252-0027-00: Einführung in die Programmierung

Übungsblatt 1

Abgabe: 3. Oktober 2023, 23:59

In der ersten Übung haben Sie ein Eclipse-Projekt als ZIP-Datei heruntergeladen und importiert. Von nun an werden Sie die Übungsprojekte aus einem sogenannten *version control system* (VCS) beziehen. Solche Systeme erlauben es einer Programmiererin, verschiedene Versionen desselben Programm-Codes zu verwalten und mit anderen Programmierern zusammenzuarbeiten.

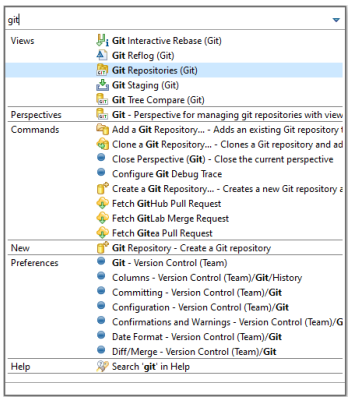
In dieser Vorlesung verwenden wir **Git** als Versionierungssystem. Sie können damit Ihre Fortschritte für eine Übung speichern und jederzeit zu früheren Versionen ihrer Lösungen zurückkehren. Die Dateien, die Sie ins System übertragen, werden auf einem Server an der ETH gespeichert.

Die Übungsabgabe läuft ebenfalls über das Git-System ab. Zum Abgabezeitpunkt wird die letzte Lösung, die Sie ins System übertragen haben, direkt an einen Assistenten geschickt.

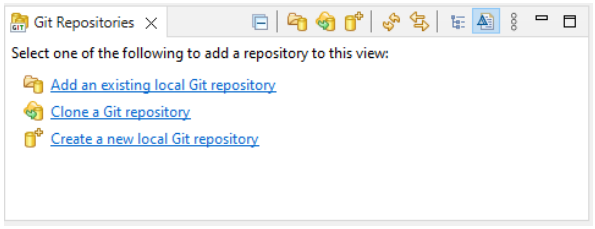
Aufgabe 1: Auschecken des Übungsprojekts

Git hat zwei Teile: den *Server*-Teil, der in unserem Fall auf einem Server an der ETH ausgeführt wird, und den *Client*-Teil, der auf Ihrem PC läuft. Falls Sie sich auf myStudies für die Vorlesung registriert haben, haben wir für Sie einen persönlichen Ordner auf dem Git-Server angelegt. Einen solchen Ordner nennen wir auch *Repository*. In Ihrem Repository befindet sich bereits das Projekt für diese Übung. Dieses werden Sie nun mithilfe ihres Git-Clients *auschecken* (Englisch *checkout*) und in Eclipse importieren.

1. Wählen Sie das QuickAccess Suchfenster oben rechts aus und suchen Sie nach "git". Dann wählen Sie im erscheinenden Menü unter *Views* die Option *Git Repositories (Git)*.



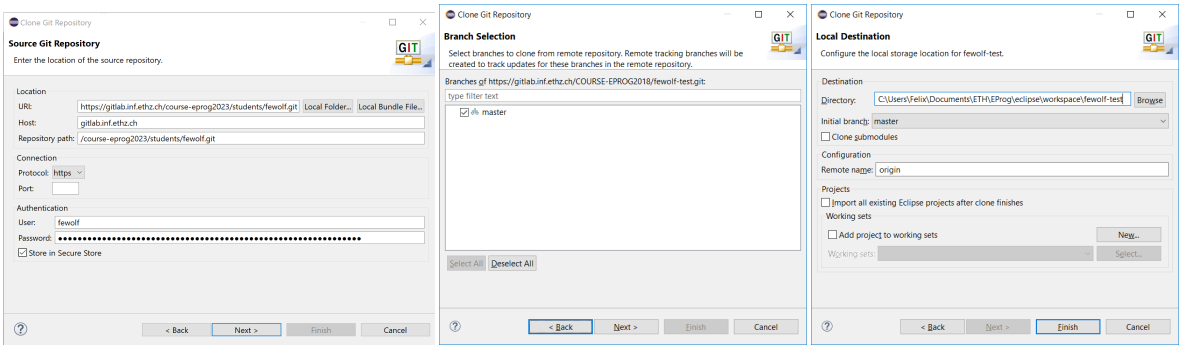
2. Nach dem vorherigen Schritt öffnet sich in Eclipse eine neue Ansicht. Dort wählen Sie *Clone a Git repository*.



