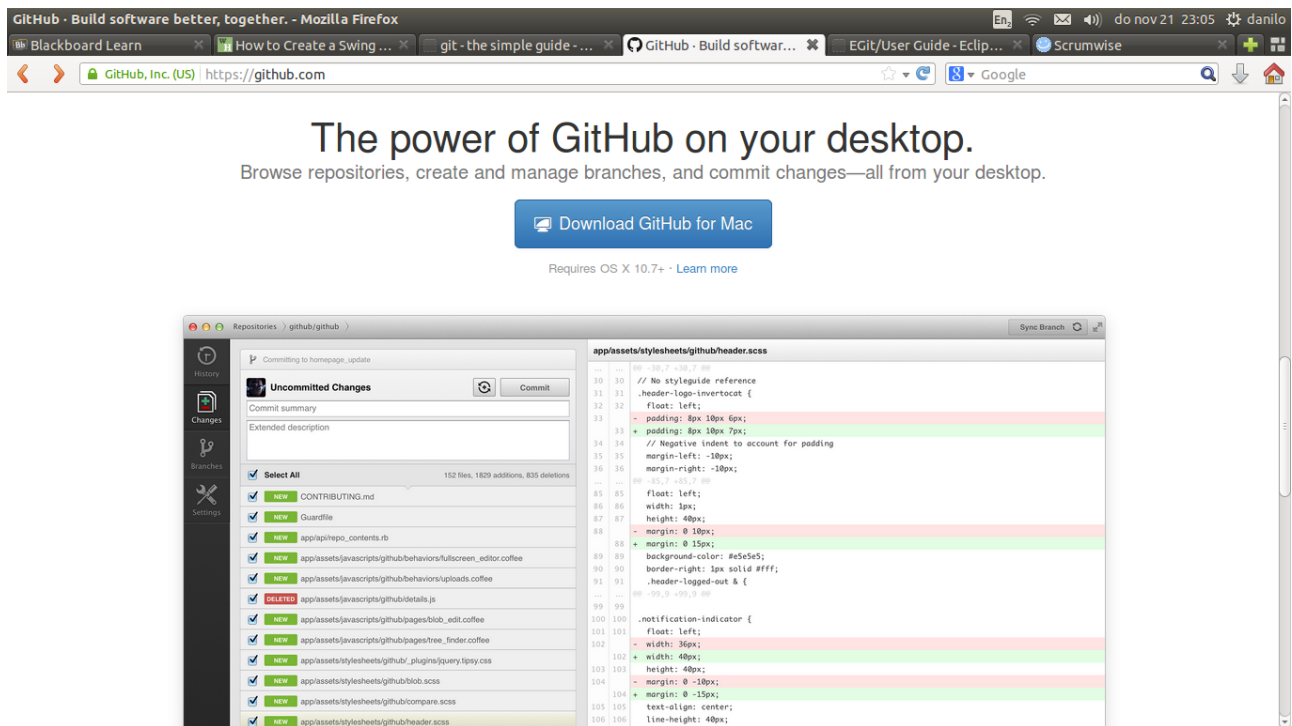


INSTALLATION

1. Als je dit nog niet hebt gedaan:

- ga naar www.github.com en installeer git voor jouw OS. (als je uitlogt en naar beneden scrollt op de

