

# Softwaretechnik

## Aufgabenblatt Nr. 7

### A1 Hybrid Mobile Application

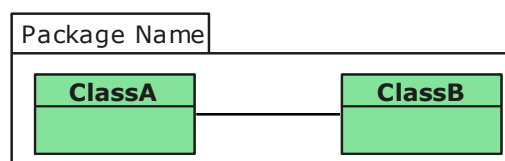
Sie wollen objektorientiert ein hybrides Application Framework entwickeln, mit dem Sie einheitlich graphische Benutzerschnittstellen in Android und iOS steuern können. Sie benötigen die folgenden Komponenten:<sup>1</sup>

- Interface *MobileElement* mit dem Attribut *title* sowie den Operationen *render()* und *register(MobileForm form)*.
- Generalisierung *MobileForm* mit folgenden Klassenmitgliedern:
  - Attribut *title*.
  - Liste der enthaltenen mobilen Elemente (als Teile-Ganze-Beziehung).
  - Operation *addElement(string title)* zur zentralen Erzeugung von neuen mobilen Elementen. Während der Erzeugung wird die erzeugende *MobileForm* registriert.
  - Operation *render()*, die die gleichnamige Operation in allem enthaltenen mobilen Elementen aufruft.
- Von *MobileForm* selbst sollen keine Instanzen erzeugt werden können.
- Konkrete Elemente und konkrete Formen für Android und iOS, die erzeugt werden können. Die Betriebssysteme einer konkreten Form und den darin enthaltenen konkreten Elementen passen immer zusammen.
- Alle oben genannten Klassen sind im Paket *UserInterface* enthalten. Aus Sicherheitsgründen sollen neue konkrete Elemente von außerhalb des Pakets nur über die Operation *addElement* von *MobileForm* erzeugt werden können.
- Die Klasse *Application* im Paket *Mobile* mit den folgenden Eigenschaften:
  - Attribut *form* vom Datentyp *MobileForm*.
  - Private Operation *getOS()*, die das aktuelle Betriebssystem als String zurückgibt (Stringwerte »Android« und »iOS«).
  - Private Operation *initialize()*, die dem Attribut *form* eine neue Instanz passend zum Betriebssystem zuweist.
  - Operation *run()*, die *initialize()* aufruft, drei neue Elemente der konkreten Form hinzufügt und anschließend alle Elemente rendert.

Aufgaben:

- a) Modellieren Sie den Klassenverband als UML-Klassendiagramm.<sup>2</sup> Nutzen Sie für Ihre Lösung das Entwurfsmuster *Factory Method*.
- b) Modellieren Sie die Abfolge der Aufrufe als UML-Sequenzdiagramm. Beginnen Sie mit der Operation *run()* von *Application* als gefundene Nachricht. Nutzen Sie dabei die abstrakten, und nicht die konkreten Klassen.
- c) Implementieren Sie den Rumpf der Lösung in einer geeigneten objekt-orientierten Programmiersprache Ihrer Wahl, z.B. Java oder C#.

*Hinweis:* Pakete und darin enthaltene Klassen werden in UML wie folgt modelliert



<sup>1</sup> Wenn nicht explizit angegeben, dann sind die genannten Klassenmitglieder öffentlich.

<sup>2</sup> Sie können bei der Modellierung auf die Angabe der Getter und Setter verzichten.

## A2 Online-Reservierungssystem für Impfungen

Sie erheben die Anforderungen für ein Online-Reservierungssystem für Impfungen, siehe Aufgabe A5 auf Aufgabenblatt Nr. 4. Sie wollen nun auf fachlicher Ebene den Nachrichtenfluss zwischen Patienten und Web-Anwendung modellieren. Sie haben bereits die folgenden Komponenten identifiziert:

- Der Patient als menschlicher Akteur
- Die Web-Anwendung als GUI
- Ein Identitätsmanagement als Hintergrunddienst
- Ein Termin-Verwaltungssystem als Hintergrunddienst

Der reguläre Nachrichtenfluss zwischen diesen Komponenten gestaltet sich wie folgt:

- Der Patient besucht die Web-Anwendung und meldet sich mit seinem Benutzernamen und Passwort an (Sie können davon ausgehen, dass der Patient bereits registriert ist).
- Die Webseite prüft die Gültigkeit der Anmeldung im Identitätsmanagement.
- Bei ungültigen Anmeldedaten wird dem Patienten durch die Web-Anwendung angeboten, sein Passwort zurückzusetzen. Dazu wird eine entsprechende Operation im Identitätsmanagement aufgerufen. Der Nachrichtenfluss endet dann.
- Ansonsten ruft die Web-Anwendung die möglichen Wochentage für eine Impfung vom Termin-Verwaltungssystem ab und zeigt diese an.
- Der Patient wählt einen der möglichen Tage aus. Die Web-Anwendung ruft dann die möglichen Uhrzeiten vom Termin-Verwaltungssystem ab und zeigt diese an.
- Der Patient wählt eine Uhrzeit aus und bestätigt die Buchung. Dadurch wird der Termin im Termin-Verwaltungssystem reserviert und im persönlichen Kalender im Identitätsmanagement eingetragen.
- Falls bei der Buchung angegeben, erhält der Patient optional eine E-Mail mit dem gebuchten Termin. Diese wird vom Identitätsmanagement gesendet.

Modellieren Sie den Nachrichtenfluss zwischen Patienten und den beteiligten Systemen durch ein UML-Sequenzdiagramm.