

NetBeans

Sisältö

- Johdanto.
- NetBeansin asennus.
- Projektin luominen.
- Tietovaraston perustaminen.
- Projektin asetukset.
- Ohjelman muokkaaminen.
- Kääntäminen ja ajaminen.
- Debuggaus.

Johdanto

- NetBeans on graafinen **ohjelmointiympäristö**.
 - Editori, kääntäjä, tulkki, debuggeri, versionhallinta ja niin edelleen on yhdistetty (integroitu) samaan ohjelmaan.
 - Integrated Development Environment (IDE).
 - Alun perin Java-ohjelmien kehittämisen. Tukee nykyisin Javan lisäksi muun muassa Python- ja JavaScript-kieliä.
 - Kehitys alkoi Kaarlen yliopistossa Prahassa 1996.
 - Apache Software Foundation -järjestön hallinnoima avoimen lähdekoodin projekti vuodesta 2016.
- **Ilmainen** ohjelmisto, joka on saatavilla Windows-, Mac- ja Linux-ympäristöihin.
 - Ilmaisuudestaan huolimatta teollisuustason työkalu.

Johdanto

- **Raskas** ohjelmisto, jonka juoheva käyttö vaatii tehokkaan tietokoneen.
 - Analysoi lähdekoodia sitä kirjoitettaessa.
 - Osa hitaudesta johtuu siitä, että NetBeans on toteutettu Javalla.
- Parhaimmillaan laajoissa projekteissa.
 - Tee pienet ohjelmat jollain kevyemmällä työvälineellä.
- Ominaisuuksiltaan vastaavan tyyppisiä ympäristöjä paljon.
 - Muun muassa ilmainen Eclipse ja kaupallinen MS Visual Studio.
 - Ohjelmointiympäristöt muistuttavat toisiaan: Jos NetBeans on tuttu, on siirtymä johonkin toiseen ympäristöön melko vaivatonta.

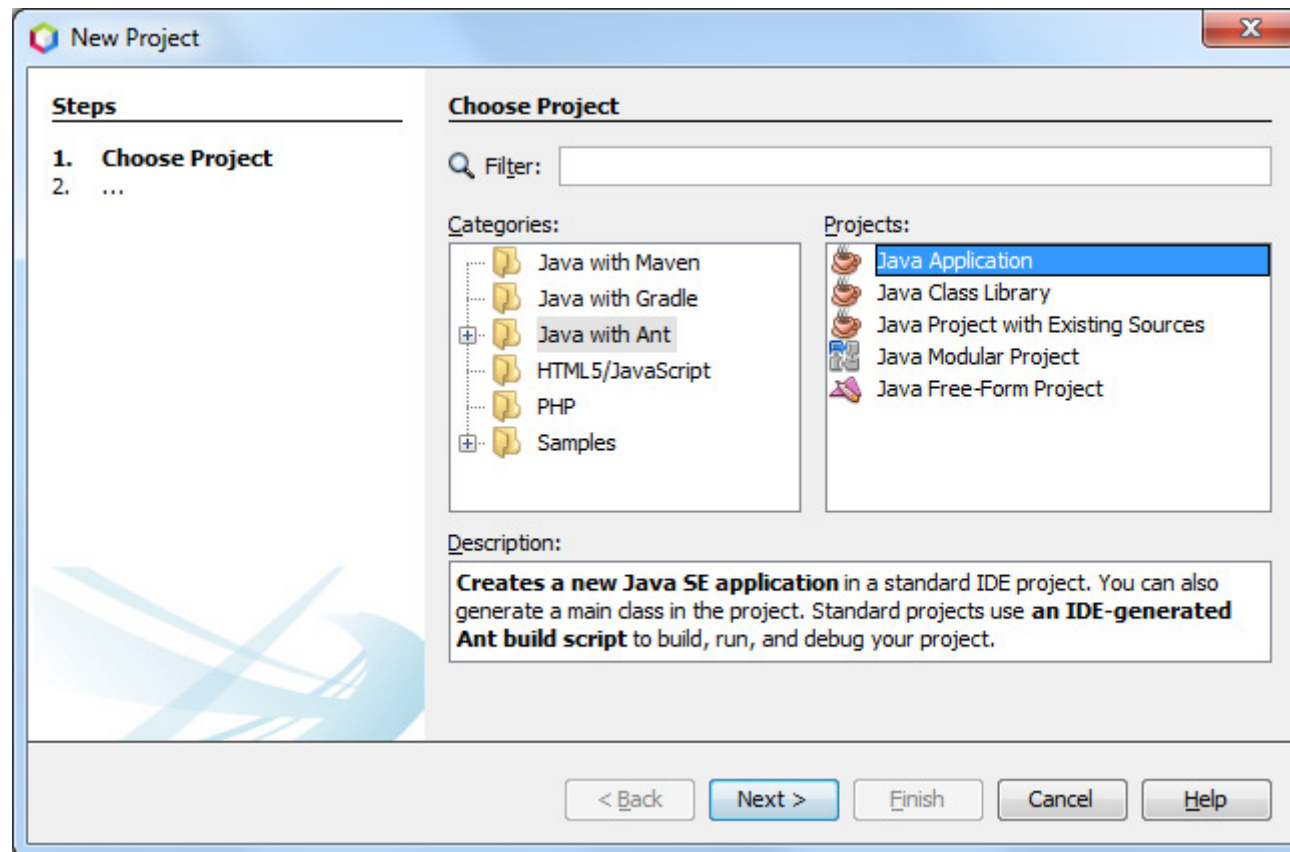
Asennus

- Äärimmäisen helppoa Gitin asennukseen verrattuna.
- Saatavilla verkossa Apachen kotisivujen kautta.
 - <http://netbeans.apache.org/download/>
- Valitse koneesi käyttöjärjestelmälle tehty asennusohjelma.
 - Saatavilla Windowsiin myös zip-paketissa, jolloin ohjelmaa ei tarvitse asentaa, vaan se voi suorittaa puretusta paketista.
- Tarvitsee Java-ohjelmien kehitykseen Java JDK:n, joka on asennettava erikseen Oraclen sivuilta.
 - Tunnistaa JDK:n automaattisesti.
 - JDK:n asennuksesta on kerrottu lisää täällä:
<https://coursepages.uta.fi/tiep5-1/syksy-2019/java-jdk/>

