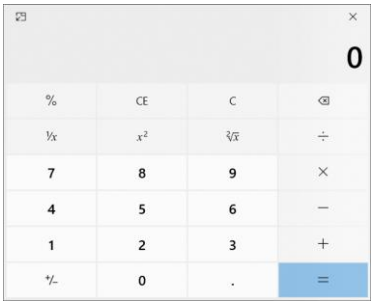
Teamaufgabe 2:
Agile Softwareentwicklung
Git, User-Stories, und KANBAN-Board

Jakob Droste, Jianwei Shi, Eklekta Kristo & David Sebode

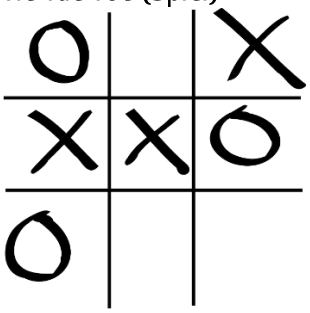
Abgabetermin: **16.01.2023 um 15 Uhr** auf StudIP-ILIAS

In dieser Teamaufgabe sollen Sie agile Entwicklungsmethoden in einem (Mini-) Projekt benutzen, um eine Java-Anwendung samt grafischem User Interface (GUI) zu entwickeln. Sie können dabei frei eine Anwendung aus der folgenden Liste auswählen:

Taschenrechner



Tic Tac Toe (Spiel)



Börsenkurse Beobachter

Quelle: Seite 101, Teil 2, Vorlesungsfolien

Bei nicht konkretisierten Informationen im Bild, interpretieren Sie diese bitte selbstständig. Ihr(e) Teamleiter*in entscheidet, welche Anforderungen umgesetzt werden sollen.

Arbeiten Sie mit Git

Erstellen Sie ein Git-Repository unter gitlab.uni-hannover.de und arbeiten Sie mit Git. Dabei sollen Sie mehrere Branches anlegen und verwenden. Es soll mindestens zwei Branches geben, z.B. einer davon für die Ablieferung (Delivery, normalerweise als *main* bezeichnet), der andere für die Entwicklung (Development, normalerweise als *dev* bezeichnet). Machen Sie regelmäßig und einzeln Commits, sodass kenntlich wird, wer was gemacht hat. Folgen Sie bei der Entwicklung den Java-Code-Konventionen von Google (https://google.github.io/styleguide/javaguide.html).

Benutzen Sie User Stories bzw. KANBAN-Board

Am Anfang des Projektes sollten Sie bereits die wichtigsten User Stories formuliert haben. Halten Sie sich dabei an die Formulierung, die Sie in der Vorlesung (Seite 61, Teil 1, Vorlesungsfolien) kennengelernt haben! Aktualisieren Sie diese regelmäßig und fügen Sie dem Projekt bei Bedarf neue User Stories hinzu. Wenn nötig, gruppieren Sie diese unter Epics. Geben Sie zusätzlich Aufwand (Story Points) für jede User-Story ein (Empfehlung: 1 bis 5). Stufen Sie zudem die Prioritäten der Stories mit A/B/C ein (A für sehr hoch, C für nicht so hoch). Alle User Stories befinden sich auf einem KANBAN-Board (Beispiel: Seite 52, Teil 3, Vorlesungsfolien). Hilfe dazu finden Sie im Beiheft (Seite 24-26).

Organisation

Sie haben einen Monat Zeit für die Aufgabe. Treffen Sie sich regelmäßig. 1 Mal pro Woche ist ausreichend. Nach jedem Treffen machen Sie einen Screenshot Ihres KANBAN-Boards. Protokollieren Sie den Stand des Boards in einem Bericht. Aufwand, Priorität, Inhalt der User Stories sollen auch angegeben werden, wenn diese Informationen nicht im Screenshot zu sehen sind.

