variabler kan bare få verdier av den riktige typen.
- Derfor: *I Java må alle variabler deklarereres.*

```
// Java
int lengde = 3, bredde = 2, areal;
double omkrets, r = 3;
areal = lengde * bredde;
omkrets = 2 * 3.1416 * r;
```

Javas viktigste typer

int	heltall (dvs 0, 255, -3, ...)	
double	flyttall (dvs 0.0, 1.5, -22.873, ..)	
boolean	logisk verdi (dvs true , false)	
char	enkeltegn (dvs ' 1 ', ' a ', ' * ', ...)	NB: enkel apostrof (fnutt)

Dessuten finnes klassen

String	tekst (dvs " abcd ", "", " \$50 ", " 146 ", ..)	NB: dobbel apostrof
---------------	--	---------------------

Innpakning

- I Java må *alt* ligge i klasser
- Utføring av programmet starter alltid i en metode som heter **main** – deklarereres slik:

```
class Hei {  
    public static void main(String[] arg) {  
        System.out.println("Hei!");  
    }  
}
```

- Filen må ha samme navn som klassen, her **Hei.java**

Hvor mange feil finner du?

Feil.java

```
class Feil {
    public static void main (String[] args) {
        pris = 25;
        pris = 2*pris
        if pris > 200 {
            pris = 200;
        }
        else:
            pris = pris + 50;
        print("Pris: " + pris);
    }
}
```

Kjøring av Java-programmer

- Du må installere Java Development Kit (JDK)
- Du trenger to programmer for å kjøre et Java-program
 - **javac** oversetter (kompilerer) .java-filen til en eller flere .class-filer
 - **java** utfører .class-filen(e)

```
$ javac Hei.java
$ java Hei
Hei!
$
```

```
class Hei {
public static void main(String[] arg) {
    System.out.println("Hei!");
}
```

Et komplett lite eksempel

FeilUtenFeil.java

```
class FeilUtenFeil {
    public static void main (String[] args) {
        int pris = 25;
        pris = 2*pris;
        if (pris > 200) {
            pris = 200;
        }
        else
            pris = pris + 50;
        System.out.println("Pris: " + pris);
    }
}
```

*Dette fungerer fordi det bare er én setning som hører til else-grenen – da **trenger** man ikke ha krøllparenteser rundt den. Men det er mer oversiktlig (og sikrere i forhold til senere utvidelser) om man alltid bruker krøllparenteser*

\$ javac FeilUtenFeil.java

\$ java FeilUtenFeil

Komme i gang

- Viktige lenker fra semestersiden (under Emneressurser):
 - Java for menneskelige lesere (inkl deg selv) – Kodestil i Java
 - Installasjon og oppsett av Java
 - Oppkobling til og arbeid på Ifis servere
- Mer om programmering i Java denne uken og neste
 - Klasser og objektorientert programmering i morgen
 - Arrayer, metoder, input fra terminal og filhåndtering neste mandag
 - Mer om datastrukturer og representasjon neste tirsdag
 - Trening og flere detaljer i seminar og oppgaver inkl *Innlevering 0* (frist 18.1)!

Oppsummering

- Det finnes ikke noe beste språk for alle formål!
- Det er en styrke å kunne flere språk
- Alle informatikere må regne med å lære flere språk
- Flere språk gir dypere forståelse

- Det er egentlig bare én måte å lære et nytt programmeringsspråk på:
 - les dokumentasjonen (lærebok, notater, Java 8 API)
 - prøv selv med mange eksempler
 - om ikke alt fungerer som forventet, gjenta