3. Im erscheinenden Dialog wählen sie *Clone URI* und geben Sie im Dialog danach die URI zu Ihrem Repository ein. Sie lautet

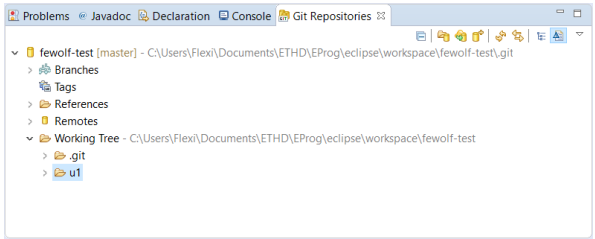
   `https://gitlab.inf.ethz.ch/course-eprog2023/students/<nethz-account>.git`

   wobei Sie `<nethz-account>` mit Ihrem N-ETHZ-Account ersetzen.



4. Zusätzlich werden Sie nach Benutzernamen (User) und Passwort gefragt. Dort geben Sie Ihren N-ETHZ-Account und das passende Passwort ein. Sobald alles eingetragen ist, klicken Sie zweimal auf *Next*.

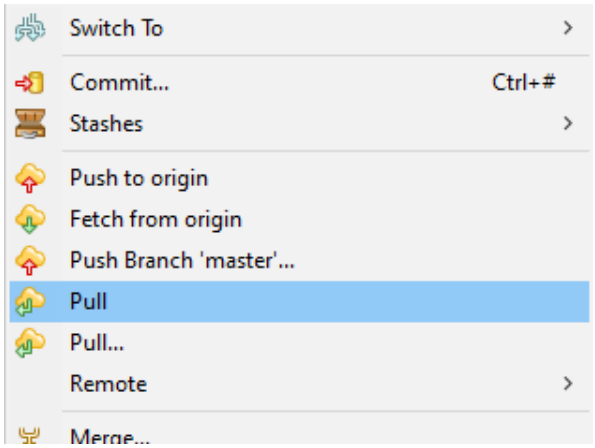
5. Auf der nächsten Seite können Sie den Ordner angeben, der für das Repository verwendet wird. Nachdem Sie sich für ein Ziel entschieden haben, klicken Sie auf *Finish* um den Checkout abzuschliessen. Hiernach sollten Sie in der zuvor erschienenen Ansicht Ihr Repository sehen.



In den nächsten Schritten werden Sie ein erstes Projekt aus Ihrem Repository in Eclipse importieren.

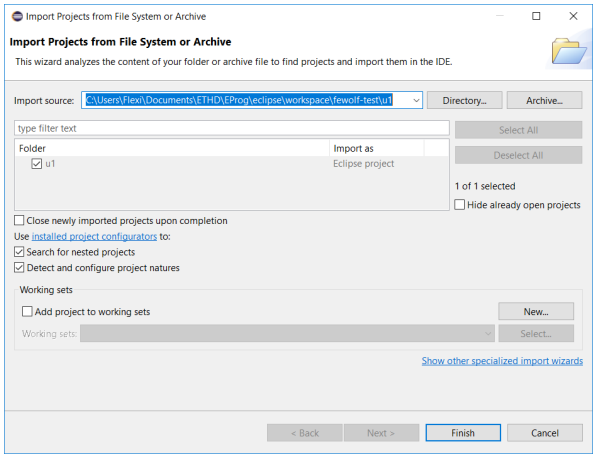
1. Zuerst rechtsklicken Sie auf Ihr Repository (der äusserste Menüeintrag mit *[main]* nach dem Namen, bei welchem im oberen Bild stattdessen *[master]* steht) und wählen *Pull*. Dies

garantiert, dass Sie an der neusten Version Ihres Repositories arbeiten. Sie machen das Gleiche, um neue Projekte und gegebenenfalls Feedback von Ihrem Assistenten zu erhalten.



