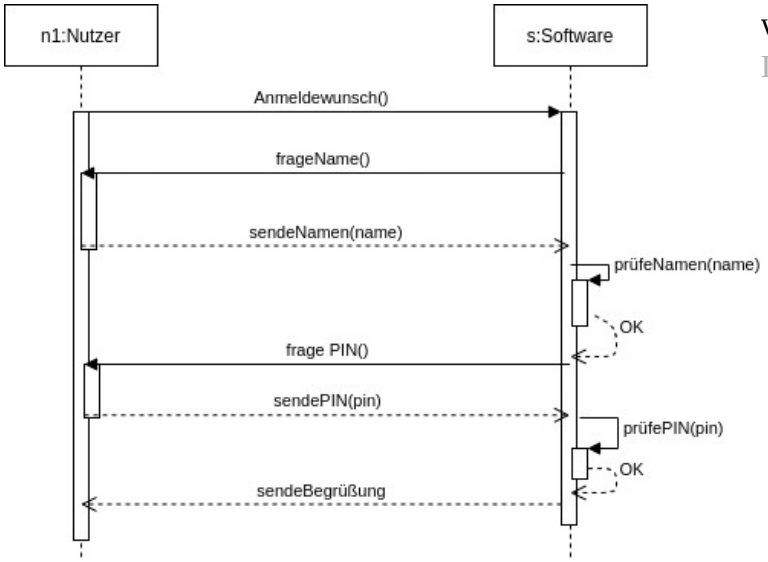
Arbeitsauftrag 2

Lösungsvorschlag

für die 4. Stunde am Freitag dem 17. Juli 2020

(Abgabe bis Montag den 20. Juli, 12:00 Uhr)

Wiederholung:



w1:

Im nebenstehenden Diagramm:

a. wie lauten die Namen der beteiligten Klassen und wie die Namen der beteiligten Objekte

Name	Klasse
n1	Nutzer
s	Software

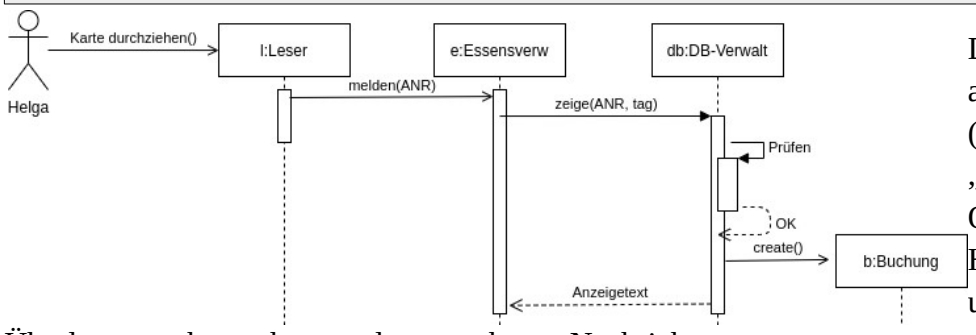
b. Beschreiben Sie kurz den Vorgang den das Diagramm modelliert.

Nutzer n1 möchte sich bei der Software s anmelden. Diese fragt ihn zuerst nach seinem Namen und prüft diesen mit OK, dann fragt sie nach der PIN und prüft diese mit OK. Daraufhin sendet die Software eine Begrüßung.

w2:

Identifizieren Sie in folgendem Szenario die „handelnden Klassen“ und erstellen Sie ein Sequenzdiagramm: (Helga kann als „Actor“ betrachtet werden)

Die Schülerin Helga holt bei bomfood Essen ab. Sie zieht die bomCard durch den Leser. Der Leser meldet die Ausweis-Nr. an die Essensverwaltung. Mit Tagesdatum und Ausweis-Nr fordert die Essensverwaltung die Datenbankverwaltung auf, Helgas Tagesbestellungen auf dem Bildschirm zu zeigen. Die Datenbank prüft mit eigenen Methoden die übergebenen Daten, erstellt eine Buchung "abgeholt" und gibt die Bestellung auf dem Bildschirm aus.



