Prof. Dr. Harald Brandenburg
Hochschule für Technik und Wirtschaft (HTW)
Fachbereich 4 (Wirtschaftswissenschaften II)
Wilhelminenhofstraße 75 A
12459 Berlin (Oberschöneweide)
Raum WH C 605

Fon: (030) 50 19 - 23 17
Fax: (030) 50 19 - 26 71
h.brandenburg@htw-berlin.de

Dienstag, 20. Mai 2014

Programmierung 2

SS 2014

Aufgabe 6: **Gruppe 1:** 27.05.2014 **Gruppe 2:** 03.06.2014

Schreiben und dokumentieren Sie ein objektorientiertes Java-Programm, das die Programmplanung eines Fernsehsenders bei der Übertragung eines Fußballspiels wie folgt simuliert.

- Das Fußballspiel beginnt planmäßig um 20:45 Uhr. Es kann aber auch zu einer (kurzen) Verzögerung kommen.

- Die beiden Halbzeiten des Spiels dauern mindestens 45 Minuten. Es kann aber auch Nachspielzeiten geben.

- Die Pause zwischen den Halbzeiten dauert in der Regel 10 bis 15 Minuten.

- Traditionell beginnen bei diesem Sender die Nachrichten um 23:00 Uhr. Die Zeit zwischen dem Ende des Spiels und dem Beginn der Nachrichten wird für Interviews (und die Moderation) genutzt.

- In diesem Normalfall soll Ihr Programm z.B. folgende Ausgabe erzeugen, wobei der Beginn des Spiels und die Dauer der Halbzeiten und der Pause mit Hilfe eines Zufallszahlengenerators festgelegt werden und die Zeit für die Interviews in Abhängigkeit davon ermittelt wird:

```
Spielbeginn                    :  20:46
Ende 1. Halbzeit               :  21:36
Beginn 2. Halbzeit             :  21:50
Spielende                      :  22:35
Spieldauer                     :  01:49
Zeit fuer Interviews           :  00:25
Beginn der Nachrichten         :  23:00
```

- Es kann aber auch zu einer Verlängerung des Fußballspiels kommen. Zu beachten ist dann noch die Dauer folgender Zeiträume:

  - Pause bis zum Beginn der Verlängerung (z.B. 3 bis 6 Minuten)
  - 1. und 2. Halbzeit der Verlängerung (jeweils 15 Minuten plus Nachspielzeit)
  - Pause der Verlängerung (z.B. 1 bis 4 Minuten)

- In diesem Fall müssen die Nachrichten verschoben werden. Dabei verfolgt der Sender die Strategie, dass für Interviews mindestens 10 Minuten zur Verfügung stehen müssen und die Nachrichten immer nur zu einer Uhrzeit beginnen, deren Minutenanteil glatt durch 10 teilbar ist. Eine typische Ausgabe in diesem Fall sieht daher so aus:

```
Spielbeginn                    :  20:45
```

```
Ende 1. Halbzeit                  :  21:35
Beginn 2. Halbzeit                :  21:46
Ende 2. Halbzeit                  :  22:33
Beginn Verlaengerung              :  22:39
Ende 1. Halbzeit Verlaengerung    :  22:57
Beginn 2. Halbzeit Verlaengerung  :  23:01
Spielende                         :  23:16
Spieldauer                        :  02:31
Zeit fuer Interviews              :  00:14
Beginn der Nachrichten            :  23:30
Verspaetung folgender Sendungen   :  00:30
```

- Nun ist es auch noch möglich, dass ein Elfmeterschießen erforderlich wird. Dann sind weitere Zeiträume zu berücksichtigen:

  - Pause bis zum Beginn des Elfmeterschießens (z.B. 5 bis 10 Minuten)
  - Dauer des Elfmeterschießens (z.B. 15 bis 30 Minuten)

- Auch in diesem Fall wird der Beginn der Nachrichten nach derselben Strategie verschoben, so dass Ihr Programm z.B. folgendes ausgibt:

```
Spielbeginn                       :  20:48
Ende 1. Halbzeit                  :  21:37
Beginn 2. Halbzeit                :  21:47
Ende 2. Halbzeit                  :  22:34
Beginn Verlaengerung              :  22:37
Ende 1. Halbzeit Verlaengerung    :  22:55
Beginn 2. Halbzeit Verlaengerung  :  22:56
Ende 2. Halbzeit Verlaengerung    :  23:14
Beginn Elfmeterschiessen          :  23:22
Spielende                         :  23:37
Spieldauer                        :  02:49
Zeit fuer Interviews              :  00:13
Beginn der Nachrichten            :  23:50
Verspaetung folgender Sendungen   :  00:50
```

[**Hinweise**:

- Im Zentrum Ihrer Lösung soll eine Klasse stehen (hier **`MeineZeit`** genannt), die die zur Manipulation von Zeitangaben benötigten Operationen zur Verfügung stellt.

- **`MeineZeit`** könnte z.B. folgende Gestalt haben:

  - Nur zwei private ganzzahlige Attribute für die Stunde und die Minute einer Zeitangabe der Form **`23:37`**. Zulässige Werte für die Stunde sind **`0`** bis **`23`**, für die Minute **`0`** bis **`59`**.

  - Mehrere Konstruktoren, z.B. **`MeineZeit(int dieStunde, int dieMinute)`** und **`MeineZeit(int anzahlMinuten)`**.

    Dabei ist zu berücksichtigen, dass der Input geeignet normalisiert werden muss. Während **`new MeineZeit(23, 37)`** ein Objekt erzeugt, das die Zeitangabe **`23:37`** repräsentiert, soll auch **`new MeineZeit(25, 71)`** zulässig sein und ein Objekt erzeugen, das die Zeitangabe **`02:11`** repräsentiert. **`new MeineZeit(319)`** soll ein Objekt erzeugen, das die Zeitangabe **`05:19`** repräsentiert, und das von **`new MeineZeit(1441)`** erzeugte Objekt **`00:01`** repräsentieren. Lediglich negative Werte als Input sind nicht zulässig und sollen zum Abbruch des Programms führen.[1]

  - Etliche Methoden, darunter z.B.

    `public MeineZeit addiereMinuten(int anzahlMinuten)`

1 Mit Hilfe der **`throw`**-Anweisung und einer **`IllegalArgumentException`**.

`public MeineZeit bildeDifferenz(MeineZeit zeitAngabe)`

Die Methode **`addiereMinuten`** soll für jeden positiven Input funktionieren. So liefert **`x.addiereMinuten(9)`** für das Objekt **`x`**, das **`23:47`** repräsentiert, z.B. das Objekt, das die Zeitangabe **`23:56`** repräsentiert, und **`x.addiereMinuten(134)`** das Objekt, das **`02:01`** repräsentiert. Negative Werte als Input sind nicht zulässig und sollen zum Abbruch des Programms führen.

Bei der Methode **`bildeDifferenz`** ist darauf zu achten, dass sie auch dann einen sinnvollen Wert liefert, wenn die Input-Zeitangabe vor der Zeitangabe des aktuellen Objekts liegt. So soll **`x.bildeDifferenz(new MeineZeit(23,55))`** für das Objekt **`x`**, das **`23:47`** repräsentiert, das Objekt liefern, das die Zeitangabe **`00:08`** repräsentiert, dagegen **`x.bildeDifferenz(new MeineZeit(0,9))`** das Objekt, das **`00:22`** repräsentiert.

]