2. Als nächstes doppelklicken Sie auf Ihr Repository und dann auf *WorkingTree* um zum ersten Projekt zu gelangen. Importieren Sie das Projekt durch Rechtsklick auf `u01` und Auswahl von *Import Projects...*

3. Im anschliessenden Dialogfenster wählen Sie *Finish* um den Import abzuschliessen.



Sie sollten nun links in Ihrem Package Explorer das Projekt "U01 <*N-ETHZ-Account*>" und darin die Dateien "HelloProgrammer.java" und "EBNF.txt" sehen.

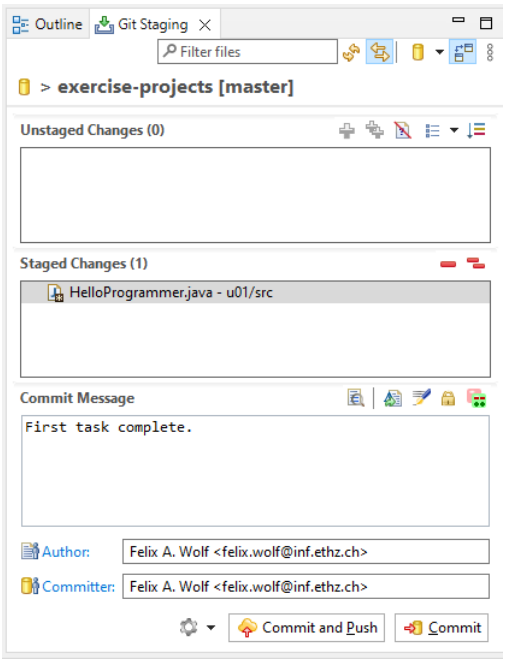
Aufgabe 2: Programm verändern

Verändern Sie das Programm, das Sie in der letzten Aufgabe ausgecheckt haben. Statt `"Hello Programmer"` soll Ihr Programm `"Hello, my name is` <*ihr Name*>`"` ausgeben (wobei wir es Ihnen überlassen, ob Sie Ihren vollständigen Namen, Ihren Vornamen, oder irgendeinen Spitznamen ausgeben). Ändern Sie dazu den Programmtext, speichern Sie die Datei und führen Sie das Programm erneut aus.

Ausserdem sollten Sie das Programm um Kommentare erweitern, wie in der Vorlesung erklärt (mindestens Author des Programms und Veranstaltung, für die Sie das Programm schreiben).

Aufgabe 3: Committen der Änderungen

Sie werden nun Ihre Änderungen zurück ins Git-Repository übertragen (auch: *committen* oder *einchecken*). Ein *Commit* besteht aus den Änderungen, die Sie an den Dateien im Repository vorgenommen haben (oder zusätzlichen Dateien, die noch nicht im Repository vorhanden sind), und aus einem Kommentar, der diese Änderungen beschreibt. Wenn Sie einmal eine frühere Version einer Datei oder eines Projekts anschauen möchten, sind gute Commit-Kommentare sehr hilfreich.



Um Änderung zu committen, rechtsklicken Sie in der *Git Repositories* Ansicht auf Ihr Repository und wählen Sie *Commit...*. Ziehen Sie mit Drag & Drop Ihre geänderten Dateien von *Unstaged Changes* zu *Staged Changes*. Nur die Änderungen unter *Staged Changes* werden Teil des Commits. Geben Sie unter *Commit Message* auch einen passenden Commit-Kommentar ein.

