

Programmieren und Software-Engineering I

Übung 14

Name: _____ Klasse: _____ Datum: _____

Lernziele:

- Benutzereingabe
- Testdokumentation
- switch

Aufgabe 1: Datum II (Ex_14_01_DateII)

Erstelle ein Java-Programm, welches prüft, ob es sich um ein gültiges Datum handelt. Dabei sollen auch Schaltjahre geprüft werden. Ein Schaltjahr liegt prinzipiell dann vor, wenn die Jahreszahl durch 4 teilbar ist.

Es gilt dabei aber folgende Ausnahme: Jahreszahlen, die durch 100 teilbar, aber nicht durch 400 teilbar sind, sind keine Schaltjahre (d.h. 1600 und 2000 waren Schaltjahre. Die Jahre 1700, 1800, 1900 oder 2100 waren bzw. sind keine Schaltjahre.).

Beispiel:

Tag:	15
Monat:	11
Jahr:	1865
Ausgabe:	Der 15.11.1865 ist ein gültiges Datum.
...	
Tag:	29
Monat:	2
Jahr:	2004
Ausgabe:	Der 29.2.2004 ist ein gültiges Datum.
...	
Tag:	29
Monat:	2
Jahr:	1900
Ausgabe:	Der 29.2.1900 ist kein gültiges Datum.
...	
Tag:	31
Monat:	4
Jahr:	2008
Ausgabe:	Der 31.4.2008 ist kein gültiges Datum.

Testdokumentation

Beschreibung wie das Programm getestet wurde.

Testfall	Tag	Monat	Jahr	Ausgabe erwartet	Ausgabe tatsächlich

Aufgabe 2: Lotto (Ex_14_02_Lotto)

Erstelle ein Java-Programm, welches einen gültigen Lottotipp erzeugt (d.h. sechs VERSCHIEDENE Zahlen zwischen 1 und einer Obergrenze).

Die Obergrenze wird dabei vom Benutzer eingegeben und MUSS zwischen 10 und 50 liegen (mit sofortiger Prüfung!!!)

Beispiel:

Obergrenze eingeben (10-50): **88**
Ungültige Obergrenze!!!

Obergrenze eingeben (10-50): **7**
Ungültige Obergrenze!!!

Obergrenze eingeben (10-50): **45**
Zahlen: 34, 12, 39, 1, 5 und 27.

Programmieren und Software-Engineering I

Übung 14

Aufgabe 3: Münze (Ex_14_03_Coin)

Erstelle ein Java-Programm, welches einen mehrmaligen Münzwurf simuliert (Zufallszahl 0 oder 1). Die Würfe (K oder Z) werden ausgegeben. Ebenso der längste Run (d.h. die längste Serie von Kopf oder Zahl hintereinander), sowie die längste Serie, wo Kopf und Zahl abwechselnd geworfen wurden.

Beispiel: KKZKZKZZZZKKZZZZZZ...

Längster Run:	Zahl (8 x)
Längste Serie:	7 x
Gesamtwürfe:	100 x

Aufgabe 4: Schere, Stein, Papier (Ex_14_04_RockPaperScissors)

Erstelle eine Java-Programm das das bekannte Spiel Schere-Stein-Papier implementiert für einen Spieler gegen den Computer simuliert. Der Spieler gibt dafür eine Zahl ein (etwa: 1-2-3), der Computer „tippt“ per Zufallszahl. Danach erfolgt die Auswertung des aktuellen Versuchs nach den bekannten Regeln: Schere schlägt Papier, Papier schlägt Stein und Stein schlägt Schere. Der Sieger eines Versuchs erhält einen Punkt, bei einem Unentschieden bekommt keiner einen Punkt.

Wer als erster eine festzulegende Punkteanzahl erreicht hat, gewinnt das Spiel.

Ablaufbeispiel:

Schere – Stein – Papier

Siegepunkte: 2

Spieler 1:	1
Computer:	3

Punkt für Spieler 1.

Spieler 1:	2
Computer:	2

Unentschieden.

Spieler 1:	1
Computer:	2

Punkt Computer.

Spieler 1:	3
Computer:	1

Punkt für Computer.

Computer gewinnt.

WICHTIG

Alle Programme müssen einen Programmkopf (=Beschreibung) enthalten.

z.B.:

```

/*****
*      Name:      Max Mustermann
*      Hü:        4
*      Bsp:       2
*      Datum:     20.10.2015
*      Dateiname:  HUE_04_02_Dreieck.java
*      Beschreibung: Es wird für gegebene Seitenlängen a, b und c geprüft, ob es ein gleichseitiges, ein
                    gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges
                    Dreieck ist.
*****/
```

```
public class HUE_04_02_Dreieck {
```