
Java Programmering 0: Installasjon av JAVA utviklingsmiljø

DAT100 – Grunnleggende Programmering H2017

Institutt for data- og realfag

Dere skal legge inn JAVA utviklingsmiljø på egen PC som dere får bruk for i faget DAT100. Undervisning i DAT100 baserer seg på bruk av egen PC. Det er derfor **viktig** at alle får dette til.

Programvaren som skal installeres er **Java Development Kit (JDK)** og **Eclipse Integrated Development Environment (Eclipse IDE)**. Programvaren kan lastes ned fra nett og JDK må installeres før Eclipse installeres.

Informasjon om tilkopling til trådløst nett (eduroam) finnes i en separat veiledning.

Lenker for nedlastning i dette dokumentet er også tilgjengelig via <http://dat100hib.github.io/H2017>

Denne veiledningen finner du også i mappen **Programmeringslab** i faget DAT100 på **its learning**.

Du må gjøre tre ting for å få denne første programmeringsoppgaven godkjent:

1. Demonstrere den kjørende applikasjonen i oppgave 1 og oppgave 4 til lab-assistent eller faglærer. Dette kan gjøres på programmerings-laben, ifm. forelesning, eller ved å gjøre en individuell avtale.
2. Levere inn filen `Velkommen.java` via its learning for faget. Filen finnes i mappen `Velkommen/src/no/hvl/dat100` i arbeidsområdet (workspace) for Eclipse.
3. Levere inn Eclipse-prosjektet i oppgave 3 på oppbevaringsplassen på github classroom.

Dette for å sikre at alle har et fungerende JAVA utviklingsmiljø:

Deadline for innlevering/godkjenning av øvingen er 1. september.

Ta kontakt med faglærer eller lab-assistent dersom du har problemer med øvingen. I mappen Java utviklingsmiljø i faget DAT100 på its learning er der videre et forum for spørsmål og svar relatert til installasjon av programvaren.

Nedlastning av filer fra nett

JDK og Eclipse til Windows og Linux kommer i to versjoner: en for 64-bit operativsystem og en for 32-bit operativsystem. De fleste har et 64-bit operativsystem:

- På **Windows** kan du bruke **Control Panel (Kontroll Panel)** → **System** for å se om du har et 32 eller 64-bit system.
- På **Linux** kan du bruke kommandoen `uname -m`. Hvis output inneholder `x86_64` da har du et 64-bit system og ellers et 32-bit system.

JDK: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Velg i586 varianten for 32-bit operativsystem og x64 for 64-bit operativsystem. For RPM-baserte Linux distribusjoner kan `.rpm` filen velges – ellers velg `tar.gz` filen.

JDK og JAVA dokumentasjon:

<http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html>

Eclipse: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/oxygen>

Velg Eclipse (Oxygen 4.7) IDE for Java Developers. Husk å velge rett operativsystem (Windows, Mac, Linux) samt 32-bit/64-bit versjon (se informasjon 32- eller 64-bit operativsystem ovenfor).

Installasjon av Java Development Kit (JDK)

Detaljerte instruksjoner for installasjon av JDK finnes via:

http://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html

Windows:

1. Dobbeltklikk på programmet `jdk-8u144-windows-x64.exe` (for 64-bit system) eller `jdk-8u144-windows-i586.exe` (for 32-bit system).
2. Følg instruksjonene. Foreslåtte valg vil vanligvis fungere fint.

Linux

- Installasjon ved bruk av `tar.gz` filen:
 1. Kopier `tar.gz` filen til den mappen hvor du ønsker å installere JDK.
 2. Pakk ut `tar.gz` filen med kommandoen
`tar -zxvf jdk-8u144-linux-x64.tar.gz` (64-bit system) eller
`tar -zxvf jdk-8u144-linux-i586.tar.gz` (32-bit system).
 3. Tilføy `bin` mappen der JDK er installert til miljøvariabelen `PATH`. Dette er avhengig av Linux distribusjonen og shell. For `bash` shell kan det gjøres ved å tilføye:
`PATH=$HOME/Java/jdk1.8.0_144:$PATH` til filen `~/.bash_profile`.
- Installasjon via `.rpm` filen (krever RPM-basert Linux system og superbruker-rettigheter) kan gjøres ved bruk av kommandoen
`rpm -ivh jdk-8u144-linux-x64.rpm` (64-bit system) eller
`rpm -ivh jdk-8u144-linux-i586.rpm` (32-bit system)
- Test installasjonen ved å utføre kommandoen `javac -version` som bør gi følgende (eller tilsvarende) output: `javac "1.8.0_144"`