Mit *Commit and Push...* schliessen Sie den Commit ab und laden Ihre Änderungen auf den ETH-Server hoch. Falls darauf ein weiteres Fenster erscheint, klicken Sie dort *Preview* und dann *Push*. Sollten Sie auf *Commit* anstatt *Commit and Push...* klicken, wird Ihr Commit nicht auf den Server der ETH *gepusht*. D.h., dass Sie diesen Schritt nun manuell per Rechtsklick auf Ihr Repository und Auswahl von *Push* oder *Push to origin* nachholen sollten. Wenn wir in Zukunft von *committen* sprechen, dann ist implizit auch das *Pushen* des Commits auf die ETH-Server gemeint, da ansonsten ihr/e Assistent/in Ihre Änderungen nicht sehen kann (und Sie somit keine Punkte für Bonusaufgaben erhalten können).

Falls Sie zu einer früheren Version einer Datei zurückkehren möchten, rechtsklicken Sie auf die Datei und wählen Sie *Replace With* → *Commit...*. Sie sehen nun in der Ansicht die verschiedenen Versionen der Datei, inklusive Commit-Nachricht. Wählen Sie die gewünschte Version und wählen *Select* um den Vorgang abzuschliessen.

Machen Sie nun zur Übung noch ein paar weitere Änderungen am Programm und committen (und pushen) Sie sie abermals. Der letzte Push, den Sie vor dem Abgabetermin einer Übung vornehmen, bestimmt Ihre Abgabe.

xkcd: Git Commit by Randall Munroe (CC BY-NC 2.5, slightly modified)

Aufgabe 4: EBNF

In dieser Aufgabe erstellen Sie verschiedene EBNF-Beschreibungen. Alle Beschreibungen werden in der Text-Datei "EBNF.txt" abgelegt, welche sich in Ihrem "u01"-Ordner, bzw. im "U01 <*N-ETHZ-Account*>"-Projekt befindet. Sie können die Datei direkt in Eclipse bearbeiten.

Verwenden Sie "<=" als Zeichen für "ist definiert als" (d.h. ein < und ein =, ohne Zwischenraum). Der Name der letzten Regel ist durch die Aufgabenbeschreibung vorgegeben, andere Namen können Sie frei wählen. Da reine Textdateien keine Kursivschrift unterstützen, stellen wir die Namen von Regeln zwischen < und >, also z.B. `<name>`.

a) Erstellen Sie eine Beschreibung `<geradezahl>`, die als legale Symbole alle geraden Zahlen (d.h. Zahlen, die ohne Rest durch `2` teilbar sind) zulässt. Beispiele sind `+02`, `4`, `10`, `-20`.

b) Zeigen Sie in einer Tabelle, dass Ihre Beschreibung das Symbol "`28`" als gerade Zahl erkennt.

c) Erstellen Sie eine Beschreibung `<x2ygemischt>`, die als legale Symbole genau jene Wörter zulässt, in denen für jedes "`X`" zwei "`Y`" als Paar auftreten. Beispiele sind `XYY`, `YYX`, `XYYYYX`, `XXYYYY`.

d) Die folgenden EBNF-Beschreibungen sind nicht äquivalent. Finden Sie ein *kürzestmögliches* Symbol, das von der einen Beschreibung als legal erkannt wird, aber nicht von der anderen. (Fangen Sie mit einfachen Kombinationen von A und B an.)

   EBNF-Beschreibung: `<beispiel1>`
   `<beispiel1>` ⇐ `[A][B]`

   EBNF-Beschreibung: `<beispiel2>`
   `<beispiel2>` ⇐ `[A[B]]`

e) Erstellen Sie eine EBNF Beschreibung `<doppelt>`, die als legale Symbole genau jene Wörter zulässt, in denen die doppelte Anzahl "`Y`" nach einer Folge von "`X`" auftritt. Beispiele sind `XYY`, `XXYYYY`, usw.

Damit Ihr/Ihre Assistent/in Feedback zu Ihren Lösungen geben kann, committen Sie Ihre EBNF-Beschreibungen auf die selbe Weise, wie Sie Ihre Änderungen am Programm committet haben.