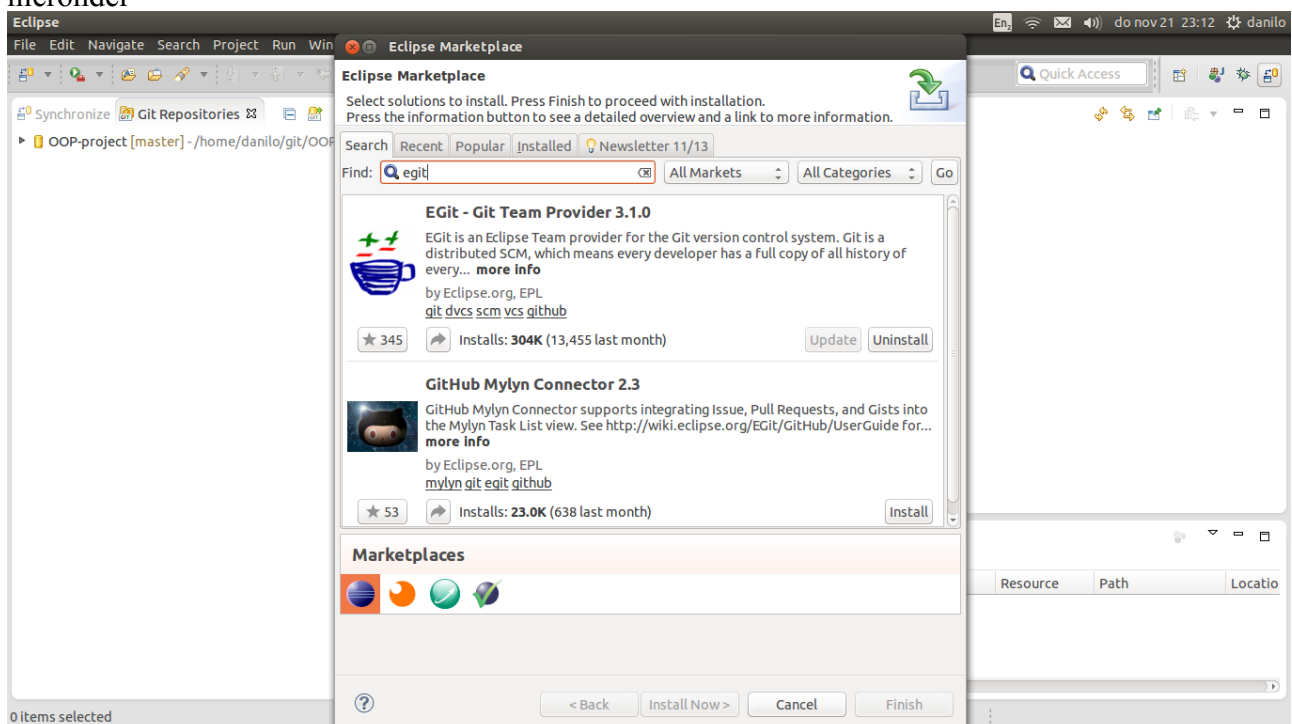


homepage zie je een link) zie ook de foto hieronder.

2. installeer EGit voor eclipse. Hoe doe je dit:

- open eclipse, ga naar menu option -> 'Help' en dan -> 'Eclipse marketplace' (ik krijg hier geen screenshot van sorry)

- zoek naar 'egit', installeer degene die als kopje heeft 'EGit – Git Team provider x.x.x' -> zie foto hieronder



