

Portfolioprüfung

Teilaufgabe 1

Zimmermann, Hufenbach, Himmelspach, Leneweit

Nordakademie

8. August 2018



Erstellen Sie

- ▶ ein neues Projekt (Numbers)
- ▶ in diesem eine Klasse mit dem Namen `Numbers`
- ▶ einen Konstruktor in der Klasse `Numbers`
- ▶ eine Methode (`fibonacciS`), die die Fibonacci Zahl für eine übergebene, nichtnegative ganze Zahl berechnet und diese an den Aufrufer zurück gibt. Die Methode soll eine Schleife verwenden.
- ▶ eine Methode (`fibonacciR`), die die Fibonacci Zahl für eine übergebene, nichtnegative ganze Zahl berechnet und diese an den Aufrufer zurück gibt. Die Berechnung soll über rekursive Aufrufe erfolgen.

- ▶ Benutzen Sie die Definition:

$$\begin{aligned} f_0, f_1 &=_{Def} 1 \\ \forall n \in \mathbf{N}_0 : f_{n+2} &=_{Def} f_{n+1} + f_n \end{aligned}$$

- ▶ berechnen Sie `fibonacciS(63)` und `fibonacciR(63)` und dokumentieren das Ergebnis in einem Methodenkommentar.
- ▶ eine Methode (`roman`), die eine übergebene ganze und positive Zahl ($0 < n \leq 3999$) in die entsprechende römische Zahl überführt und diese als Zeichenkette an den Aufrufer zurück gibt. Beispiel: `roman(1999)` gibt den String "MCMXCIX" zurück.

1. Programmieren Sie in einem neuen Projekt ein Tamagotchi, das Sie glücklich machen sollen.
 - 1.1 Das kann der Tamagotchi: `play`, `eat`, `sleep` und `pet`
 - 1.2 Der Status des Tamagotchi wird durch die `int` (Exemplar-)Variablen `hunger`, `mood` und `fatigue` beschrieben.
 - 1.3 Für diese Variablen gibt es Schwellwerte `hungerLimit`, `moodLimit` und `fatigueLimit`, oberhalb derer Ihr Tamagotchi als `hungry`, `happy` und `tired` gilt. Diese Schwellwerte für `hunger`, `mood` und `fatigue` werden für jedes neue Tamagotchi beim Erzeugen im Konstruktor übergeben und unveränderlich festgehalten.
 - 1.4 `hunger`, `mood` und `fatigue` sind selbst initial 0.
 - 1.5 Mit der Methode `getStatus()` können Sie das Tamagotchi um seinen Zustand befragen und erhalten einen `String` zurück (Siehe Punkt 3).

2. Die Business Methoden verändern den Zustand des Tamagotchis gemäß folgender Tabelle. Die angegebene Veränderung erfolgt nur, wenn die Bedingung erfüllt ist.

	Bedingung	hunger	mood	fatigue
play	not hungry	+2	+2	+3
eat	not tired	-3		+2
sleep	hungry	+1	-1	=0
sleep	not hungry	+1	+1	=0
pet		+1	+1	

3. Der Status eines Tamagotchi kann abgefragt werden. Er ist einer der folgenden Zustände, die als String zurückgegeben werden:
- ▶ "tired"
 - ▶ "hungry"
 - ▶ "happy"
 - ▶ "indifferent"

Liegen mehrere Zustände (siehe 1.3) vor, wird nur der mit der höchsten Priorität angegeben. Dabei tired wichtiger als hungry und hungry als wichtiger happy. Liegt keiner der Zustände tired, hungry und happy vor, dann ist der Zustand indifferent.

4. Erzeugen Sie ein neues Tamagotchi (welches nicht happy ist!). Bringen Sie ihr Tamagotchi durch essen, schlafen, spielen und streicheln in den happy Zustand. Setzen Sie dabei die Schwellwerte für hungry, tired und happy so, dass es wenigstens einmal essen und einmal schlafen muss. Dokumentieren Sie die dazu notwendigen Methodenaufrufe (Tipp: denken Sie an die Protokollierungsfunktion der Konsole). Fügen Sie die notwendigen Methodenaufrufe in die README.TXT Datei ein, die in Ihrem BlueJ-Projektordner liegt.

1. Speichern Sie Ihre Projekte als jar Dateien
<xxx>_<yyy>_Numbers.jar und <xxx>_<yyy>_Tamagotchi.jar
und achten Sie darauf, dass der Source Code in den .jar Dateien
enthalten ist. <xxx> und <yyy> stehen dabei für Ihre Nachnamen.
2. Laden Sie die beiden Dateien in moodle hoch.
3. Geben Sie sich high-five und gehen Sie zum gemütlichen Teil des
Abends über.