

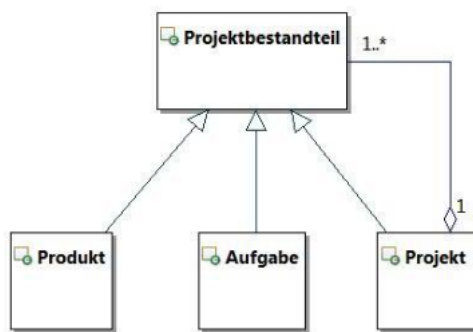
12. Übung

Abgabe: bis 18.02.2022

Aufgabe 1 - Klassen der Projektmanagement-Software

Entwickeln Sie eine sehr einfache Projektmanagement-Software mit mindestens den folgenden Klassen:

- Projekt:
 - Name (nicht leer)
 - Beschreibung
 - Stundensatz in Euro (> 0)
- Aufgabe:
 - Name (nicht leer)
 - Beschreibung
 - Aufwand in Stunden (> 0)
- Produkt:
 - Name (nicht leer)
 - Beschreibung
 - Produktionskosten in Euro (> 0)

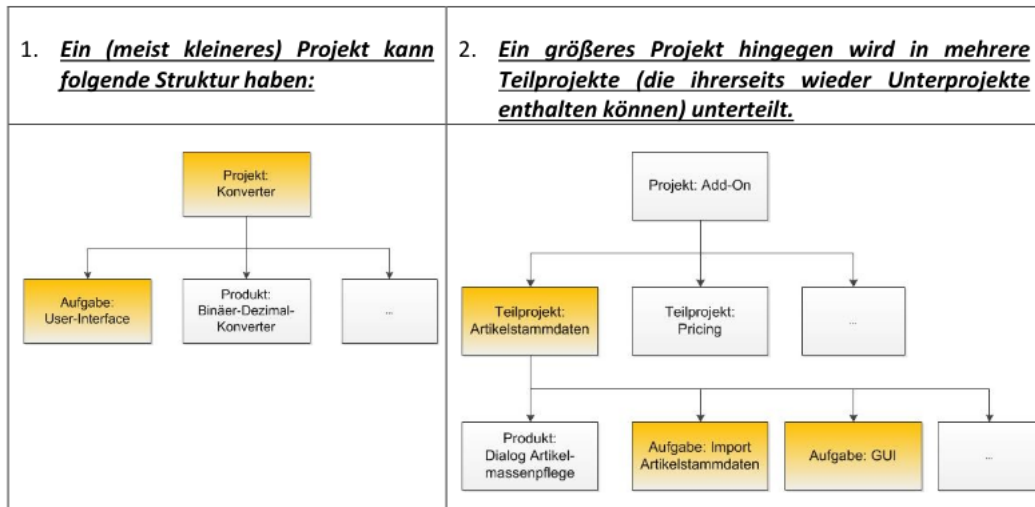


Ein Projekt kann Aufgaben, Produkte und Teilprojekte enthalten und verwalten, d.h. es müssen Projektbestandteile hinzugefügt oder gelöscht werden können. Dabei sollen die Projektbestandteile innerhalb der Klasse **Projekt** jeweils gleichbehandelt werden. Die Klassenhierarchie sollte sinnvollerweise so wie oben skizziert aussehen. Es gibt also eine abstrakte Oberklasse **Projektbestandteil**, von der **Produkt**, **Aufgabe** und **Projekt** erben. **Projekt** wiederum enthält ein Array mit Projektbestandteilen und verwaltet die Projektbestandteile. Somit kann ein **Projekt** auch weitere Projekte enthalten und so ergibt sich für ein **Projekt** eine Baumstruktur (siehe unten).

Jede der o.g. Klassen benötigt eine Methode zum Ermitteln der Kosten. Der **Projektbestandteil** ist als abstrakte Klasse zu realisieren, da die Kosten erst in Abhängigkeit der Bindung ermittelt werden können. **Projektbestandteil** besitzt also eine abstrakte Methode `berechneKosten()`, die die Kosten des jeweiligen Projektbestandteils berechnet und zurückgibt.

Die Kosten für ein **Projekt** ergeben sich folgendermaßen:

- Projektstundensatz * Stundenzahl der Aufgabe +
- Produktionskosten der Produkte +
- Kosten der Unterprojekte

Beispiele:**Aufgabe 2 - Steuerungsdatei**

Eine Steuerungsdatei enthält die Daten für einen Projektbaum. Lesen Sie eine Datei und berechnen Sie anschließend die Projektkosten. Die Datei kann Informationen über hinzuzufügende und zu löschende Projektbestandteile enthalten. Der Anfang einer Zeile gibt an, ob ein Projektbestand hinzugefügt (+) oder gelöscht werden soll (-). Wenn ein Projektbestandteil hinzugefügt werden soll, folgen in der Zeile die weiteren Informationen. Wenn ein Projektbestandteil gelöscht werden soll, wird einfach nur ein Minus-Zeichen und hinter dem Strichpunkt der Name des zu löschenden Bestandteils notiert.

Eine Zeile zum Hinzufügen eines neuen Projektbestandteils sieht folgendermaßen aus:

+ Art; name; beschreibung; aufwand; name uebergeordnetes Projekt

Bestandteile der Zeile:

- Art:
 - PROD: Produkt
 - AUF: Aufgabe
 - PROJ: Projekt
- name: Name des Projektbestandteils
- beschreibung: eine Beschreibung für den Projektbestandteil
- aufwand: Entweder ein Stundensatz, ein Aufwand oder die Produktionskosten
- name uebergeordnetes Projekt: Enthält den Namen des Projekts, zu dem die eingelesene Zeile gehört, falls es ein übergeordnetes Projekt gibt, sonst leer.