MacOS

1. Dobbeltklikk på filen `jdk-8u144-macosx-x64.dmg`
2. Flytt filen Eclipse inn på et passende mappe på maskinen.
3. Test installasjonen ved å utføre kommandoen: `javac -version` i en terminal som bør gi følgende (eller tilsvarende output): `javac "1.8.0_144"`

Installasjon av Eclipse IDE (viktig at JDK er installert først)

1. Flytt `.zip` filen (Windows) via høyre-klikk og **Pakk ut alle ...** / `tar.gz` filen (Linux) / `tar` filen (Mac OS) til den mappen hvor du ønsker å installere Eclipse.
2. Pakk ut `.zip` filen (Windows) / `tar.gz` filen (Linux) / `tar` filen (Mac OS).
3. Nå kan man starte Eclipse ved å åpne mappen du valgte for å pakke ut filene og videre til `eclipse` mappen. Der finner du en programfil som heter `eclipse`. Start denne.
4. Når du får spørsmål om workspace, oppretter du et workspace som heter DAT100 (det er praktisk med ett workspace pr. fag og det er mulig å opprette flere workspaces senere).

Windows Installasjonsproblemer

Path miljøvariablen

På noen Windows installasjoner kan det være nødvendig og sette miljøvariablen `Path` så den inneholder stien til `bin` mappen der JDK verktøyene er installert. Dette kan gjøres via **Control Panel** → **System** → **Advanced System Settings** → **Environment Variables...**

Stien til mappen vil avhengige av hvor du valgte å installere JDK. Som standard vil den være: `C:\Program Files\Java\jdk1.8.0_144\bin`.

VIKTIG: stien til JDK skal **tilføyes** til den eksisterende verdi av `Path` dvs. ikke slett det eksisterende innholdet men tilføy følgende på enden (husk `;` først):

```
;C:\Program Files\Java\jdk1.8.0_144\bin
```

HUSK: Stien vil også avhenge av hvilken versjon av JDK du evt. har lastet ned. Hvis du har downloaded en nyere versjon av JDK (eks. versjon 145 isteden for versjon 144) må stien endres tilsvarende.

Test installasjonen via en **Command Prompt** og kommandoen:

```
javac -version
```

Dette bør gi følgende (eller tilsvarende) output: `javac "1.8.0_144"`

Om man endrer i `Path` da må man starte en ny kommando prompt for å få effekt av endringen. Den aktuelle verdi av `Path` kan man se ved å skrive `Path` i kommando prompten.

Konflikt med andre JVM installasjoner

På noen Windows-plattformer kan det forekomme at en versjon av java's virtuelle maskin er installert som forhindrer Eclipse i å starte. Da må man endre i oppstarts-filen `eclipse.ini` som finnes i eclipse mappen ved å tilføye:

```
-vm
```

```
C:/Program Files/Java/jdk1.8.0_144/bin/javaw.exe
```

før linjen som begynner med `-vmargs`. Husk å rette i linjen ovenfor om man har valgt å installere JDK i en annen folder eller har installert en annen versjon av JDK.

En endret versjon av `eclipse.ini` filen finnes også i mappen **Java** i faget DAT100 på it's learning.

Oppgave 1: Første JAVA Eclipse-prosjekt

Formålet med denne øvingen er å sikre at du har en fungerende installasjon av JDK og Eclipse på egen PC.