Projektit

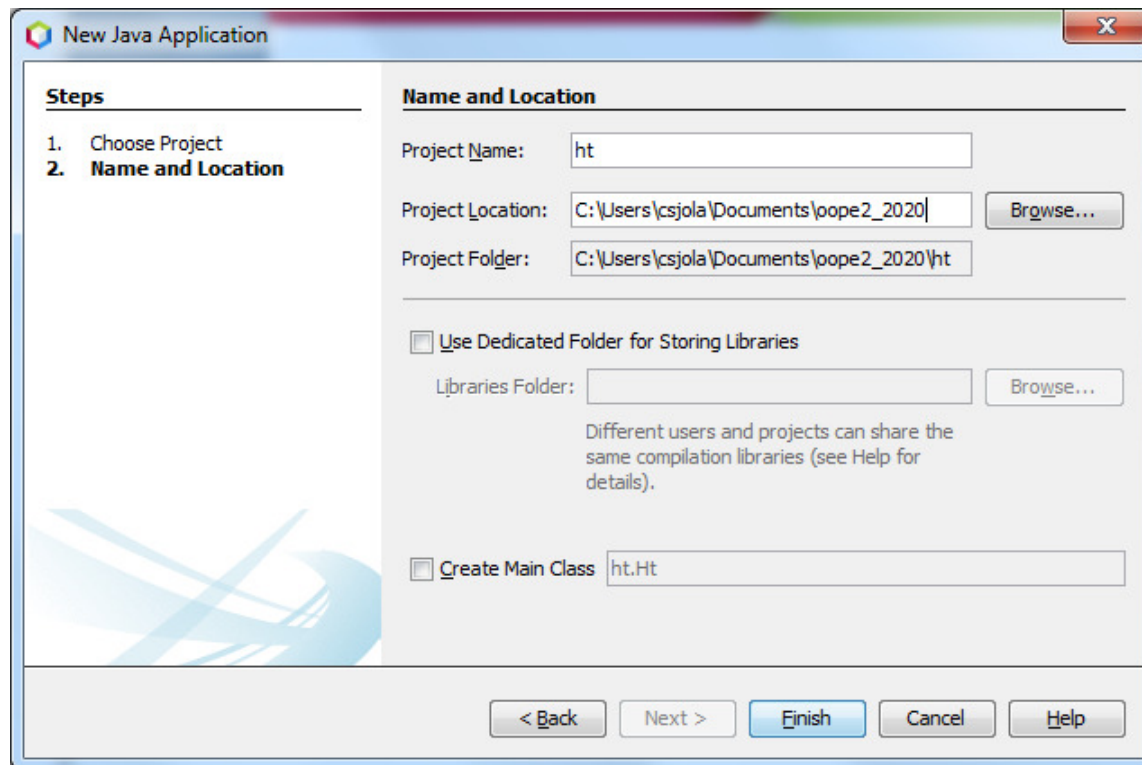
- Jokainen ohjelma on NetBeansissa projekti, joka luodaan dialogien ohjaamana.
 - Projekti sisältää muun muassa lähdekoodin, käännetyn lähdekoodin ja automatisoidun käännöksen määrittelyt.
 - Aloitetaan valitsemalla File | New Project...
- Tällä kurssilla riittää mahdollisimman yksinkertainen projekti: “Java with Ant”.
 - Ant on vanhahtava työkalu projektien automaattiseen kääntämiseen. Tarjolla on ovat myös uudemmat Gradle ja Maven, joita ei kuitenkaan nyt suositella käännösautomaatioon, koska tässä dokumentissa myöhemmin annettavat Git-ohjeet on sovitettu Ant-projekteille.

Projektit



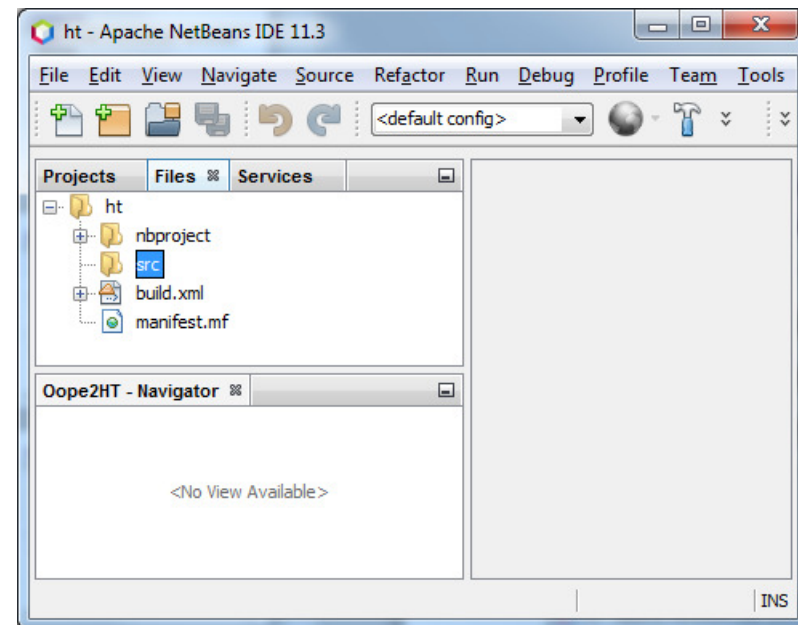
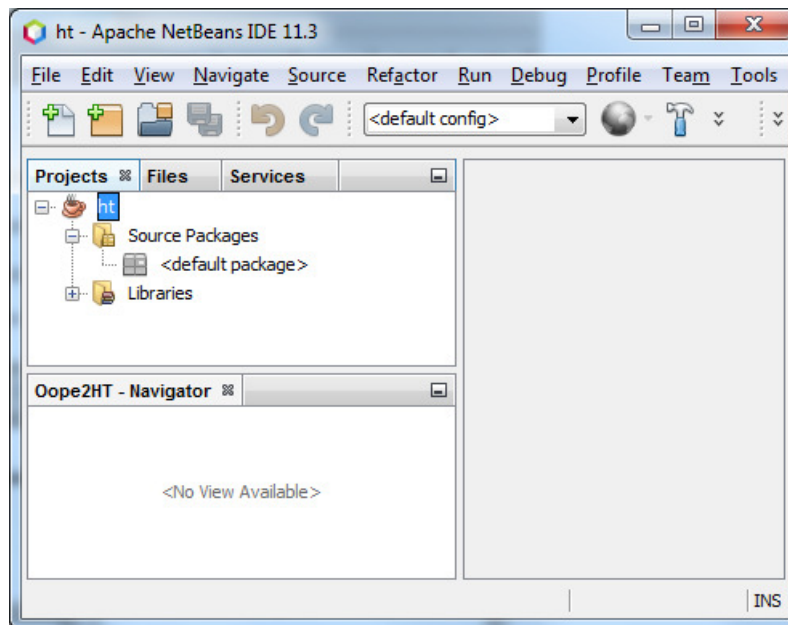
Projektit