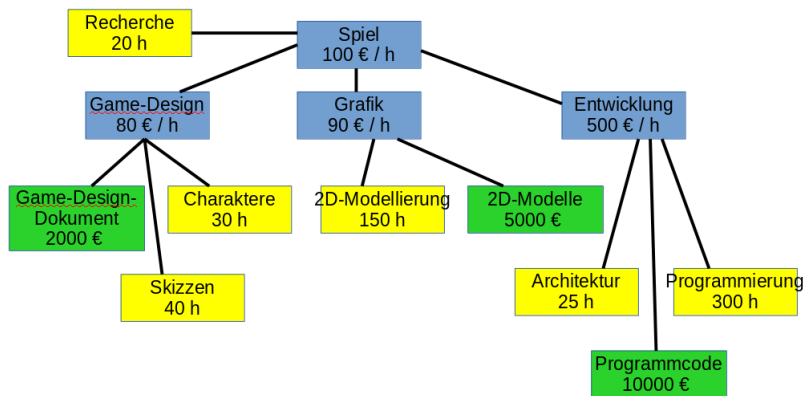
Eine Zeile zum Löschen eines neuen Projektbestandteils sieht folgendermaßen aus: - name

Beispiel für eine solche Datei für die Entwicklung eines Spiels:

- + PROJ; Spiel; Wir programmieren ein Spiel; 100;
- + AUF; Recherche; Zuerst wird recherchiert; 20; Spiel
- + PROJ; Game-Design; Game-Design entwickeln; 80; Spiel
- + PROD; Game-Design-Dokument; GDD schreiben; 2000; Game-Design
- + AUF; Charaktere; Die Charaktere werden geplant; 30; Game-Design
- + AUF; Skizzen; Zeichnen erster Skizzen; 40; Game-Design
- + PROJ; Grafik; Entwerfen der Grafik; 90; Spiel
- + AUF; 3D-Modellierung; 3D-Modelle entwerfen; 150; Grafik
- + AUF; 2D-Modellierung; 2D-Modelle entwerfen; 80; Grafik

- + PROD; 3D-Modelle; ; 10000; Grafik
- + PROD; 2D-Modelle; ; 5000; Grafik
- 3D-Modellierung
- 3D-Modelle
- + PROJ; Server; Server vorbereiten; 130; Entwicklung
- + AUF; ServerPlanen; ; 40; Server
- + AUF; ServerEinrichten; ; 80; Server
- + AUF; Zeichnen; ; 90; Grafik
- + PROJ; Entwicklung; Entwicklung des Spiels; 500; Spiel
- + AUF; Architektur; Architektur entwerfen; 25; Entwicklung
- + AUF; Programmierung; Programmcode schreiben; 300; Entwicklung
- + PROD; Programmcode; ; 10000; Entwicklung
- ServerPlanen
- ServerEinrichten
- Server

Aus dieser Datei ergibt sich dann letztendlich (nachdem auch die Löschvorgänge verarbeitet wurden) die folgende Projektstruktur:



Die Kosten dieses Projekts berechnen sich also wie folgt:

- $100 \text{ €} * 20 \text{ h} +$
- $80 \text{ €} * 30 \text{ h} + 80 \text{ €} * 40 \text{ h} + 2000 \text{ €} +$
- $90 \text{ €} * 150 \text{ h} + 5000 \text{ €} +$
- $500 \text{ €} * 25 \text{ h} + 500 \text{ €} * 300 \text{ h} + 10000 \text{ €}$

Aufgabe 3 - Protokoll

Alle Klassen implementieren eine to-String-Methode für die Ausgabe. Dabei wird nach jeder eingelesenen Zeile aus der Datei eine Ausgabe erzeugt, d.h. die gesamte bis zu diesem Zeitpunkt erzeugte Projektstruktur wird ausgegeben. Die Ausgabe wird in eine Datei geschrieben. Zusätzlich werden nach jedem Schritt die Gesamtkosten ausgegeben.

Eine Ausgabe kann bspw. wie folgt aussehen:

Nach der ersten Zeile aus der Steuerungsdatei:

Projekt Spiel, Stundensatz: 100 Euro

Kosten: 0 Euro

Nach der zweiten Zeile aus der Steuerungsdatei:

Projekt Spiel, Stundensatz: 100 Euro

Aufgabe Recherche 20 h

Gesamtkosten: 2000 Euro

Nach der dritten Zeile aus der Steuerungsdatei:

Projekt Spiel, Stundensatz: 100 Euro
Aufgabe Recherche 20 h
Projekt Game-Design, Stundensatz: 80 Euro
Gesamtkosten: 2000 Euro

Nach der vierten Zeile aus der Steuerungsdatei:

Projekt Spiel, Stundensatz: 100 Euro
Aufgabe Recherche 20 h
Projekt Game-Design, Stundensatz: 80 Euro
Produkt Game-Design-Dokument, Produktionskosten: 2000 Euro
Gesamtkosten: 4000 Euro

Aufgabe 4 - Aufruf

Das Programm wird folgendermaßen über die Kommandozeile aufgerufen: `javac Projektverwaltung steuerungsdatei.txt protokolldatei.txt`
Überprüfen Sie die Dateien ggf. auf Existenz, Lesbarkeit usw.