Das Diagramm kann auch anders aussehen (kein eigenes Objekt „Buchung“, ein eigenes Objekt „Bildschirm, Helga nicht als Actor, usw); wichtig die Überlegung, ob synchrone oder asynchrone Nachricht

Bei konkreten Fragen → Mail an leberecht@cvb-schule.de

w3:

Welche Aussagen sind richtig? Geben Sie in Stichpunkten an, was bei den nicht richtigen falsch ist:

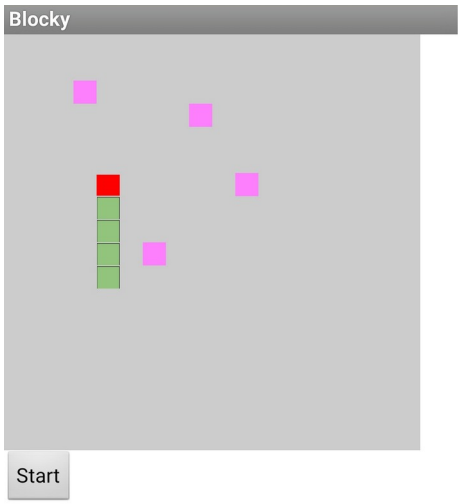
a. Eine asynchrone Nachricht heißt so, weil die Antwort etwas später kommt.
   Nein, asynchrone Nachrichten erfordern **keine** Antwort

b. Eine Nachricht wird programmtechnisch so umgesetzt, dass eine eigene Methode aufgerufen wird.
   Nein, es werden **Methoden des Zielobjekts** aufgerufen

c. Jedes Objekt im Diagramm muss erzeugt werden, bevor es Nachrichten versenden kann.
   Jein, ein sendendes Objekt muss existieren, aber **nicht explizit erzeugt** werden, es kann schon „einfach da“ sein.

d. Sendet ein Objekt eine Nachricht an sich selbst, wird dies als zweiter Aktivitätsbalken direkt am Aktivitätsbalken wiedergegeben.
   Ja

Arbeiten Sie im Dokument „*UML_Klassendiagramm.pdf*“ nochmal den Teil über „Beziehungen von Klassen“ durch und lösen Sie dann die Aufgaben.



Die Abbildung zeigt eine AppInventor-App ähnlich dem Spiel „*Snake*“. Auf dem Bildschirm (Screen1 - weißer Hintergrund) kann ein Spieler eine Schlange über das Spielfeld (Canvas1 - hellgrauer Hintergrund) bewegen. Die Schlange besteht aus einzelnen ImageSprites. Ein ImageSprite mit dem roten Quadrat als *picture*-Attribut stellt den Kopf Kopf dar, ImageSprites mit einem grünen Quadrat als *picture*-Attribut sind der Rumpf.

Die vier einzelnen ImageSprites mit dem lila Quadrat als *picture*-Attribut sind „Gift“, das nicht berührt werden darf

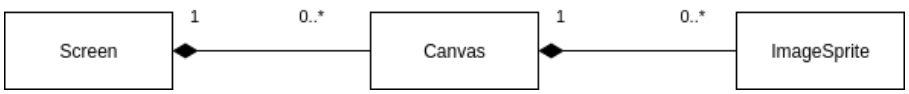
A1:

Betrachten Sie die Schlange, den Kopf und die Rumpfteile jeweils als eigene Klasse und modellieren Sie in einem vereinfachten Diagramm (ohne Attribute und Methoden) die Assoziation der drei Klassen mit Angabe der Multiplizitäten.

„Diamonds“ müssten nicht sein, die „4“ würde ich aber erwarten

A2:

Ein *Canvas* kann nur „in“ einem *Screen* existieren und *ImageSprites* lassen sich nur „in“ einem *Canvas* plazieren. Stellt die Beziehung dieser drei Klassen eine Aggregation oder sogar eine Komposition dar? (Begründung) Zeichnen Sie in einem vereinfachten Diagramm (ohne Attribute und Methoden) diese Beziehung mit Angabe der Multiplizitäten.



Die einzelnen Bestandteile sind „Existenzabhängig“ vom Ganzen, also eine Komposition