Anforderung der Abgabe

Am Ende des Projektes sollen Sie einen Bericht (als **eine PDF-Datei**) hochladen, der folgende Informationen enthält:

- Auswahl der Anwendung (Rechner/Tic Tac Toe/Börsenkurse Beobachter).
- Übersicht des KANBAN-Boards nach jedem Treffen.
- Eine kurze textuelle Zusammenfassung über jedes Treffen.
- Ausgabe des *git log* Kommandos für alle Branches.
- *Ein Link zum Git-Repository (bitte stellen Sie diesen Repository als **veröffentlicht** ein).
- *Ein Link zu einem Demo-Video, das **Kernfunktionalitäten** der Software zeigt. Das Video dauert max. 3 Minuten.
  *: Sie können die Links im Feb. 2023 deaktivieren.

Im Zentrum stehen die Versionsverwaltung des Projekts, das Schreiben der User Stories und die Arbeit mit dem KANBAN-Board. Das über das Repository abgegebene Programm sollte funktionieren, muss aber nicht perfekt aufpoliert sein.

Eine Abgabe auf StudIP-ILIAS soll nur von **EINEM** Gruppenmitglied (dem/der Teamleiter*in) eingereicht werden. Auf StudIP-ILIAS werden Sie dazu aufgefordert **eine PDF**-Datei hochzuladen und Angaben (Name und Matrikelnummer) **ALLER** Teammitglieder (auch Teamleiter*in) zu geben. Eine Teamgruppe besteht aus 3 bis 5 Personen. Abgaben von kleineren oder größeren Gruppen (1 bis 2, über 5 Personen) werden NICHT bewertet. (Siehe auch Seite 5 von 1. ÜS für Organisation)

Kriterien für das Bestehen

Wenn alle der folgenden Punkte in Ihrer Abgabe erfüllt sind, bekommen Sie einen Bonuspunkt!

1. Satzstruktur der User Stories ist richtig (Sie bekommen dazu Feedback von Tutoren).
2. Sie arbeiten nach Prioritäten.
3. Sie haben die *.gitignore* Datei richtig benutzt (keine leere *.gitignore* Datei).
4. Git Commits kommen von jedem Teilnehmer.
5. Mehrere Branches wurden benutzt.
6. Die Software funktioniert.
7. Die Java-Code-Konventionen wurden befolgt (Sie bekommen für Naming-Konventionen Feedback von Tutoren).

Stellen Sie Ihre Fragen zu der zweiten Teamaufgabe gerne im Forum, damit wir sie direkt für alle beantworten können!

Zusätzliche Hinweise bzw. Quellen

Git

Einführung

git - Der einfache Einstieg https://rogerdudler.github.io/git-guide/index.de.html

Zweig (Branch)

About branches: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches

Git-flow-Workflow: https://www.atlassian.com/de/git/tutorials/comparing-workflows/gitflow-workflow

Offizielle Quelle

Buch: https://git-scm.com/book/en/v2

Dokumentation: https://git-scm.com/docs

Videos

Einführung zu Git und GitHub: https://www.youtube.com/watch?v=3ZIpJHZBbi8

Online Kurs zu Git von MIT Computer Science & Artificial Intelligence Lab, https://missing.csail.mit.edu/2020/version-control/

Mögliche Online Plattformen für KANBAN-Board:

https://padlet.com/

https://trello.com

Java Tutorial für GUI

Tipp: IntelliJ besitzt einen Scene Builder für XML bzw. FXML Dateien, d.h. man kann dort Interfaces für JavaFX erstellen, welchen man dann im Programm laden kann. Das macht das Erstellen der GUI deutlich einfacher, als diese von Anfang an zu programmieren. Als Student können Sie die Pro-Lizenzen für die Jetbrains Produkte kostenlos bekommen.

JavaFX: https://www.jetbrains.com/help/idea/javafx.html

Scene Builder: https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html

Video Erstellung Software

Erstellung

OBS Studio: https://obsproject.com/

Cloud-Dienst für Video-Datei

Laden Sie Ihr Video unter einem Cloud-Dienst hoch. Empfehlung: https://www.luis.uni-hannover.de/de/services/speichersysteme/dateiservice/cloud-dienste/seafile/

Bitte laden sie **keine Videos (!)** auf Ihren Git-Repository hoch.

Viel Erfolg mit der 2. Teamaufgabe!