

Übungsblatt 5

Aufgabe 1 (if-else, 3 Punkte)

Gegeben: Ein Film mit einer Anzahl Toten, Anzahl Schwerstverletzte und einer Laufzeit in Minuten.

Gesucht: Ausgabe ob der Film angesehen werden soll oder nicht. Im letzten Fall mit Gründen. Es gibt zwei Gründe einen Film nicht anzusehen:

1. Der Film ist zu brutal: es gibt mehr als 7 Tote oder mehr als 127 Schwerstverletzte.
2. Der Film ist zu lang: länger als 118,4 Minuten.

Implementieren Sie drei Funktionen, die jeweils berechnen, ob man den Film anschauen soll, ein Film zu lang oder zu brutal ist. Der Rückgabedatentyp dieser Funktionen ist boolean.

Implementieren Sie eine Klassenmethode, die auf dem Bildschirm ausgibt, ob man den Film ansehen sollte oder nicht und im letzten Fall einen oder beide der Gründe ausgibt.

Drei Ausgabebeispiele:

Film ansehen.

Film nicht ansehen.

Zu brutal.

Zu lang.

Film nicht ansehen.

Zu lang.

Alle zutreffenden Gründe müssen in obiger Reihenfolge ausgegeben werden. Die Werte zur Berechnung müssen als Parameter übergeben werden.

Aufgabe 2 (switch, 3 Punkte)

Erweitern Sie die Klasse `Datum` aus der vorherigen Übungsaufgabe um eine Funktion, die den Wochentag eines Datums berechnet.

Ein Datum sei mit drei int-Werten für Tag (1-31), Monat (1-12) und Jahr kodiert.

Implementieren Sie eine Klassenmethode

```
public static int getWochentag(int tag, int monat, int jahr)
```

die den Wochentag des Datums bestimmt und als Zahl von 0 (= Montag) bis 6 (=Sonntag) zurück gibt. Dabei sollen nur Jahre ab 1900 berücksichtigt werden.

Um den Wochentag zu bestimmten gehen Sie wie folgt vor: Zuerst muss die Gesamtanzahl der Wochentage von 1900 bis zum gegebenen Datum errechnet werden (s.u.). Dies ist letztlich die Hauptschwierigkeit. Wenn diese Zahl berechnet ist, muss sie nur noch mit Rest durch 7 geteilt werden. Das ergibt den Wochentag, wobei 0 der Sonntag ist.

Die Gesamtanzahl Wochentag schrittweise berechnen:

1. Vom gegebenen Jahr 1900 abziehen und dies mit 365 multiplizieren.
2. Die fehlenden Schalttage hinzu addieren ($\text{jahr} - 1900$) / 4. Wenn das Jahr selbst ein Schaltjahr ist, dann ist für Januar und Februar ein Tag zu viel berechnet. In diesem Fall Eins abziehen.
3. Pro Monat die entsprechende Anzahl Tage pro Monat hinzu addieren (beim Februar immer 28 Tage, da das Schaltjahr schon berücksichtigt ist). Verwenden Sie hier eine switch-Anweisung zu Fallunterscheidung. Verwenden Sie das switch so, dass ähnliche Berechnungen zusammengefasst werden.
4. Noch den Tag dazu addieren.

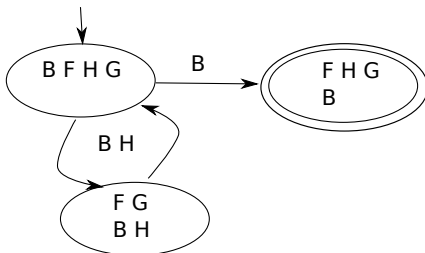
Aufgabe 3 (Fallunterscheidungen, 3 Punkte)

Betrachten Sie folgende Denksportaufgabe:

Ein Bauer (B), ein Huhn (H), ein Fuchs (F) und Getreide (G) befinden sich am Ufer eines Flusses. Der Bauer möchte mit einem Boot alles auf das andere Ufer bringen. Leider kann der Bauer nur sich und höchstens eins der drei Dinge mit ins Boot nehmen. Wenn allerdings der Fuchs mit dem Huhn ohne den Bauern zusammen an einem Ufer bleibt, frisst der Fuchs das Huhn. Wenn das Huhn ohne den Bauern mit dem Getreide an einem Ufer bleibt, dann frisst das Huhn das Getreide. In welcher Reihenfolge muss der Bauer mit einem der drei Dinge von einem zu anderen Ufer fahren, damit sich alle zu Schluss am anderen Ufer befinden?

Dieses Problem lässt sich systematisch mit einem endlichen Automaten lösen. Ein einzelner Zustand des Automaten beschreibt, welche Dinge sich am welchen Ufer befindet. Eine Transition beschreibt das Übersetzen des Bauern mit einen der drei Dingen.

Folgender Automat zeigt den Anfangszustand und zwei mögliche Zustandsübergänge. Die Dinge am linken Ufer sind im Zustandsknoten oberhalb der Dinge am rechten Ufer angegeben. Wenn der Bauer alleine übersetzt, dann frisst der Fuchs das Huhn. Aus diesem Endzustand kommt der Bauer nicht mehr heraus. Beim anderen Zustandsübergang fährt der Bauer mit dem Huhn über das Ufer. Wenn er mit dem Huhn wieder zurückfährt, dann befindet sich der Automat im Anfangszustand.



Erweitern Sie den Automaten um alle möglichen Zustandsübergänge und Zustände.

Lesen Sie daraus die kürzeste mögliche Lösung ab.

Auch wenn das Problem jetzt gelöst ist, soll dieser Automat noch implementiert werden, um eine komplette Simulation zu erhalten.

Die Zustände werden am besten mit ganzen Zahlen codiert: Nummerieren Sie alle Zustände der Reihe nach durch. Da bei einer Transition immer der Bauer mit dabei ist, reicht es aus, die drei Dinge und Nichts mit ganzen Zahlen zu codieren.

Implementieren Sie eine Klasse, mit einer Funktion

```
public static int bootFahren(int aktuellerZustand, int ding),
```

die die Zustandsübergangstabelle des Automaten implementiert. Die Funktion soll den Folgezustand als Funktionswert zurückgeben. Für nicht definierte Fälle, kann die Funktion einfach -1 zurückgeben. Sie werden viele Fallunterscheidungen benötigen.

Sie können die Klasse noch um eine weitere Funktion erweitern, die einen Zustand als Parameter übergeben bekommt, um dann einen zugehörigen Text auf dem Bildschirm auszugeben, wie etwa "Linkes Ufer: Bauer und Huhn. Rechtes Ufer: Fuchs und Getreide" oder "Fuchs frisst Huhn".

Rufen Sie die Methoden alle in der richtigen Reihenfolge auf, so dass der Bauer alles vom linken zum rechten Ufer schafft.