- Projektin tiedostot ja hakemistot tallennetaan hakemistoon, jonka nimi johdetaan projektin nimestä (alla *ht*). Ajoluokkaa ei luoda, koska se sijoittuu pakkausten ulkopuolelle. Projekti luodaan klikkaamalla *Finish*.



Projektit

- NetBeans avaa projektin luomisen jälkeen työnäkymän, joko koostuu editori-ikkunasta ja sen vieressä olevista projekti- ja navigointi-ikkunoista. Projekti-ikkunassa näytetään eri välilehdillä muun muassa projektin rakenne ja sen tiedostot. Lähdekooditiedostot ja pakkaushakemistot ovat *src*-hakemistossa.

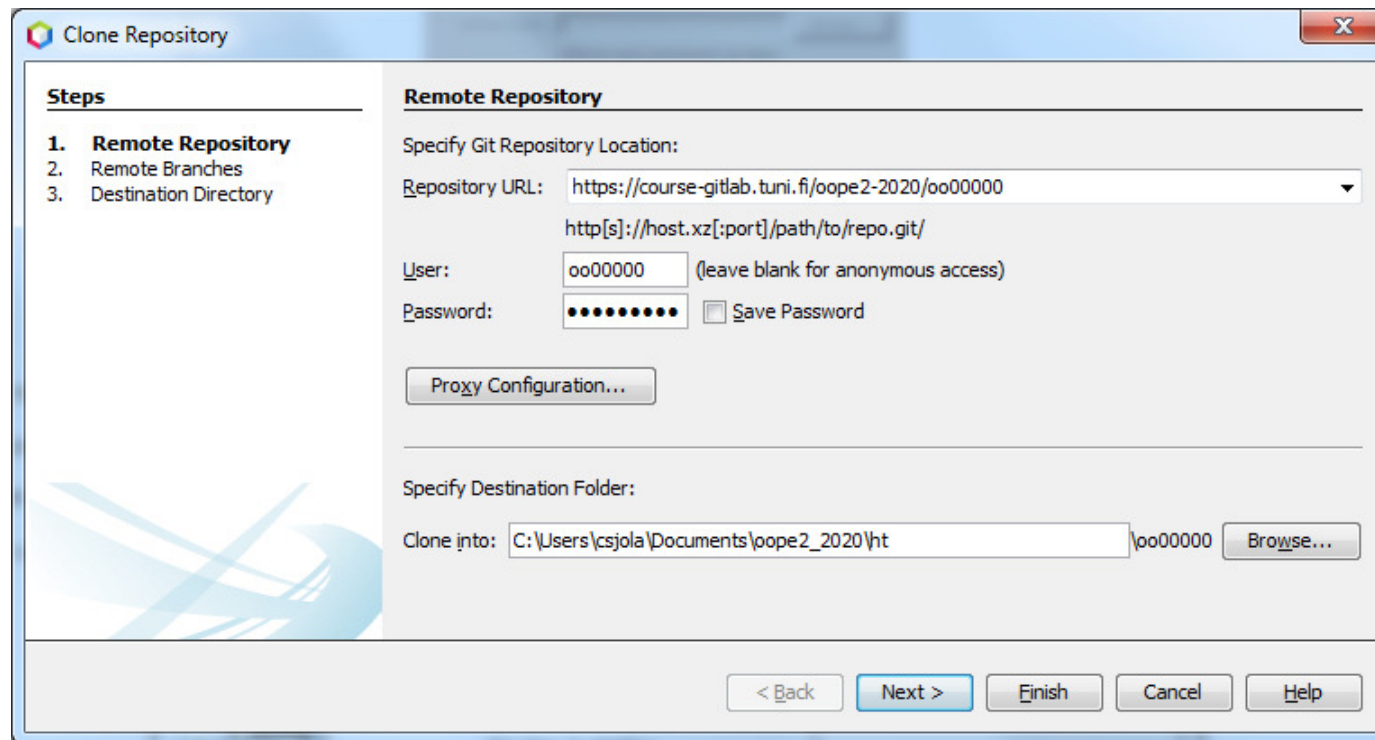


Tietovaraston perustaminen

- Tuni-GitLabissa olevan projektipohjan yhdistäminen NetBeans-projektiin on vaikeaa, koska sekä GitLabilla että projektipohjalla on oma juurihakemisto lähdekoodille.
- Etätietovaraston voi kuitenkin kloonata Ant-projekteissa siten, että Gitin työhakemistoksi on mahdollista valita NetBeans-projektin *src*-hakemisto ilman, että kloonauksen yhteydessä luodaan projektipohjan juurihakemistoa.
- Kloonaus aloitetaan valitsemalla: Team | Git | Clone...
- Git on asennettava ennen kloonausta.