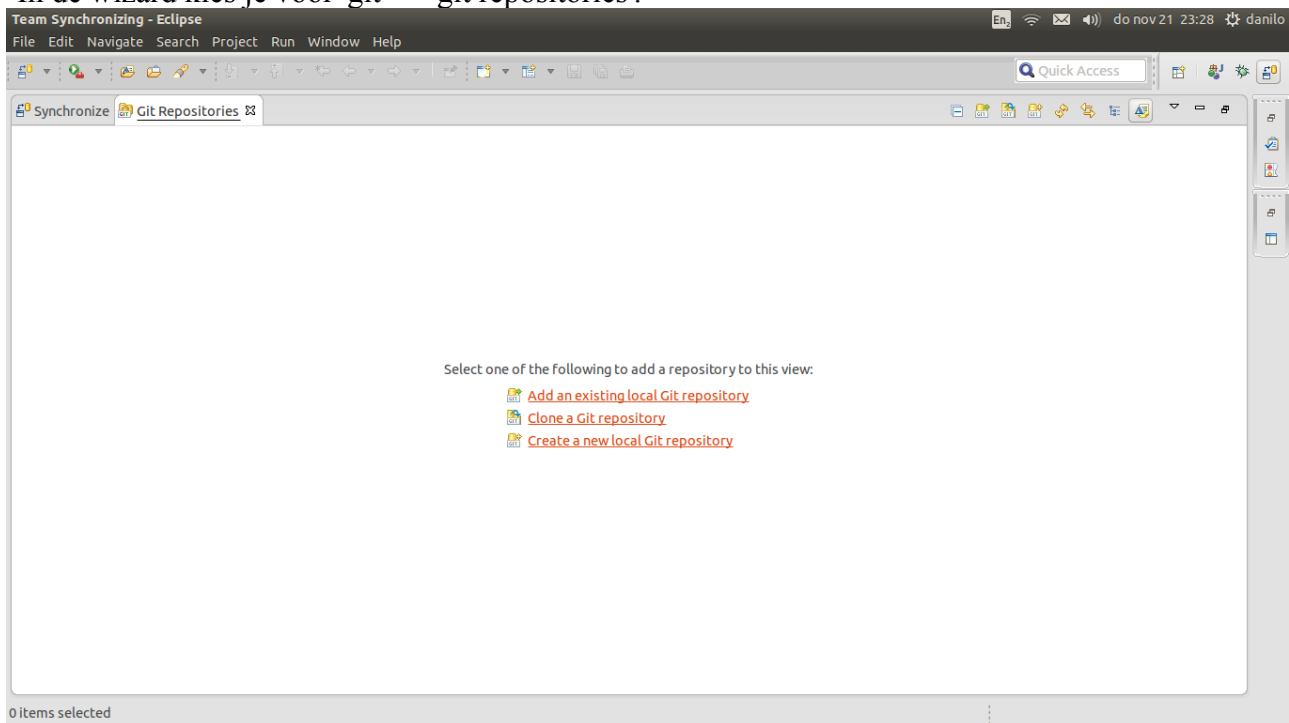
|
V

SET-UP

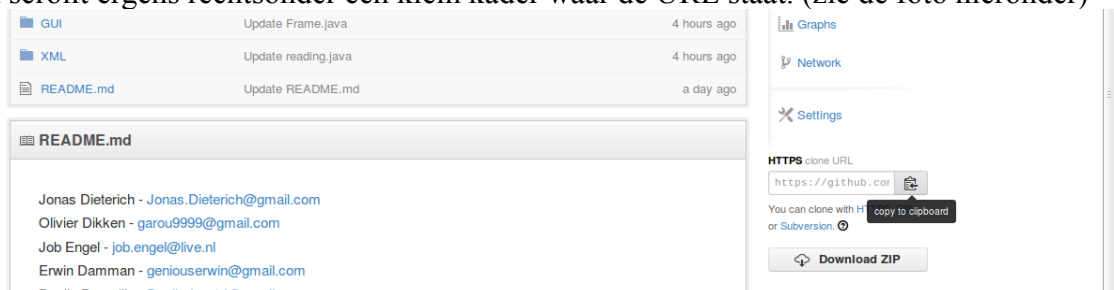
NOTE: in je eclipse package explorer heb je als je met je RMB op een project drukt een nieuwe optie. ('team'). Deze ga je heel vaak gebruiken. Je hebt als het goed is nog geen local repositories gelinkt aan eclipse. Bij 'team' staat nu dus ook alleen 'apply patch' en 'share project' (als 'share project' niet werkt moet je eerst de project map openen, je zult zien dat het dan wel werkt)

de stappen:

1. Ga in de eclipse menu naar -> 'Window' -> 'show view' -> 'other'.
- In de wizard kies je voor 'git' -> 'git repositories'.

