



Hausaufgabe 2

Hinweis: Für die Bearbeitung dieser Hausaufgabe ist Gruppenarbeit **nicht** zulässig! Bitte bedenken Sie dabei, dass Plagiate mit entsprechender Analysesoftware leicht zu erkennen sind. Bereits eine abgeschriebene Hausarbeit führt zum Nicht-Bestehen des gesamten Moduls. Unterschreiben Sie bitte auch die am Ende der Aufgabe angefügte Erklärung und reichen diese zusammen mit Ihrer Lösung ein. Bitte geben Sie diese Hausaufgabe **bis Sonntag, den 15.01.2023 um 23:59 Uhr** über ILIAS <https://lms.hs-osnabrueck.de/> ab. Ihre Lösung besteht dabei aus den folgenden **drei** Dateien: Klassendiagramm (als pdf, jpg oder png), Java-Archiv, Eidesstattliche Erklärung im PDF. Der Dateiname jeder Datei ist immer HA2_Nachname_Vorname.*

Aufgabe: Entwicklung eines Planungstools für Ihre Solarteur:in

- a) Erstellen Sie ein Klassendiagramm mit den benötigten Klassen und ihren Beziehungen. Geben Sie bei den verwendeten Assoziationen auch die jeweiligen Kardinalitäten an. (5 Punkte)
- b) Implementieren Sie die ToDo-Listen Anwendung in Java gemäß nachfolgender Anforderungen. (25 Punkte)

Funktionale Anforderungen

Das neue Jahr startet bald und viele Menschen haben den Vorsatz Ihr Leben besser zu organisieren. Genauso geht es auch Ihrer befreundeten Solarteur*in, für die Sie einen Algorithmus zur Planung von Photovoltaikanlagen auf Dachflächen in der ersten HA1 erstellt haben.

Im Rahmen der HA2 bittet Sie Ihre Solarteur*in um ein kleines Planungstool, womit er/sie die vielen Kundenaufträge verwalten möchte und immer einen Überblick hat, welches Kundenprojekt in welchem Zustand ist. Die Anwendung soll dabei **Solarinstallationsprojekte** (kurz **Projekte**) von **Kund:innen** umfassen. Projekte bestehen dabei aus **Aufgaben**. Ein Beispiel eines typischen Solarinstallationsprojekts besteht dabei oft aus den folgenden Aufgaben:

- Machbarkeitsanalyse
- Ortstermin bei Kund:in
- Angebotserstellung
- Materialbestellung
- Aufbau Gerüst
- Installation Solarmodule
- Anschluss an Stromnetz
- Übergabe der PV-Anlage an Kund:in
- Abbau Gerüst
- Rechnungsstellung
- ...



Im Folgenden sind die funktionalen Anforderungen des Planungstools aufgeführt:

- **[0] Menü:** Das Planungstool soll beim Start ein Menü anzeigen, dass die folgenden Einträge enthält:

1. Kund:in anlegen
2. Projekt anlegen
3. Projekt anzeigen
4. Aufgabe zu Projekt hinzufügen
5. Aufgabe in Projekt erledigen
6. Programm beenden

Durch die Eingabe der jeweiligen Nummer startet, dann die jeweilige Funktionalität. Zum Beispiel durch Eingabe der [1] kann eine neue Kund:in angelegt werden. Nach der Beendigung der aufgerufenen Funktionalität, wird das Menü wieder angezeigt.

- **[1] Kund:in anlegen:** Beim Anlegen einer Kund:in soll die Benutzer:in aufgefordert werden einen **Namen** und eine **Adresse** einzugeben. Zusätzlich soll noch automatisch eine Kundennummer vergeben werden. **Kundennummern** haben dabei die Form „K+<fortlaufende Nummer>“, also z.B. „K123“. Die **erste Kund:in soll die Nummer „K100“** haben. **Nach der Eingabe der Daten soll wieder das Hauptmenü angezeigt werden.**

Bitte beachten:** Obwohl Sie später das Anlegen von unbegrenzt vielen Kund:innen unterstützen wollen, werden in der ersten Version Ihres Planungstools nur zehn Kund:innen unterstützt. **Speichern Sie die Kund:innen dazu in einem Array mit zehn Feldern, das Sie in der main-Methode der Test-Klasse anlegen.

- **[2] Projekt anlegen:** Beim Anlegen eines Projektes soll die Benutzer:in aufgefordert werden einen Namen, einen Ort und ein geplantes Datum für die Installation der PV-Anlage einzugeben. Das Datum kann zur Vereinfachung als String gespeichert werden¹. Anschließend soll ein Kunde für das Projekt ausgewählt werden. Dazu sollen alle gespeicherten Kunden passend angezeigt werden. Durch die Eingabe der Kundennummer soll der Kunde dann dem Projekt hinzugefügt werden. Genau wie bei den Kunden soll zusätzlich noch automatisch eine Projektnummer vergeben werden. Projektnummern haben dabei die Form „P+<fortlaufende Nummer>“, also z.B. „P1012“. Das erste Projekt soll die Nummer „P1000“ haben.