1. Start Eclipse
2. Velg **File | New | Java Project** → Skriv inn Velkommen som prosjektnavn. Trykk **Finish**
3. Velg **File | New | Class** → Skriv inn `no.hvl.dat100` som pakkenavn og Velkommen som klassenavn. Kryss av for å generere **public static void main ...** automatisk. Trykk **Finish**

Skriv inn:

```
System.out.println("Nå virker det!");
```

slik at hele filen ser slik ut:

```
public class Velkommen {  
    public static void main(String[] args){  
        System.out.println("Nå virker det!");  
    }  
}
```

4. Velg **Run | Run** for å utføre programmet. Dersom du får spørsmål om det skal utføres som **Java Application** eller **Java Applet**, så velger du **Java Application**
5. Du vil nå få utskriften i konsoll-vinduet (nederst i Eclipse vinduet).
6. Modifiser programmet slik det skriver ut en linje med 10 stjerner `*****` før og etter selve teksten.

Oppgave 2: Sjekke ut kode fra github

Vi skal bruke www.github.com til å distribuere kodeeksempler fra forelesninger. Til dette formålet er der opprettet en oppbevaringsplass (*repository*) med navnet H2017 på github som vil inneholde Eclipse-prosjekter for eksemplene.

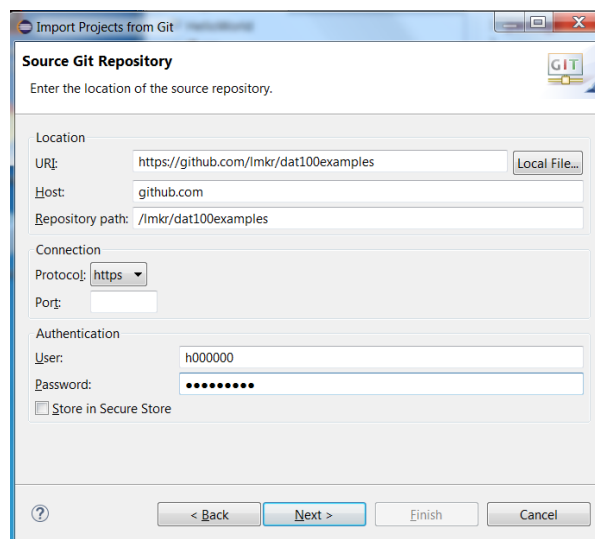
For å få tilgang til oppbevaringsplassene som skal brukes i faget må du opprette en brukerkonto på www.github.com og angi et brukernavn, epost og passord:

- **Bruk studentnummeret dit som brukernavn (*username*).** Eksempelvis h428400 om du har studentnummer 428400.
- I *Step 2 (Choose your plan)* velg *Unlimited public repositories for free*.

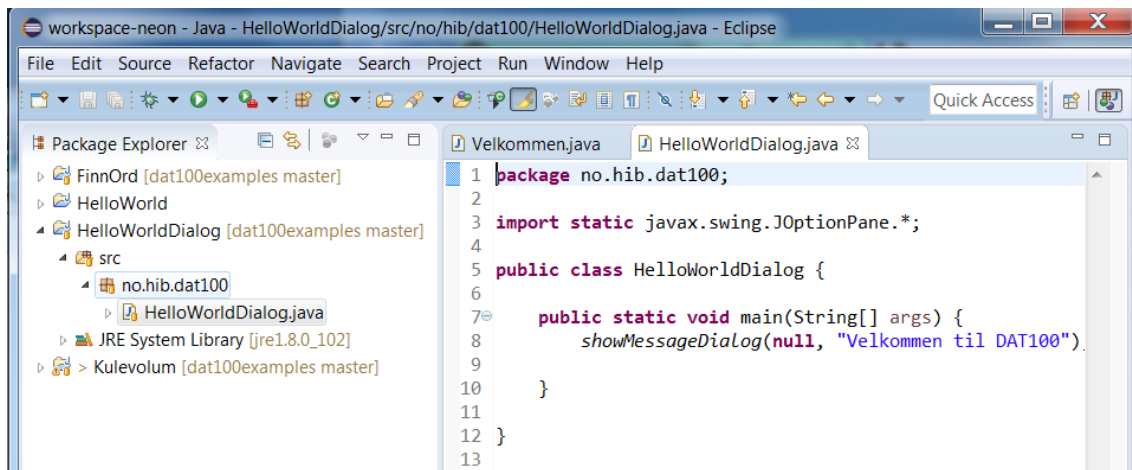
Du vil få en e-post fra github der du blir bedt om å verifisere at e-post adressen er gyldig.