Tietovaraston perustaminen

- Kloonaukseen tehdään samasta osoitteesta kuin komentorivillä. Samoin kirjautumistiedot ovat samat: Tuni-tunnus ja sen salasana. Kohdehakemistoksi valitaan NetBeans-projektin hakemisto.



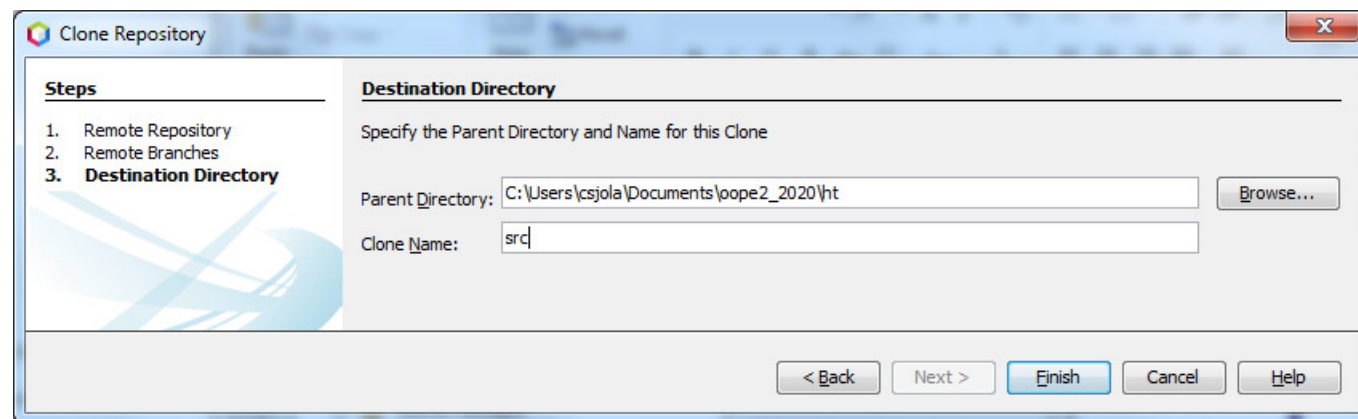
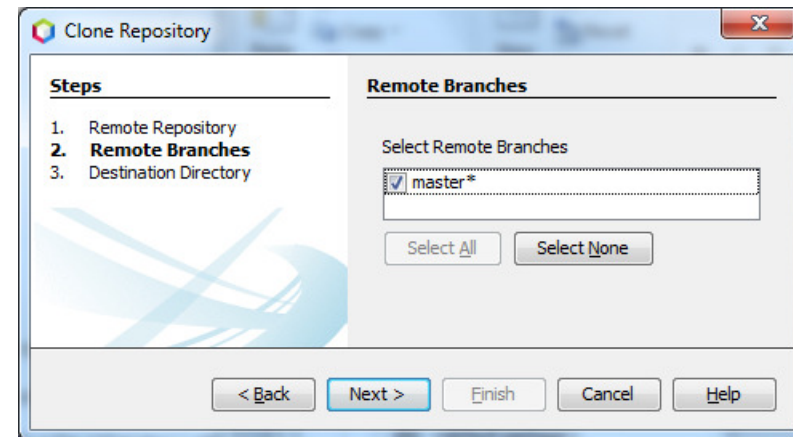
The screenshot shows the 'Clone Repository' dialog box in NetBeans. The 'Steps' panel on the left indicates the current step is '1. Remote Repository'. The 'Remote Repository' section contains the following fields:

- Specify Git Repository Location:**
 - Repository URL:** A dropdown menu showing 'https://course-gitlab.tuni.fi/oope2-2020/oo000000'. Below it is a hint: 'http[s]://host.xz[:port]/path/to/repo.git/'.
 - User:** A text field containing 'oo000000' with a hint '(leave blank for anonymous access)'.
 - Password:** A text field with masked characters '••••••••' and a checkbox for 'Save Password'.
 - A 'Proxy Configuration...' button.
- Specify Destination Folder:**
 - Clone into:** A text field containing 'C:\Users\csjola\Documents\oope2_2020\ht' followed by 'oo000000' and a 'Browse...' button.

At the bottom of the dialog are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