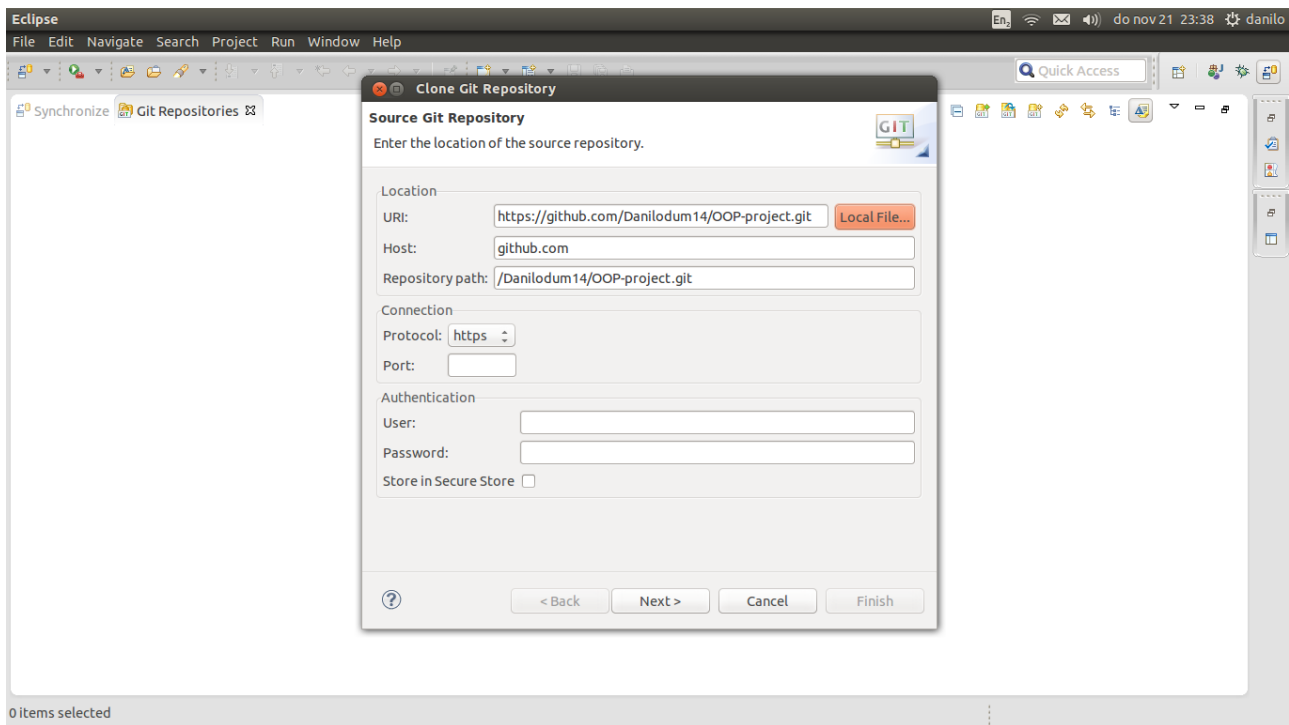


2. Je hebt nu een nieuwe view in je workspace. Dit zou er als volgt uit moeten zien (zie foto)
3. klik op 'clone a Git repository'.
4. Ga nu naar onze remote repository webpage op www.github.com // op onze main page staat als je wat naar beneden scrollt ergens rechtsonder een klein kader waar de URL staat. (zie de foto hieronder)



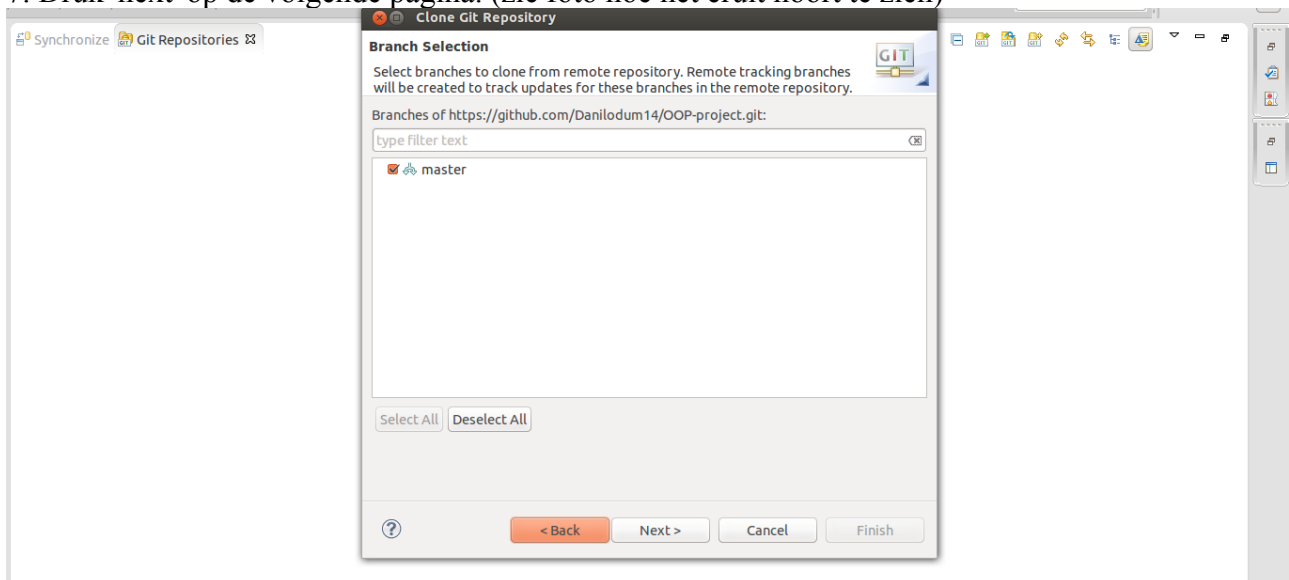
-kopieer de 'HTTPS clone URL'

5. plak de URL in de eclipse 'clone git repository' wizard, zie de foto hieronder, na het plakken wordt dit automatisch zo ingevuld.