Du kan nå sjekke ut Eclipse-prosjekter med kodeeksempler på egen PC via Eclipse:

1. Velg **File | Import... | Git | Projects from Git | Clone URI**
2. I dialogboksen limer du inn følgende <https://github.com/dat100hib/H2017.git> i feltet URI. Dette angir lokasjon av oppbevaringsplassen for eksempelkode på github.



3. Under *Authentication* legger du inn brukernavn og passord for github-kontoen din og trykker **Next**.
4. I dialogboksen *Branch Selection* velg *master* og trykk **Next**.
5. I dialogboksen *Local Destination* må du under *Directory* velge en mappe på PC'en der du ønsker å lagre den lokale kopien av oppbevaringsplassen.
6. I dialogboksen *Select a wizard to use for importing projects* velg *Import existing Eclipse Projects* og trykk **Next**.
7. I dialogboksen *Import Projects* velg *Select All* og trykk **Finish**. Da vil de Eclipse-prosjektene med eksempelkode vi har lagt inn så langt være tilgjengelig.



Etter hvert i faget legger vi inn flere Eclipse-prosjekter på github. For å få tilgang til de må du oppdatere den lokale oppbevaringsplassen på PC'en din:

1. Velg et av eksempelprosjektene i Eclipse, høyre-klikk og velg **Team | Pull**. Dette oppdaterer oppbevaringsplassen for eksempler på din PC.
2. Velg **File | Import... | Git | Projects from Git | Existing local repository** og trykk **Next**.
3. I dialogboksen *Select a Git Repository* velg *H2017* og trykk **Next**.
4. I dialogboksen *Select a wizard to use for importing projects* velg *Import existing Eclipse Projects* og trykk **Next**.
5. I dialogboksen *Import Projects* kan du nå velge de oppdaterte Eclipse-prosjekter du ønsker å importere inn i Eclipse.

Oppgave 3: Github classroom og JUnit enhetstesting

For noen øvinger skal vi bruke JUnit til å teste java-koden som dere skriver. Videre skal vi for noen obligatoriske innleveringer bruke github classroom til å distribuere den java-koden dere skal ta utgangspunkt i og bruke github classroom til å levere inn koden. Github classroom er en overbygging på github.

Formålet med denne og neste oppgave er å bli kjent med JUnit og github classroom ved å gjennomføre en liten innlevering.

Første steg er å sjekke ut den koden dere skal ta utgangspunkt i. Dette gjøre ved å åpne en nettleser og gå til følgende adresse (URL):

<https://classroom.github.com/a/eT4XhwjD>

Ved å gå til denne adressen kommer du inn i github classroom og kan akseptere oppgaven:

1. Trykk på Authorize application

Authorize application

GitHub Classroom by @github would like permission to access your account



Review permissions



Personal user data

Email addresses (read-only)



