SWE

Projekt Projektplanung

UNTERRICHTSEINHEIT: Projekt in SWE

Dauer: 2 Blöcke

Anteilig für Endnote : Teamnote für 2-3 Personen

Arbeitsweise: Gruppenarbeit

Fragen Hr. Röder im Unterricht.

ABGABE : Präsentation im Unterricht

* Ziel: Mit dieser Aufgabe sollen Sie das typische Arbeiten im Kleinteam erlernen und Ihr Verständnis von Dokumentation, Objektorientierter Programmierung und GIT vertiefen

## Allgemeine Informationen

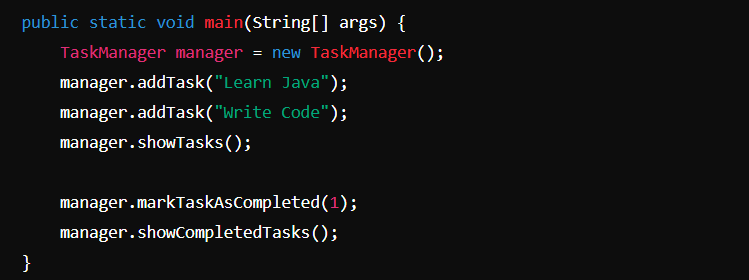
Zu lernen:   
GIT, Klassendiagramm, Javadoc und Kommentare, Aufgabenplanung, Zusammenarbeit und Onlinekommunikation.

Das Dokument

# Aufgaben:

### Projektaufgabe: Entwicklung einer einfachen To-Do Listen Anwendung in Java

#### Projektziel:

Entwickeln Sie gemeinsam eine einfache To-Do Listen Anwendung, die es Benutzern ermöglicht, Aufgaben zu erstellen, anzuzeigen und als erledigt zu markieren. Das Projekt soll vollständig geplant, entwickelt, dokumentiert und auf GitHub verwaltet werden.   
  
So soll später das Main aussehen, dass mit diesen Klassen arbeitet.   
  
**Projektphasen:**

* 1. **Planungsphase: Projektbesprechung:** Treffen Sie sich, um das Projekt zu besprechen und die Aufgaben zu klären. Es sollte dazu eine Aufgabenliste erstellt werden.   
     Diese ist als Datei Liste.txt (oder docx) später im Remote Repository sichtbar.  
     DOD1: Datei mit mindestens 10 Punkten steht im Rep.  
       
     Hinter jeder Aufgabe steht, wer das übernimmt. Es sollten mindestens (im Groben folgende Punkte aufgeteilt werden:
     1. Einer übernimmt die Programmierung und der andere kommentiert das Programmierte mit Kommentaren und Javadocs.
  2. Welche Tools brauchen Sie, wenn Sie an unterschiedlichen Rechnern zusammenarbeiten. Wie kommunizieren Sie online miteinander.  
     Stellen Sie vorab eine Liste der benötigten Tools für das Projekt zusammen  
     DOD2: Eine Datei mit der Bezeichnung Tools.txt steht im Rep.
  3. Erstellen Sie ein Klassendiagramm der 2 benötigten Klassen.
  4. **Aufgabenverteilung:** Teilen Sie die Aufgaben untereinander auf, z.B. wer sich um welche Funktionalitäten kümmert.
  5. **Erstellen eines Projektplans:** Erstellen Sie einen groben Zeitplan, der die verschiedenen Phasen des Projekts abdeckt.

1. **Einrichtung von Git und GitHub:**
   * **Repository anlegen:** Erstellen Sie ein neues Repository auf GitHub und fügen Sie beide Schüler als Kollaboratoren hinzu.
   * **Lokales Repository:** Klonen Sie das Repository auf Ihre lokalen Maschinen.
   * Sie können dies auch direkt in Eclipse oder Ihrer Entwicklungsumgebung machen.
   * **Erste Commit:** Erstellen Sie eine grundlegende Ordnerstruktur (z.B. src für den Quellcode, docs für die Dokumentation) und machen Sie den ersten Commit.
   * DOD: Screenshot des neuen Remote Repositories auf Github. Ablegen als Bild (Start.jpg) auf Github
2. **Entwicklung:**
   * **Klassenstruktur:** Entwerfen Sie eine einfache Klassenstruktur mit mindestens zwei Klassen: Task und TaskManager.
     + Task: Eine Klasse, die eine einzelne Aufgabe repräsentiert. Eigenschaften: id, description, completed.
     + TaskManager: Eine Klasse, die eine Liste von TASK verwaltet und Methoden zum Hinzufügen, Anzeigen und Markieren von Aufgaben als erledigt bietet.
     + DOD – Klassenstruktur fertig – Dem Dozenten zeigen
   * **Üben Sie abwechselndes Programmieren:**
     + Student A: Implementiert die Task Klasse.
     + Student B: Implementiert die TaskManager Klasse.
     + Student A: Erstellt Methoden zum Hinzufügen und Anzeigen von Aufgaben in TaskManager.
     + Student B: Erstellt Methoden zum Markieren von Aufgaben als erledigt und zum Anzeigen erledigter Aufgaben in TaskManager.
     + **Dokumentation:** Dokumentieren Sie den Code mit Javadoc-Kommentaren. Jeder Student dokumentiert die **vom anderen Studenten** implementierten Methoden und Klassen.  
       DOD: Dem Dozenten vorführen.
3. **Staging und Commit:**
   * Erzeugen Sie regelmäßige Commits und eventuell auch eigene Branches, die sie später wieder zusammenführen.  
     DOD: Vorführung merge von Branches
4. **Vorbereitung auf die Präsentation:**
   * **Vorführung vorbereiten:** Bereiten Sie eine kurze Präsentation (ca. 3 Minuten) vor, in der Sie die Zusammenarbeit zeigen. Einer Programmiert eine zusätzliche Methode (max 3 Zeilen Code), der andere Kommentiert das und nach dem Commit sollte es Online im Repository sichtbar sein.
   * DOD: Erfolgreiche Präsentation.