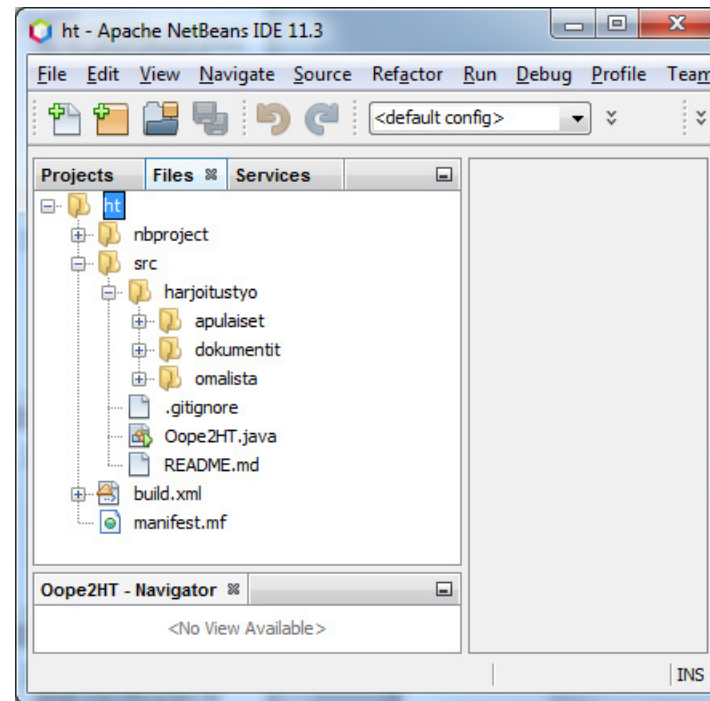
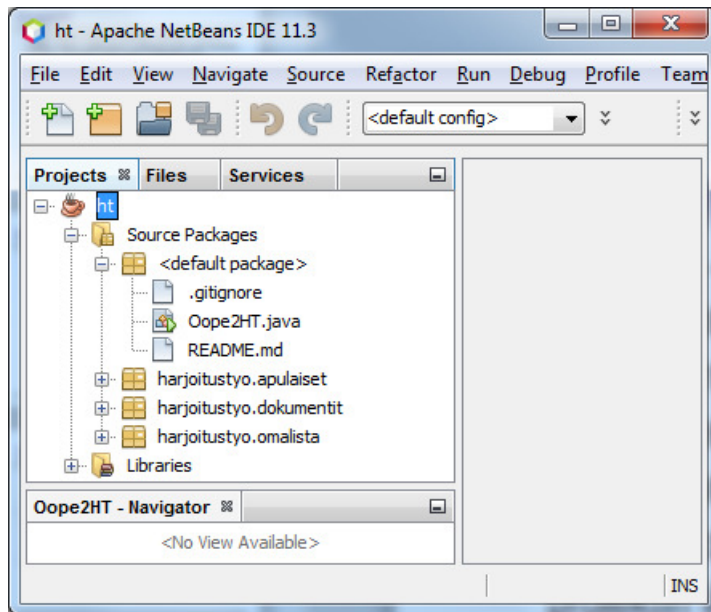
Tietovaraston perustaminen

- Kloonauksen toisessa vaiheessa riittää vain siirtyä kolmanteen vaiheeseen, jossa on erittäin tärkeää antaa kloonin nimeksi *src* eli hakemisto, joka sisältää NetBeans-projektin lähdekoodin.



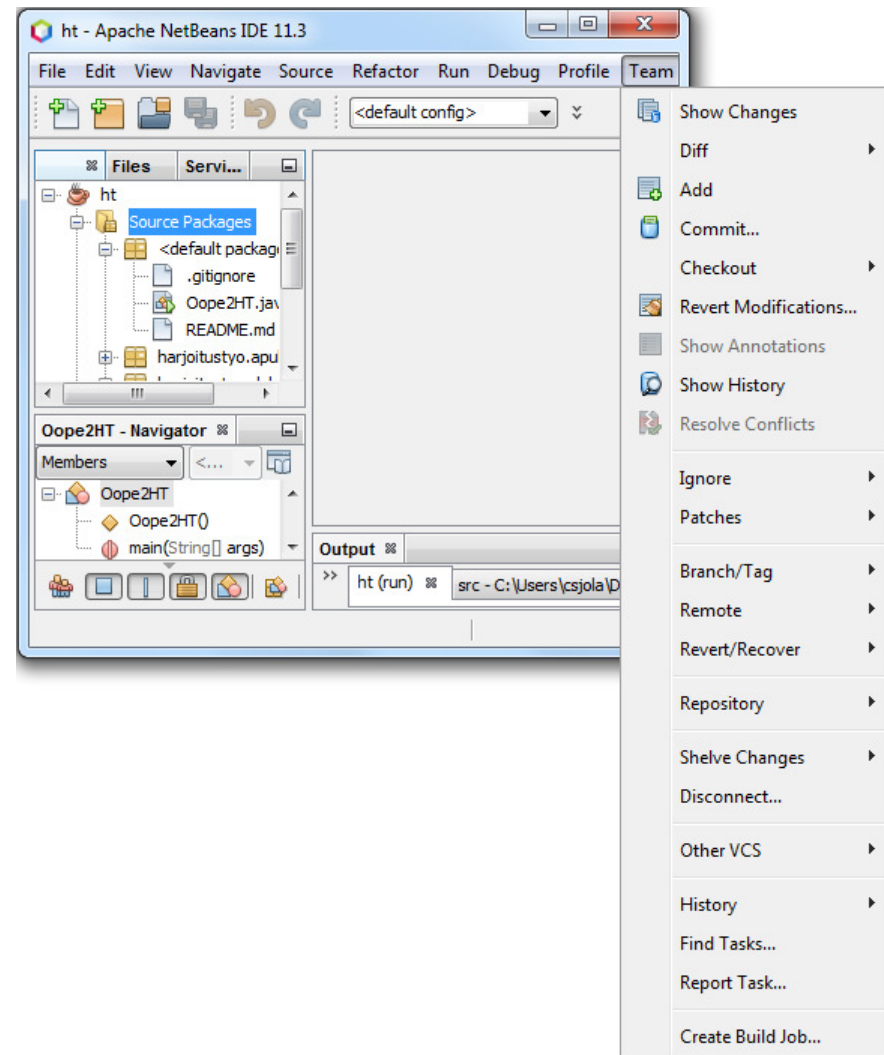
Tietovaraston perustaminen

- Lopputuloksena projektiin on liitetty projektipohjan tiedostot ja pakkaukset. Huomaa, että etävarastosta ladattiin päivitetty projekti, johon on liitetty ajoluokka ja jonka *README.md*-tiedostoa on muokattu (katso edellisen luennon materiaali).



Tietovaraston perustaminen

- Kloonauksen jälkeen Git-komennot löytyvät omina valintoinaan joko *Team*-valikosta, kun projekti-ikkunassa on aktiivisena *Source Packages* tai *src* tai hiiren oikealla näppäimellä avautuvasta hyppyvalikosta projekti-ikkunasta, kun siinä on aktivoituna jompikumpi yllä mainittu kohta.
- Git-komentoja voi kohdistaa myös *Source Packages*- tai *src*-kohtien alta valittuihin yksittäisiin kohteisiin.

