

2. Übungsblatt : Grundlagen der Programmierung [20 Punkte]

Aufgabe 1 (array, Schleifen, Simulation) [6 Punkte]:

Geburtstags-Paradoxon

Wie groß ist die Wahrscheinlichkeit, dass von x zufällig in einem Raum befindlichen Leuten zwei am gleichen Tag Geburtstag haben? (Egal an welchem **Tag**.)

Ermitteln Sie mit Hilfe einer Monte-Carlo-Simulation in Java (analog zum aus der VL bekannten Problem Gambler's Ruin) die durchschnittlich nötige Anzahl von Personen in dem Raum, sodass mindestens zwei Personen am gleichen Tag Geburtstag (einer von 365 möglichen) haben. Achten Sie darauf, statistisch zuverlässige, valide Ergebnisse zu erhalten. (Tipp: Sie sollten nicht nur ein Experiment simulieren, sondern mehrere!)

Abgabedatei im Goya: `Birthday.java`

Aufruf:

```
$ java Birthday
34.5678
```

Aufgabe 2 [4 Punkte]:

Freitag der 13.

Was ist die maximale Anzahl von aufeinanderfolgenden Tagen, in denen kein „Freitag der 13.“ vorkommt? Hinweis: Der Gregorianische Kalender wiederholt sich alle 400 Jahre (146097 Tage), so brauchen Sie sich nur um ein „400-Jahre“-Intervall zu kümmern. Beginnen Sie ihre Untersuchung mit dem 1.11.2010. (`java.util.Date`, `java.util.Calendar` dürfen nicht verwendet werden!)

Abgabedatei im Goya: `No13.java`

Aufruf & Ausgabe:

```
$ java No13
247
```

Aufgabe 3 [4 Punkte]:

Sa oder So

Fällt der 1. Januar häufiger auf einen Sonnabend oder einen Sonntag? Schreiben Sie ein Programm `Wochenende.java`, um die jeweiligen Häufigkeiten (Sa, So) in dem 400-Jahre-Intervall beginnend am 1.1.2011 zu bestimmen.

Abgabedatei im Goya: `Wochenende.java`

Aufruf & Ausgabe:

```
$ java Wochenende
Sa: 123
So: 456
```

Aufgabe 4 [6 Punkte]:

„Let’s make a Deal“

In den 70er Jahren gab es eine Spielshow „Let’s make a Deal“, wo dem Kandidaten drei Türen angeboten wurden. Hinter einer der Türen ist ein wertvoller Preis, hinter den anderen beiden Türen befindet sich nichts. Nachdem der Kandidat eine Tür gewählt hat, öffnet der Moderator eine der beiden anderen Türen (wobei er natürlich nie eine Tür mit dem Preis öffnen würde). Dem Kandidaten wird dann die Möglichkeit gegeben, zu einer anderen geschlossenen Tür zu wechseln. Sollte der Kandidat dies tun? Intuitiv könnte der Kandidat annehmen, dass seine ursprünglich ausgewählte Tür und die andere ungeöffnete Tür mit gleicher Wahrscheinlichkeit den Preis enthalten und es somit keinen Grund zum Wechseln gibt. Schreiben Sie ein Programm `MonteHall.java` um diese Erwartung mittels Simulation zu überprüfen. Ihr Programm sollte abhängig Kommandozeilenparameter N dieses Spiel N-mal für jede Strategie (Wechseln / Nichtwechseln) ausführen und die mittlere Erfolgchance für jede Strategie ($\frac{\text{\# der gewonnenen Spiel}}{\text{\# der Spiele}} = N$) ausgeben.

Abgabedatei im Goya: `MonteHall.java`

Aufruf & Ausgabe:

```
$ java MonteHall 1000
```

```
Wechseln: 0.123
```

```
Nichtwechseln: 0.456
```

HINWEIS: Die angegebenen Ausgaben zeigen natürlich nicht die korrekten Resultate, sondern sind fiktiv.

Folgende Dateien sind über Goya als **UTF-8 kodierte** Lösungsquelltexte abzugeben:

`Birthday.java`

`No13.java`

`Wochenende.java`

`MonteHall.java`