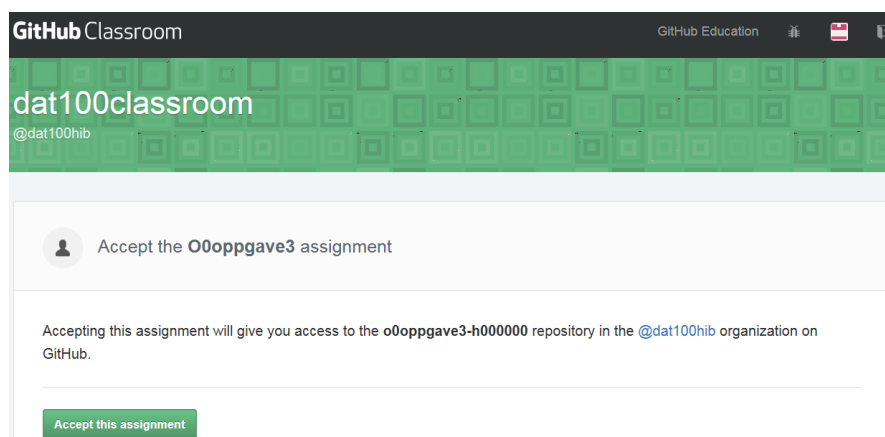
Authorize application

GitHub Classroom

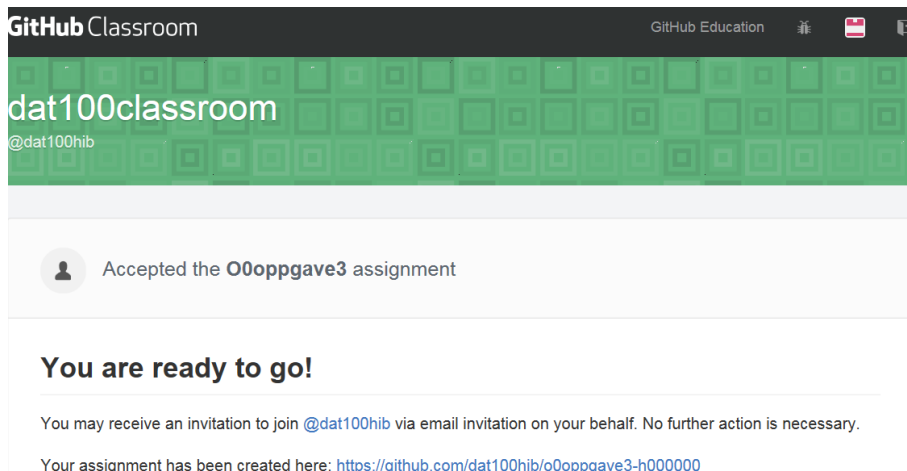
GitHub Classroom automates repository creation and access control, making it easy to distribute starter code and collect assignments on GitHub.

[Visit application's website](#)

2. Velg *Accept this assignment*.

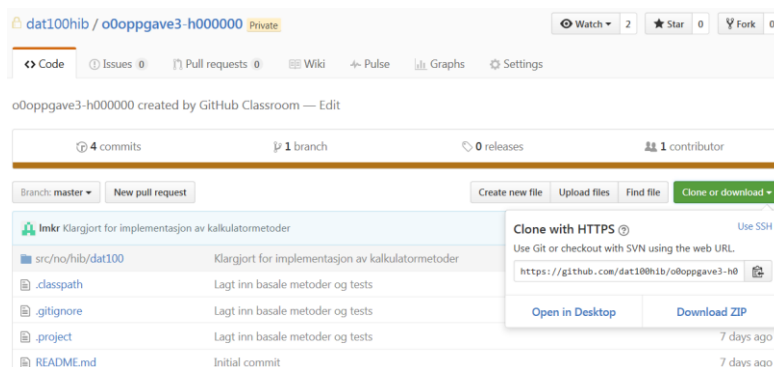


- Trykk på lenken rett etter *your assignment has been created here*. Slutten på den lenken du får vil ikke være helt identisk med det er vist på bildet siden det er en lenke til din egen oppbevaringsplass for den koden du skal ta utgangspunkt i.



Du må nå sjekke ut den koden som dere skal ta utgangspunkt i:

- Velg *Clone or Download* på github-siden i nettleseren som kom frem etter punkt 3 ovenfor.



- Velg lenken og sjekk ut koden som beskrevet i punktene 1-7 i oppgave 1 side 5-6.

Eksempelkoden inneholder et Eclipse-prosjekt *Kalkulator* som implementerer en enkel kalkulator med enkelt brukergrensesnitt.

- Kjør applikasjonen ved å velge prosjektet *Kalkulator* etterfulgt av **Run | Run**
- Kjør JUnit enhetstestene i prosjektet ved å velge filen *TestKalkulator.java* etterfulgt av **Run | Run As | JUnit Test**

Enhetstestene i punkt 2 ovenfor feiler fordi ikke alt er implementert i filen *Kalkulator.java*:

- Åpn filen *Kalkulator.java* i editoren ved å dobbel-klikke på den.
- Implementer metodene *add*, *sub*, *mul* og *div* som vist på første forelesning.

Kjør enhets-testene igjen og prøv applikasjonen via grensesnittet. Sjekk at alt fungerer som ønsket.

Når kalkulatoren fungerer som den skal må du levere inn koden din via github classroom:

- Høyre-klikk på prosjektet *Kalkulator* i Eclipse og velg **Team | Add to Index**. Dette registrerer java-filene som du har endret i prosjektet for å bli lastet opp på den lokale

- oppbevaringsplassen på PC og på oppbevaringsplassen på github classroom.
2. Høyre-klikk på prosjektet *Kalkulator* i Eclipse og velg **Team | Commit ...**
 3. Under *Commit Message* skriv eksempelvis «leverer inn koden» og trykk **Commit and Push**

Du kan sjekke at koden er levert inn ved å gå på www.github.com og velge oppbevaringsplassen for oppgaven du leverte inn.

