

## Hausaufgabe 1

Abgabe: 05.11.2023 (23:59 Uhr)

### **Abgabe:**

Für jede Aufgabe ist genau eine .java Datei im Moodle hochzuladen, insgesamt also **diesmal drei Dateien**. Jedes Programm muss für sich lauffähig sein und arbeitet mit Konsolenein- und ausgaben. Nehmen Sie, sofern angebracht, den Quellcode aus der SU als Basis.

### **Aufgabe 0:** ForeverYoung (Lernziel: Verzweigung und bestehenden Code ergänzen)

Ergänzen Sie die Datei ForeverYoung.java (Download aus Moodle) an der Stelle der Kommentare so, dass

- bei Eingabe einer Zahl kleiner 10 Folgendes auf der Konsole ausgegeben wird: „Echt jetzt? Wow.“
- bei Eingabe einer Zahl größer oder gleich 10 und kleiner oder gleich 20 „*Midlife-Crisis ist für Dich ein Fremdwort, nehme ich an.*“ auf der Konsole ausgegeben wird.
- bei Eingabe einer Zahl größer als 20 „*Weisheit reift mit dem Alter, schätze ich.*“ auf der Konsole ausgegeben wird.

### **Aufgabe 1:** PerfectGroups (Lernziel: Zerlegung in Teilschritte, komplettes Programm nach EVA-Prinzip)

Schreiben Sie ein Programm *PerfectGroups*, das eine Zahl  $n$  von der Konsole einliest. Die Zahl  $n$  steht für eine Anzahl an Personen. Das Programm soll ausgeben, in wie viele 2er-Gruppen diese Anzahl Personen aufgeteilt werden (die letzte Gruppe ist ggf. nicht ganz mit zwei Personen voll). Ist der Rest der Division von  $n$  durch 2 gleich 0, soll zusätzlich „*Perfekte Aufteilung*“ ausgegeben werden.

### **Aufgabe 2:** MiniDonuts (Lernziel: Zerlegung in Teilschritte, mehrere Fälle in richtiger Reihenfolge behandeln)

Schreiben Sie ein Programm *MiniDonuts*, das die Zahlen *small*, *big* und  $n$  von der Konsole einliest. Die Zahl  $n$  steht für die Anzahl an Donuts, die Sie einkaufen wollen. Der Supermarkt, welchen Sie besuchen, hat im Regal aktuell noch *big* Großpackungen (jeweils 5 Minidonuts pro Pack) und *small* Einzelpackungen (jeweils 1 Minidonut pro Pack). Da die Großpackungen günstiger sind, wollen Sie gerne so viele davon nehmen wie möglich und dann nur so viele Einzelpackungen, dass Sie exakt  $n$  Minidonuts einkaufen (nicht mehr, nicht weniger). Geben Sie die zu kaufende Anzahl an Großpackungen und Einzelpackungen auf der Konsole aus. Ist keine Lösung möglich, geben Sie „geht nicht“ aus.

Zerlegen Sie das Problem in Teilprobleme und überlegen Sie, in welcher Reihenfolge Sie diese im Algorithmus behandeln, **bevor** Sie programmieren. Beispielzahlen:

- $small = 7$ ,  $big = 4$ ,  $n = 13$  ergibt als Lösung 2 Großpackung und 3 Einzelpackungen.
- $small = 8$ ,  $big = 1$ ,  $n = 12$  ergibt 1 Großpackung und 7 Einzelpackungen (*in diesem Falle sind nicht genug Großpackungen da, daher sind mehr Einzelpacks als 0-4 nötig*).
- $small = 7$ ,  $big = 1$ ,  $n = 13$  ergibt „geht nicht“ (*in diesem Fall gibt es zu wenige Packungen im Supermarkt*).
- $small = 2$ ;  $big = 100$ ;  $n = 13$  ergibt „geht nicht“ (*in diesem Fall kann die Kaufmenge nicht exakt erreicht werden, da Einzelpackungen fehlen obwohl insgesamt mehr als genug Minidonuts da sind*)

**Für Tipps** markieren Sie die folgenden weißen Zeilen des PDFs, kopieren diese und fügen es irgendwo ein zum Lesen:

(Ende Tipps). Nächste Seite Knobel-Aufgabe.

**Knobel-Aufgabe (keine Punkte, aber Denkanstoß): WeekDay**

Die Datei `WeekDay.java` (in Moodle) enthält ein Programm zur Berechnung des Wochentags aus dem Datum und verwendet ausschließlich Programmkonstrukte des bisherigen Unterrichts. Lesen Sie den Quellcode langsam durch, so dass Sie nachvollziehen können, was Zeile für Zeile bei Ausführung passiert. (Das dauert eine Weile). Wechseln Sie sich ggf. mit einer/einem Lernpartner\*in alle 20 Zeilen ab, den Code „laut denkend“ durchzugehen. Stellen Sie sich gegenseitig Rückfragen zu einzelnen Teilen, wie z.B. den `if`-Anweisungen oder warum eine Teilberechnung gebraucht wird. Bei allen unklaren Programmteilen fragen Sie Ihre Tutor\*innen! (und gibt es wirklich nichts, was Ihnen unklar ist, dann erklären Sie doch mal einer Tutor\*in, wie das mit den Schaltjahren funktioniert, so dass trotzdem der richtige Wochentag rauskommt).