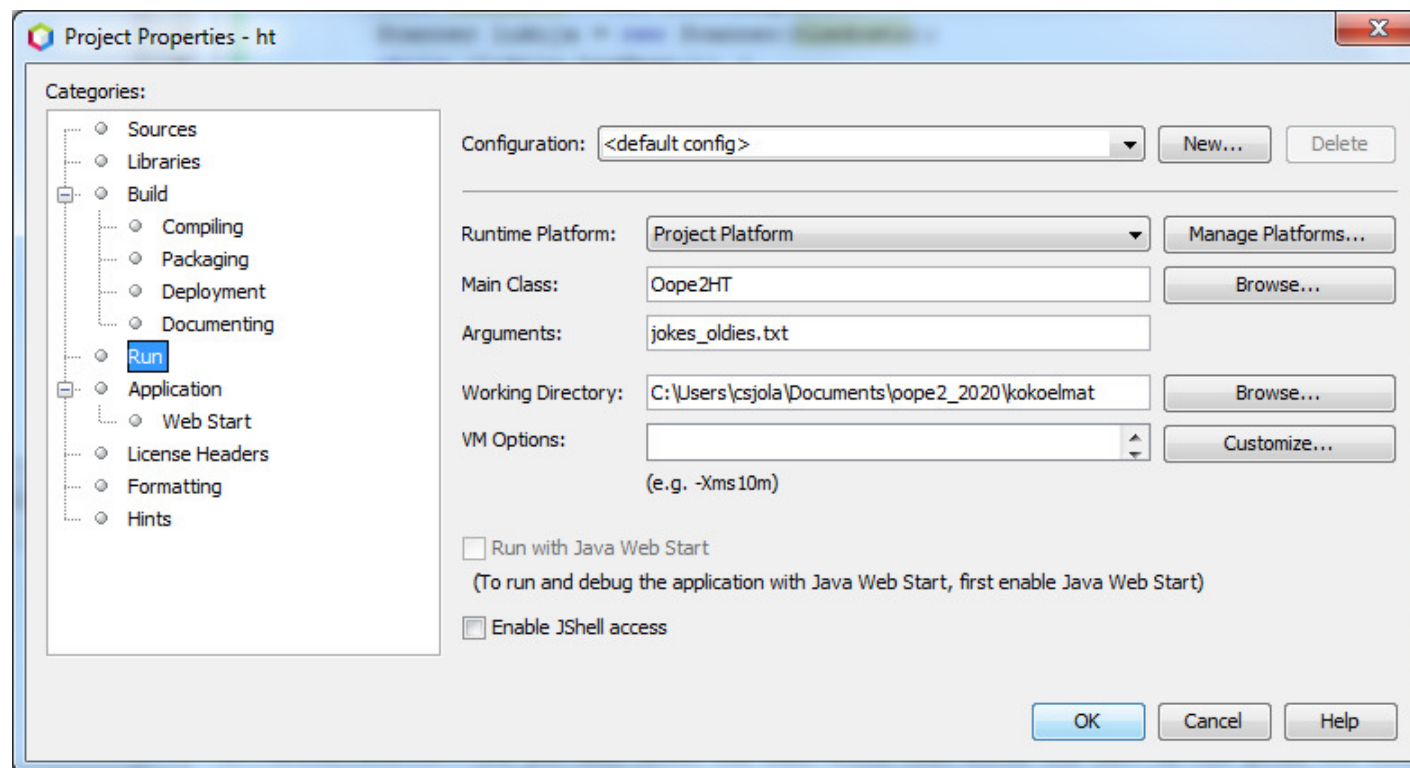


Projektin asetukset

- File | Project Properties tai Properties hiiren oikealla näppäimellä avautuvasta hyppyvalikosta, kun projektipuun juuri on aktiivinen.
- Sources: muun muassa lähdekoodin merkistön asetus.
- Formatting: muun muassa tabulaattorin laatu ja koko. Asetuksia on globaaleja ja kielikohtaisia.
- Build | Run | Main Class: ajoluokan valinta
- Build | Run | Arguments: komentoriviparametrit.
- Build | Run | Working directory: ohjelman suoritushakemisto.
- Aseta *Arguments*- ja *Working directory* -kohdat siten, että ohjelmasi löytää kokoelman ja sulkusanalistan tiedostot, kun se on valmis lukemaan tiedostoja.