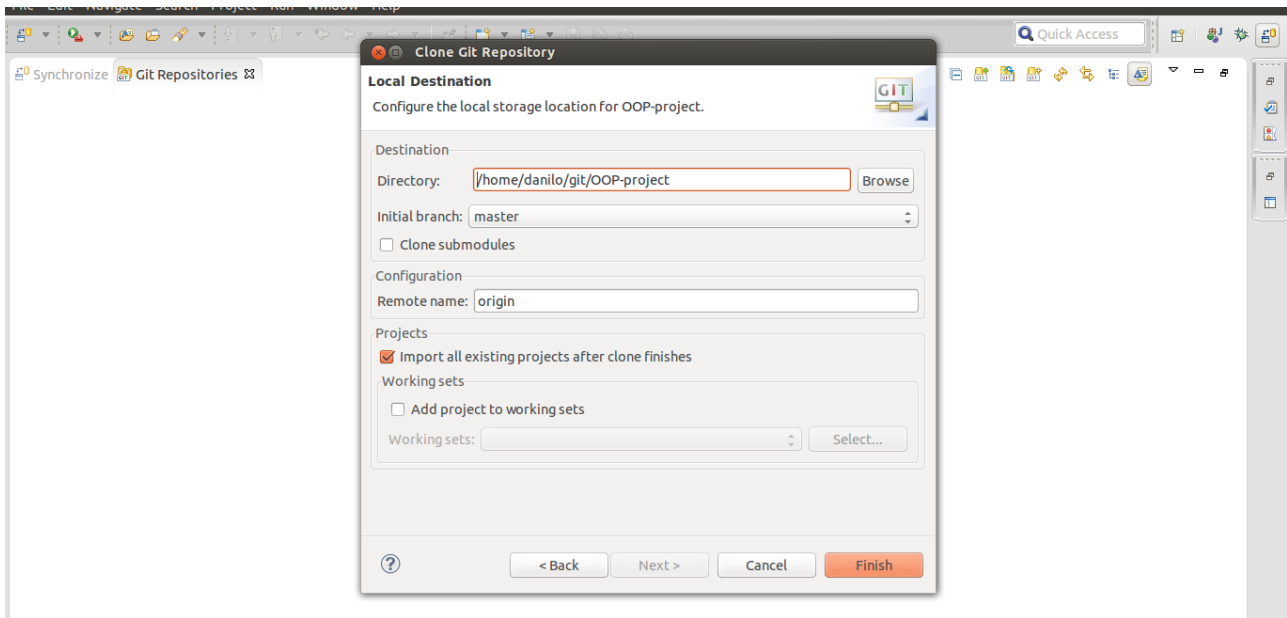


6. vul bij 'user' en 'password' je github gegevens in. Selecteer 'store in secure store' om dit maar een keer te hoeven invullen.

7. Druk 'next' op de volgende pagina. (zie foto hoe het eruit hoort te zien)



8. Op de volgende pagina moet je specificeren waar je local git repository komt te staan in je file system. Doe iets leuks. **BELANGRIJK: vink hier gelijk de checkbox 'import all existing projects after clone finishes' aan.** (Zie de foto op de volgende pagina voor extra duidelijkheid)



9. voila, in je package explorer zie je de projecten die ook te vinden zijn op onze remote repo.

10. Lees hetgene wat nu komt goed door. Ik probeer dingen waar ik tegen struikelde extra duidelijk te maken. Als iets niet werkt vraag het dan op whatsapp, want hiermee zelf lopen kutten is tijdverspilling.

PRE-WORK

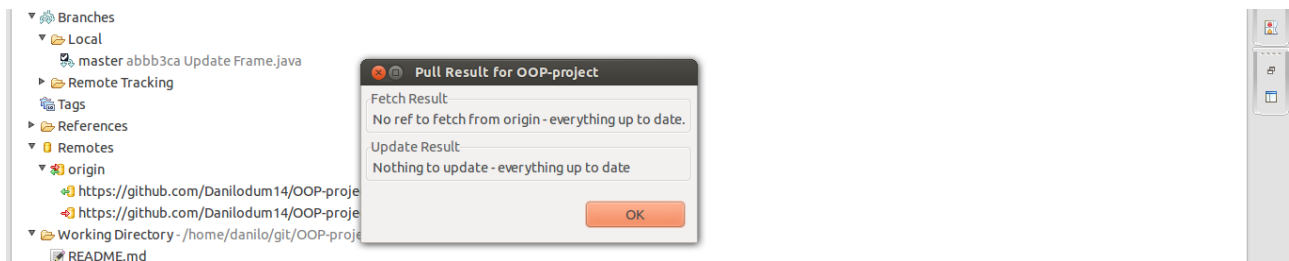
1. elke keer als je aan de slag gaat en je wilt werken met de code die we online hebben staan, moet je het volgende doen:

