

6.2.6 Übungen zur OOP

Mittwoch, 1. März 2023 11:02

6.2.6. Übungsaufgaben zur OOP

1. Mobile Schrittzähler-Apps sind spätestens seit dem Erscheinen von Smartwatches durchaus

populär. Schreiben Sie ein Programm, das jeden Schritt zählen und anschließend die gezählten Schritte mit dem Datum ausgeben kann. Zwar können wir am Computer die gegangenen Schritte aktuell nur simulieren, aber praktisch lässt sich das Grundgerüst auch auf eine mobile App mit echter Schrittzahlung übertragen.

Aufgabenstellung

- Schreibe eine Klasse mit dem Namen StepCounter, dessen Konstruktor ein Datum in Form eines Strings entgegennehmen soll.
- Schreibe in der Klasse die Methode incrementSteps(), die den Schrittzähler um einen Schritt erhöhen soll.
- Die Methode toString() soll anschließend eine Ausgabe in dieser Form erzeugen: Am <Datum> bin ich <Schrittzahl> Schritte gegangen. Die umklammerten Worte sind Platzhalter für die entsprechenden Werte.
- Schreibe anschließend ein Testprogramm außerhalb der Klasse, welches den Schrittzähler testet und eine bestimmte Anzahl an Schritten (0-1100) simuliert.

Testfälle

■ Datum: 11.11.2020

Anzahl der Schritte: 1111

Ausgabe:

Am 11.11.2021 bin ich 1111 Schritte gegangen.

■ Datum: 1.9.2021

Anzahl der Schritte: 10000

Ausgabe:

Am 1.9.2021 bin ich 10000 Schritte gegangen.

2. Nochmal der Body-Mass -Index – diesmal objektorientiert

Recht am Anfang des Jahres haben wir uns bereits mit dem Body-Mass-Index beschäftigt. Diesmal wollen wir ihn objektorientiert programmieren!

Der BMI ist ein Maß für das Gewicht in Relation zur Körpergröße, er wird wie folgt berechnet (Gewicht in kg und Körpergröße in cm): $bmi = \text{gewicht} * 10000 / \text{groesse}^2$

BMI-Index	Frauen	Männer
Untergewicht	< 19	< 20
Normalgewicht	19 – 24	20 – 25
Übergewicht	25 – 30	26 – 30
Behandlungsbedürftiges Übergewicht	> 30	

Aufgabenstellung

Schreiben Sie eine Klasse Health, die den BMI für ein gegebenes Körpergewicht und die Körpergröße berechnet.

Bestimmen Sie mit dem BMI-Wert die Gewichtskategorie und konstruieren Sie folgende Konsolenausgabe:

Mit einem BMI von <BMI> sind Sie <Kategorie>.

Die spitzen Klammern sollen mit den entsprechenden Werten ersetzt werden.

Testfall:

■ Gewicht: 75 kg, Größe: 1,80m

Name: Markus

Mit einem BMI von 23,15 sind Sie untergewichtig.

■ Gewicht: 48 kg, Größe: 1,60m

Name: Lisa

Mit einem BMI von 18,75 sind Sie untergewichtig.

3. Entwerfen Sie den Code für eine Bankkontenklasse, die sich wie folgt nutzen lässt:

```
k1=Konto("Michael",200, 0)
k2=Konto("Maria")
k3=Konto("Peter", 1000, 500)
k1.einzahlen(100)
k2.abheben(100)
k3.abheben(1200)
for i in [k1,k2,k3]
    i.anzeigen()
```

Das Programm soll zu folgender Ausgabe führen:

>>>Danke für die Einzahlung. Diese wird auf Ihrem Konto verbucht.

>>>Sie können Ihr Konto nicht soweit überziehen

>>>1200 Euro wurden abgehoben

Konto von Michael:

Guthaben: 300

Überziehungsrahmen: 0

Konto von Maria:

Guthaben: 0

Überziehungsrahmen: 0

Konto von Peter:

Guthaben: -200

Überziehungsrahmen: 500

4. Rechenquiz

Ihr kleiner Cousin Malte hat im letzten Rechentest wieder eine 6 geschrieben, da er sich mit dem Kopfrechnen schwer tut. Sie wollen Malte helfen und ihm ein Programm schreiben mit dem er seine Rechenfähigkeiten trainieren kann. Das Programm soll zufällig Aufgaben erzeugen, die auf dem großen Einmaleins basieren (also ganze Zahlen zwischen 1 und 20).

Aufgabenstellung:

Schreiben Sie eine Klasse mit Namen MultiplicationQuiz, deren Objekte zufällige Rechenaufgaben erzeugen, vom Nutzer eine Eingabe einfordern und anschließend das korrekte Ergebnis dazu bereitstellen. Zudem soll angegeben werden, ob der Nutzer eine richtige oder falsche Eingabe gemacht hat.

Die Klasse soll z.B. so verwendet werden können:

```
mq=MultiplicationQuiz()  
mq.getExercise()
```

Testfälle:

Das Programm erzeugt bei jedem Aufruf von `getExercise()` eine neue Aufgabe.

Aufruf1:

13*8=?

Ergebnis: 104

Aufruf2:

7*5=?

Ergebnis: 35

Testfälle:

Das Programm erzeugt bei jedem Aufruf von `getExercise()` eine neue Aufgabe.

Aufruf1:

13*8=?

Ergebnis: 104

Aufruf2:

7*5=?

Ergebnis: 35

Für schnelle Programmierer:

Bauen Sie für Malte das Programm so um, dass er 10 Aufgaben beantworten muss. Am Ende wird angezeigt wie erfolgreich er war. Hat er unter 6 richtige Antworten gegeben, soll erscheinen: „Du musst mehr üben!“, zwischen 6 und 9 richtigen Antworten soll erscheinen: „Weiter so!“ und bei voller Punktzahl: „Perfekt!“.

4. Für mehr Herausforderung: Wasseruhr

Im Internet gibt es Programme, die dem Nutzer ermöglichen eine Zeit einzustellen und ihn nach Ablauf der Zeit daran erinnern, dass er ein Glas Wasser trinken soll. Das soll helfen, dass tägliche Trinkpensum zu erreichen.

Schreiben Sie auch ein solches Programm mit Hilfe der Objektorientierung. Der Nutzer soll eine Zeit eingeben können (z.B. 0:05).

Das Programm soll dann

0:04

0:03

0:02

0:01

Zeit Wasser zu trinken

ausgeben.

Informieren Sie sich hierfür über das Modul `time` (`import time`) sowie die Funktion `time.time()` des Moduls.

Optional können Sie bei der Ausgabe „Zeit Wasser zu trinken“ ein Signal ausgeben lassen. Hier eignet sich beispielsweise das Modul `winsound` (`from winsound import Beep`).