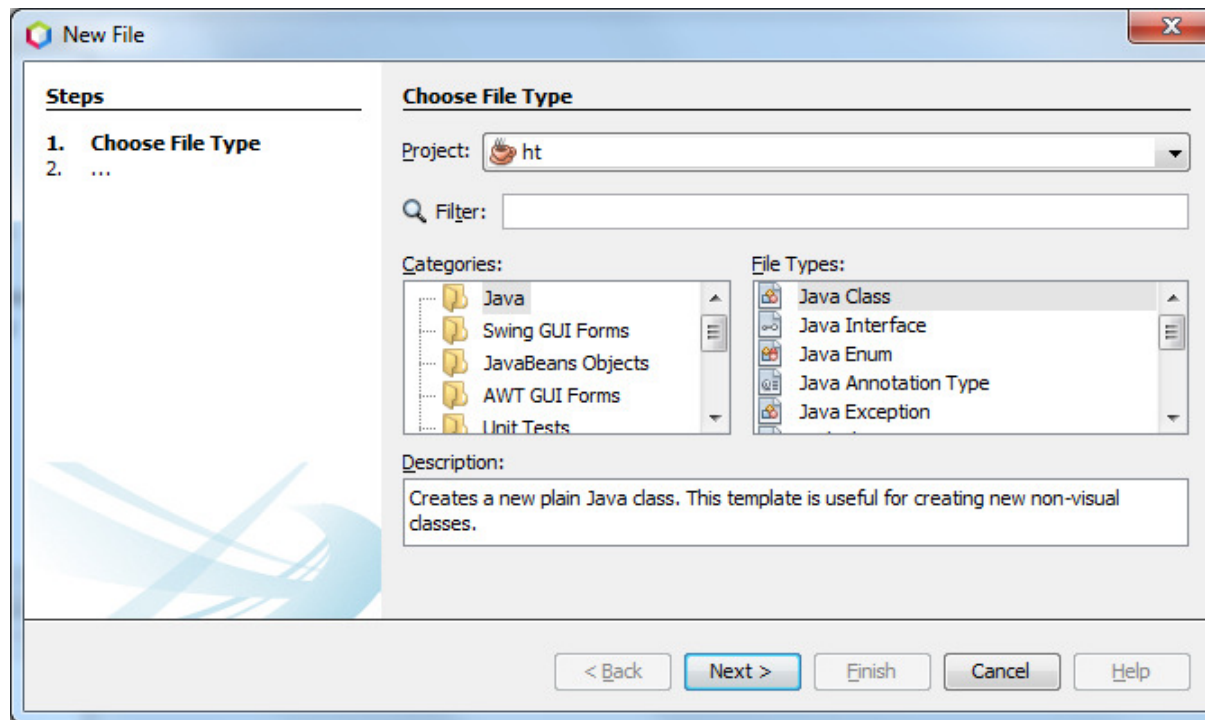
Projektin asetukset

- Oletetaan, että testitiedostot ovat hakemistossa *C:\Users\csjola\Documents\oope2_2020\kokoelmat* ja että tässä vaiheessa halutaan lukea vain tiedostossa *jokes_oldies.txt* oleva vitsien kokoelma.



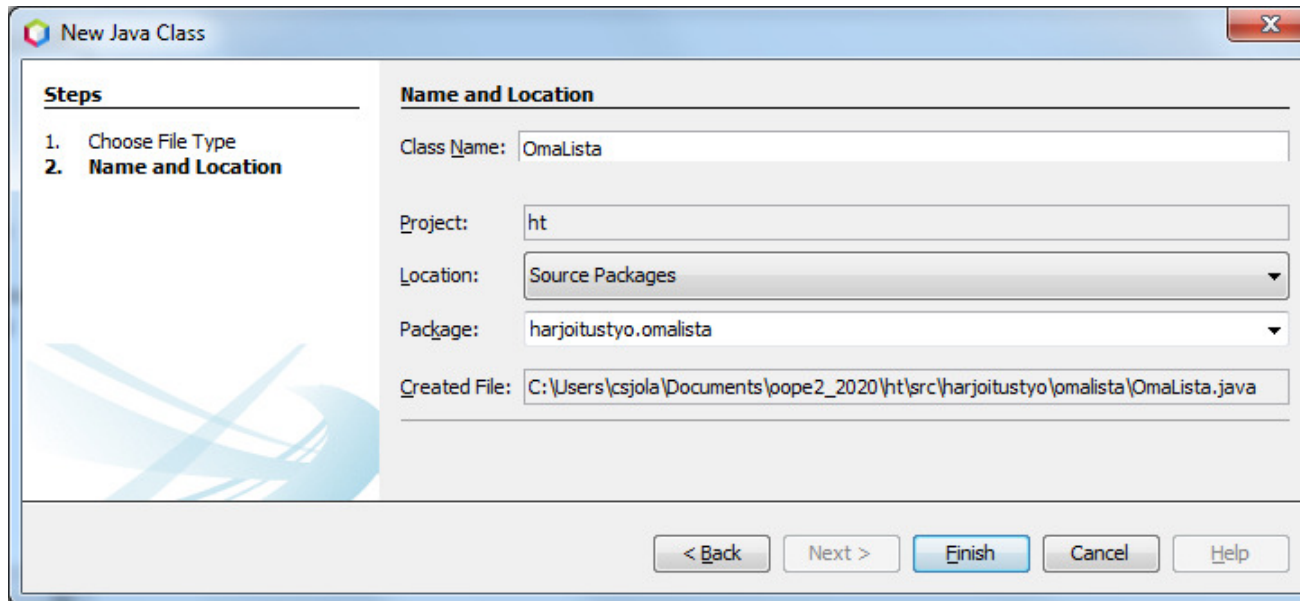
Ohjelman muokkaaminen

- Uusi luokka, rajapinta, lueteltu tyyppi, pakkaus ja niin edelleen luodaan dialogien ohjaamana: File | New File...



Ohjelman muokkaaminen

- Lähdekooditiedoston pakkaus valitaan *Package*-listasta. Muista luoda pakkaus ennen tiedosto, jos olet tekemässä uuteen pakkaukseen tulevaa tiedostoa. **Package**-lause lisätään tiedostoon automaattisesti. Tiedostoa ei ole pakko liittää mihinkään pakkaukseen. Ajoluokan tulee olla *src*-hakemistossa pakkausten ulkopuolella eli ei missään *src*:n alihakemistossa.



New Java Class

Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name: OmaLista

Project: ht

Location: Source Packages

Package: harjoitustyo.omalista

Created File: C:\Users\csjola\Documents\oope2_2020\ht\src\harjoitustyo\omalista\OmaLista.java