*NOTE: je moet dit **altijd** doen. Het voorkomt de volgende error: [rejected non-fast-forward]*

- open de 'git repositories view' en dubbelklik op de (als het goed is) de enige repository die je ziet. Je krijgt (ongeveer) het volgende te zien (zie foto)



1. RMB op 'OOP-project [master]' en kies 'pull'.



- ik kreeg het venster hierboven te zien.

NOTE: wat hier gebeurd is het (deels) synchroniseren van de remote repo met de local repo. Er worden twee processen uitgevoerd:

- fetch en merge

Je kan dit ook zelf doen:

- open het kopje 'OOP-project [master]' -> 'Remotes' -> 'origin'.
- RMB op de link met de groene pijl en kies 'fetch'.
- open het kopje 'OOP-project [master]' -> 'Local'.
- RMB op 'master xxxxxx <<latest commit>>' (dit is t enige wat onder het kopje Local staat).
- kies voor 'merge'.

2. open het kopje 'OOP-project [master]' -> 'Remotes' -> 'origin'.

- RMB op de link met de rode pijl en kies 'push'.

Dus in het kort :

1. fetch
2. merge
3. push

Je kan nu aan de slag (lees verder hoe je je veranderingen op moet slaan)

*NOTE: Ik ben deze situatie nog niet tegengekomen omdat nog niemand heeft 'gecommit', maar het zou kunnen dat iemand een nieuwe java project toevoegt aan de remote repo, ik weet niet of deze dan automatisch te vinden is in je working directory (en dus te zien in je package explorer en je workspace). OPLOSSING: je zult via de eclipse menu moeten importeren -> file -> import -> git -> Projects from git (next) -> Existing local repository (next) -> 'OOP-project' (next) -> (vink aan) 'import existing projects' (next) -> * hier de ontbrekende projects aanvinken * (etc) ->*

BELANGRIJK: *Je working directory, local repository en remote repository zijn drie verschillende dingen. Vooral de synchronisatie van de eerste twee is soms niet even duidelijk. Daarom is het extra belangrijk dat we in 'Java project' vorm te werk gaan. Deze zijn namelijk makkelijk te importeren (zie hierboven). Losse .java files moet je handmatig kopiëren/importeren naar je working directory vanuit je local repository, dit maakt het hele process een stuk trager. Je wil snel en makkelijk kunnen werken.*

WORK

Voeg een nieuwe klasse toe en stuur het naar de remote repository:

1. voeg de klasse toe zoals gewoonlijk.
2. je ziet nu in de package explorer een vraagteken in plaat van gele cylinder.