Nach der Eingabe der Daten soll wieder das Hauptmenü angezeigt werden.

¹ Wer schon etwas mehr probieren mag, darf das Datum auch als LocalDate speichern. Eine Anleitung dazu finden Sie hier: https://openbook.rheinwerk-verlag.de/java8/05_009.html



Bitte beachten: Obwohl Sie später das Anlegen von unbegrenzt vielen Projekten unterstützen wollen, werden in der ersten Version Ihres Planungstools nur zehn Projekte unterstützt. Speichern Sie die Projekte dazu in einem Array mit zehn Feldern, das Sie in der main-Methode der Test-Klasse anlegen.

- **[3] Projekt anzeigen:** Wird dieser Menüeintrag ausgewählt, soll zunächst ein Überblick über alle Projekte angezeigt werden. Der **Überblick** hat dabei die **Form: <Projektnummer> + <Projektname>**. Anschließend soll die Benutzer:in eine **Projektnummer eingeben**. Für das ausgewählte Projekt sollen dann die **Detailinformationen in folgender Form** angezeigt werden:

Projekt „PV-Anlage Familie Isaak“

Kunde: Familie Isaak, Musterstr. 123, 12345 Musterstadt

Geplantes Installationsdatum: 15.01.2023

Aufgaben:

- 1. Machbarkeitsanalyse [Status: erledigt]**
- 2. Ortstermin bei Kund:in [Status: erledigt]**
- 3. Angebotserstellung [Status: offen]**

- **[4] Aufgabe zu Projekt hinzufügen:** Um eine Aufgabe zu einem Projekt hinzuzufügen, muss die Benutzer:in zunächst das **Projekt auswählen**. Dazu wird **wie bei [3] eine Übersicht der Projekte angezeigt**. Anschließend soll die Benutzer:in eine **Projektnummer eingeben**. Für das ausgewählte Projekt sollen **dann die Detailinformationen, wie bei [3] beschrieben, angezeigt werden**.

Mit der Detailansicht vor Augen, wird die Benutzer:in nun **aufgefordert eine neue Aufgabe zum Projekt hinzuzufügen**. Dazu muss der **Aufgabenname und eine optionale Beschreibung** eingegeben werden. **Neu angelegte Aufgaben haben immer den Status „offen“**. Eine neu angelegte Aufgabe ist danach in dem zuvor ausgewählten Projekt gespeichert.

Bitte beachten: Obwohl Sie später das Anlegen von unbegrenzt vielen Aufgaben in Projekten unterstützen wollen, werden in der ersten Version Ihres Planungstools nur zehn Aufgaben pro Projekt unterstützt.

- **[5] Aufgabe in Projekt erledigen:** Um eine Aufgabe in einem Projekt abzuhaken, muss die Benutzer:in, **wie bei [3] und [4] zunächst ein Projekt auswählen**. Dazu soll eine **Übersicht der Projekte** angezeigt werden. Anschließend soll die Benutzer:in eine **Projektnummer eingeben**. Für das **ausgewählte Projekt** sollen dann die **Detailinformationen, wie bei [3] beschrieben, angezeigt werden**.



In der **Detailansicht** sind dann die Aufgaben des Projektes direkt ersichtlich und die **Benutzer:in** muss nun die Nummer der Aufgabe eingeben, die erledigt werden soll. Nach Eingabe der Nummer ist der **Status der Aufgabe „erledigt“**.

- **[6] Programm beenden:** Ja, genau, hier beenden Sie das Programm 😊

Außerdem berücksichtigen Sie folgende **nicht-funktionale Anforderungen**:

- Klassen sollen möglichst wenige Details ihrer Oberklassen „kennen“.
- Vermeiden Sie die mehrfache Implementierung „identischen“ Codes.
- Alle Objektvariablen sollen die Sichtbarkeit **private** haben.
- Zur Verwaltung von Mengen von Objekten (Kunden/Projekte/Aufgaben) sollen **Arrays** fester Länge verwendet werden.
- Fangen Sie wo nötig Fehleingaben bei der Eingabe ab (try/catch).
- Lehnen Sie eine nicht plausible Auswahl mit einer entsprechenden Meldung ab.
- Wenn mehr als 10 Kunden/Projekte/Aufgaben angelegt werden, geben Sie einen passenden Warnhinweis aus.

Viel Spaß und gutes Gelingen! 😊



Hausaufgabe 2 im Modul Objektorientierte Programmierung 1

Name: _____

Vorname: _____

Matrikel-Nr.: _____

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Hausarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ort, Datum

Unterschrift

Speichern Sie die ausgefüllte und unterschriebene Erklärung mit dem Dateinamen "HA2_Nachname_Vorname_Erklärung.pdf" und reichen Sie diese zusammen mit Ihrer Lösung ein.