< Back Next > Finish Cancel Help

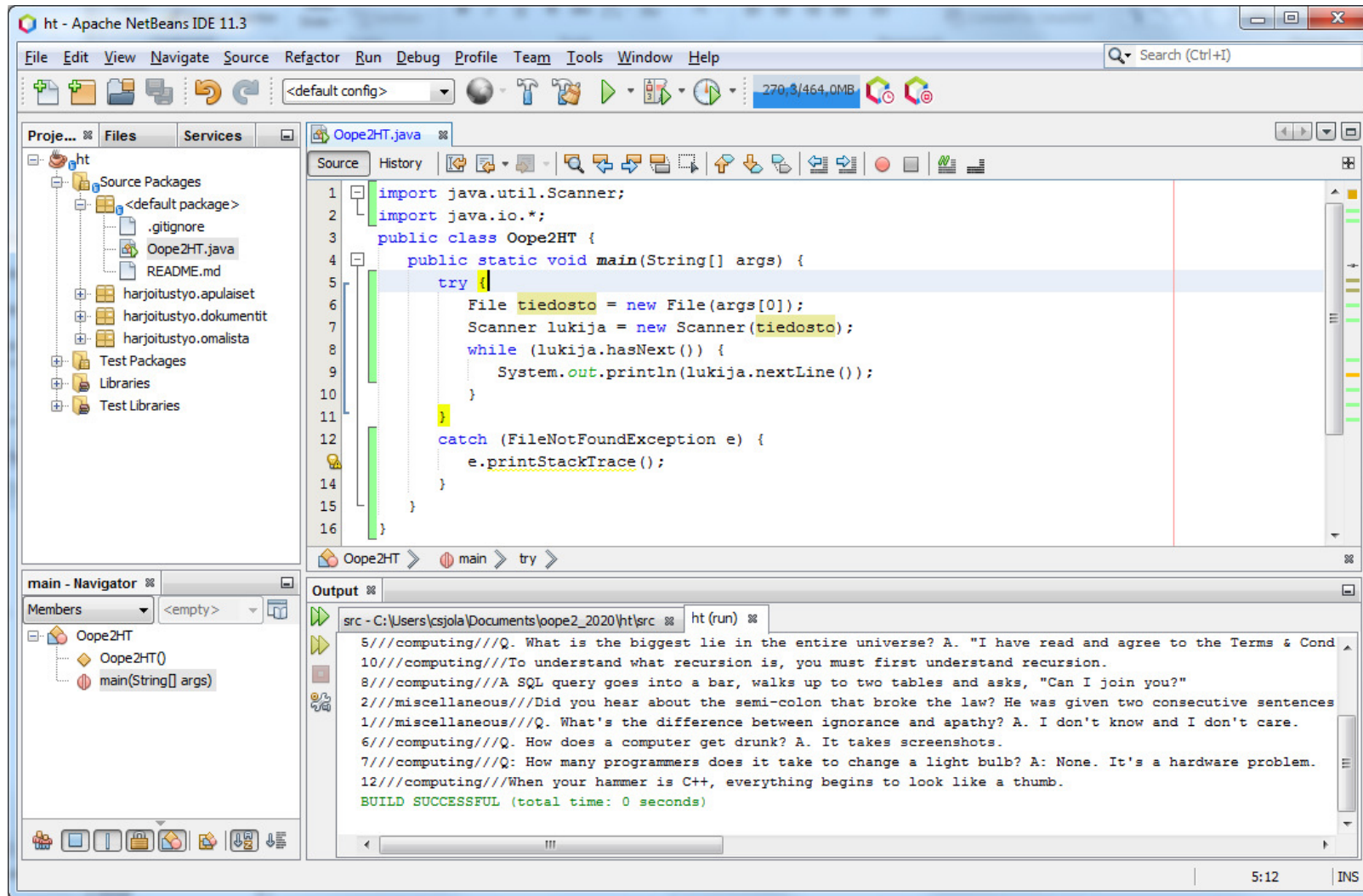
Ohjelman muokkaaminen

- Tiedosto valitaan muokattavaksi esimerkiksi tupla-klikkauksella projekti-ikkunassa.
- Navigointi-ikkuna näyttää valitun tiedoston rakenteen. Esimerkiksi luokasta listataan attribuutti ja metodit
- NetBeans analysoi lähdekoodia sen kirjoittamisen aikana.
 - Editori näyttää kielioppivirheet punaisina pallukoina ja varoitukset ja parannusvihjeensä keltaisina pallukoina.
 - Vihjeet voi kääntää pois päältä: File | Project Hints
- **Älä luota liikaa NetBeansiin** tai vastaavaan työkaluun.
 - Mikään työväline ei korvaa omaa osaamista.
 - Vinkit saattavat johtaa harhaan varsinkin, jos niitä ei ymmärrä.

Kääntäminen ja ajaminen

- Run | Build project: päättelee mitä kaikkea tulee kääntää ja kääntää projektin.
- Run | Clean and Build Project: tuhoaa ensin kaiken tavukoodin ja kääntää sitten projektin.
- Run | Run Project tai iso vihreä kolmio: kääntää ja ajaa projektin.
 - Ajettaessa editori-ikkunan alle avautuu tulosikkuna, jossa esitetään ohjelman tulosteet ja luetaan syötteet, jos ohjelmalla tekstipohjainen käyttöliittymä.

Kääntäminen ja ajaminen



Debuggaus

- Projektinhallinnan lisäksi NetBeansin arvokkaimpia ominaisuuksia on visuaalinen askeltava bebuggaus.
- Ennen debuggausta ohjelman koodiin lisätään yksi tai useampi pysäytyspisteitä (break point) klikkaamalla hiirellä niiden rivien numeroita, joille ohjelman halutaan pysähtyvän.
- Sitten käynnistetään ohjelma debuggaustilassa: Debug | Debug Project tai ajonapin oikealla puolella oleva nappi.
- Tulostusikkunaan avautuu debuggaukseen liittyviä välilehtiä. *Variables*-välilehdellä voi seurata ohjelman olioiden tilaa.

Debuggaus

- Ohjelma pysähtyy ensimmäiseen pysäytyspisteeseen, josta voidaan jatkaa eri tavoin:
 - Step over: lausekkeessa olevat metodikutsut ohitetaan ja siirrytään seuraavaan lauseeseen.
 - Step over expression: ohitetaan tietty lauseen lauseke.
 - Step into: seurataan lauseen metodikutsua.
 - Step out: palataan metodista takaisin.
- Debuggaus keskeytetään valikon isolla punaisella napilla.