Oppgave 4: Bruk av EasyGraphics

I forbindelse med forelesninger og programmeringsoppgaver skal vi bruke biblioteket `easygraphics` til å illustrere en del begreper innen programmering. Formålet med denne oppgaven er å vise hvordan man bruker `easygraphics` biblioteket i egne programmer.

For å kunne bruke `easygraphics` biblioteket i egen kode må Eclipse-prosjektet ha en referanse til en jar-fil som inneholder java-koden for biblioteket. Jar-filen `easygraphics.jar` finnes i mappen `easygraphics/lib` der du sjekket ut koden fra github i oppgave 2. Den kan også lastes via its learning.

1. Opprett et nytt Eclipse-prosjekt som i oppgave 1 med navnet *EasyGraphicsTest*. Se evt. oppgave 1 for informasjon om hvordan man oppretter et prosjekt.
2. Legg inn en ny Java-klasse *Grafikk* med følgende innhold (se oppgave 1 for informasjon om å legge til en klasse i prosjektet):

```
package no.hvl.dat100;

import easygraphics.*;

public class Grafikk extends EasyGraphics {

    public static void main(String[] args) {
        launch(args);
    }

    public void run() {
        makeWindow("Grafikk", 350, 150);
        drawCircle(150, 70, 60);
    }
}
```

Koden ovenfor gir i første omgang en rekke feilmeldinger siden vi ikke har fortalt hvor `easygraphics` biblioteket finnes.

3. Høyreklikk på prosjektet *EasyGraphicsTest* og velg **Properties | Java Build Path | Libraries | Add External JARs** og naviger til den mappen der `easygraphics.jar` filen legger. Velg `easygraphics.jar` filen.
4. Jar-filen er nå lagt til prosjektet under Referenced Libraries og feilmeldinger skulle nå forsvinne.
5. Kjør programmet og se at det gir forventet resultat.
6. Modifiser programmet så det tegner to forskjellige sirkler på to ulike steder på skjermen.
7. Skriv et program som tegner et åttetall.

Oppgave 5 (valgfri): Opprette egen github oppbevaringsplass

De som ønsker kan opprette en egen oppbevaringsplass på github til bruk for java-koden som utvikles i faget:

1. Login på www.github.com og velg *New Repository*
2. Under *Create new repository* velg *initialize this repository with a README*
3. Under *Add.gitignore* velg *Java* og trykk **Create**
4. Sjekk ut den nye tomme oppbevaringsplassen på PC'en ved å bruke Eclipse **File | Import... | Git | Projects from Git | Clone URI** som i oppgave 1 men nå med adressen på den nye oppbevaringsplassen som du finner under *Clone or Download* på github.
5. I dialogboksen *Import Projects* velg **Cancel** siden der ennå ikke er noen Eclipse-prosjekter i den nye oppbevaringsplassen.

Du kan legge egne Eclipse-prosjekter til oppbevaringsplassen:

1. Høyre-klikk på prosjektet i Eclipse og velg **File | Share Project...**
2. I dialog-boksen *Configure Git Repository* velg den nye oppbevaringsplassen under punktet *Repository* og trykk **Finish**

Du kan nå legge filer inn i oppbevaringsplassen ved å bruke **Team | Add to Index** og **Team | Commit ...** som beskrevet i punktene 1-3 i slutten av oppgave 3. Der kan være flere Eclipse-prosjekter i den samme oppbevaringsplassen så du trenger kun å gjenta punkt 1-2 ovenfor om du ønsker å legge inn et nytt Eclipse-prosjekt.

Mer informasjon

For de som ønsker mer informasjon om Eclipse og Git:

- Eclipse og Java: Velg **Help | Help Contents** og **Java development user guide**
- Bruk av Git i Eclipse: <http://eclipsesource.com/blogs/tutorials/egit-tutorial/>
- Git Reference: <http://gitref.org/>