3. RMB op de project naam en kies 'team' -> 'add to index'.

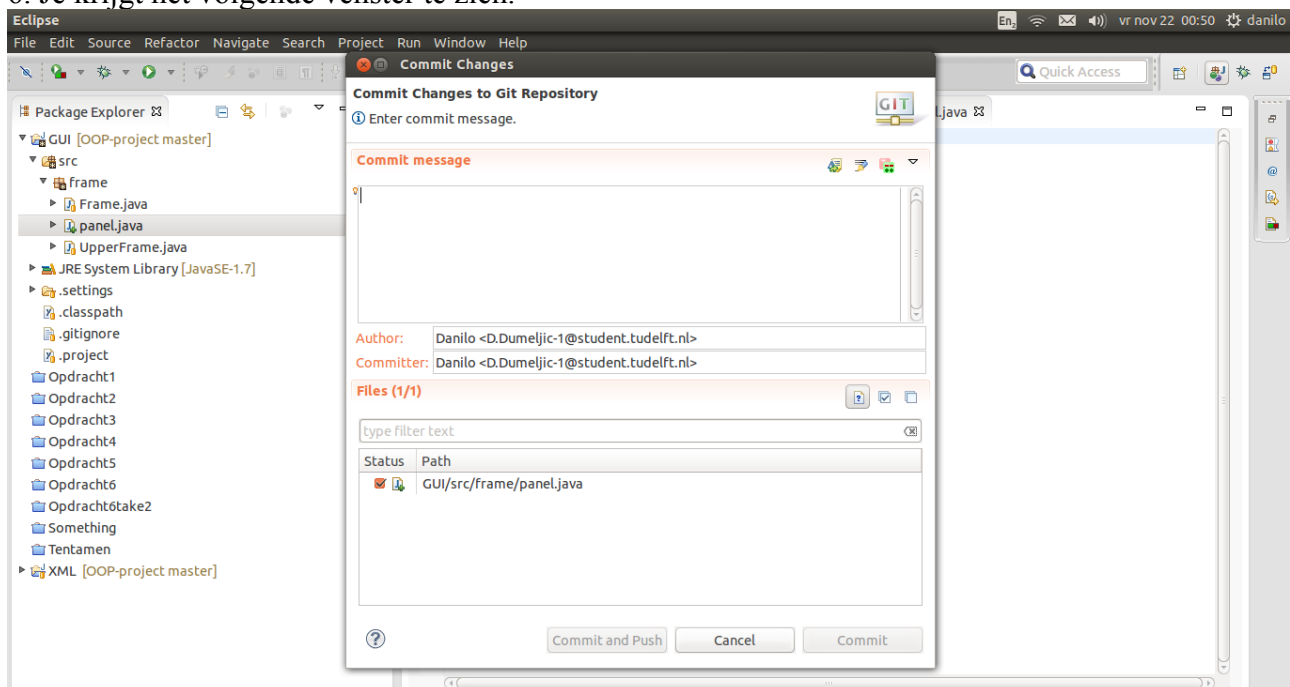
4. er komt nu een plusje voor te staan.

//4.1. implementeer eventueel al iets in deze nieuwe klasse.

NOTE: wat extra informatie is wel van toepassing. Je maakt hier lokaal een klasse aan, maar de local repository weet hier nog niks van. Daarom voeg je deze klasse eerst toe aan de index. Vervolgens wil je de veranderingen definitief maken. In andere woorden: je maakt een 'commit'. Een commit maken betekent dat hetgene wat je commit altijd teruggehaald kan worden. Je doet het niet zomaar, nee, alle commits kunnen bekeken worden, ook al werk je op dezelfde files, je kan altijd terug naar een commit (lees: versie). So... how do I commit? --->

5. RMB op de klasse die klaar is om te committen. (hier is een shortcut voor)

6. Je krijgt het volgende venster te zien.



7. Voeg hier **altijd** een **nuttig** bericht aan toe. Dit krijgt de rest van de groep ook te zien en het is fijn om in een oogopslag te kunnen zien wat er nou precies is toegevoegd. (je moet overigens een bericht toevoegen, so please, useful (concise) messages s'il vous plait)

8. kies voor 'commit' of 'commit and push'.

NOTE: handig om te weten; EGit biedt een paar views, een daarvan is 'Git Staging'. Een handige view voor commits. (alle aangepaste documenten die aan de index zijn toegevoegd zullen in de 'unstaged' gedeelte terechtkomen, je kan ze naar 'staged' slepen om ze te committen.

NOTE: Het zou kunnen dat ook al heb je voordat je begonnen bent met werken netjes gepulled en gepusht, je alsnog een [rejected non-fast-forward error] krijgt. Dit kan gebeuren als iemand anders, terwijl jij code aan het schrijven was, iets heeft gepushed naar de remote repo.

OPLOSSING: je moet alsnog eerst fetchen van de remote en dan de code die iemand anders heeft geschreven mergen met wat jij zelf hebt. Het kan zijn dat hier overlap tussen ontstaat, maar het kan ook zijn dat jullie aan verschillende files zaten en alles goed verloopt. Ik heb een conflict gezien

bij Jonathan en we hebben het nog niet op kunnen lossen, dus ik weet niet precies hoe dat in zijn werk gaat. Als iemand enig idee heeft dan hoor ik dat graag. Over het algemeen, als we goed communiceren over wie wat aan het doen is (gister middag had Jonathan tests geschreven en het bleek dat een groupmember unknowingly dezelfde tests had geschreven, don't let this happen....), over het algemeen zullen we dan geen conflicts ondervinden hoop ik. Ga niet al te lang zelf kутten...

Dat was t :).