

DHBW Karlsruhe, Vorlesung Programmieren, „Events“ (1)

Aufgabe 1 „Euroumrechner“

Erweitern Sie die Aufgabe Euroumrechner des Aufgabenblatts „Swing (JFC) so, dass auf Events an den einzelnen Buttons reagiert wird:

Sobald ein Button gedrückt wurde soll die Benutzereingabe im Textfeld

- durch eine Fehlermeldung ersetzt werden, wenn es sich um keine Zahl handelt;
- mit dem Faktor 1,95583 multipliziert werden, wenn „EUR→DM“ gedrückt wurde;
- durch den Faktor 1,95583 dividiert werden, wenn „DM→EUR“ gedrückt wurde.

Wenn der Button „Abbruch“ gedrückt wurde soll die Anwendung beendet werden.

Hinweise:

Die Klasse Euroumrechner könnte das Interface `java.awt.event.ActionListener` und dessen Methode `actionPerformed(ActionEvent e)` implementieren.

Mit der Methode `addActionListener(this)` können Sie dann jedem Button diesen Eventhandler zuweisen. Innerhalb der Methode `actionPerformed` müssen Sie dann allerdings unterscheiden, welche GUI-Komponente das Ereignis ausgelöst hat.

Für kleine Programme mit wenigen Komponenten und möglichen Events ist diese Vorgehensweise ok.

Für größere Programme ist es meist sinnvoll, Events als anonyme Klassen zu implementieren und direkt an die entsprechende GUI-Komponente (hier ein `JButton`) zu koppeln, z.B.

```
public JButton eur2dm = new JButton("EUR -> DM");
eur2dm.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        /* nun wissen wir, dass der Button eur2dm gedrückt wurde
           und können entsprechend reagieren */
    }
});
```

Das Beenden einer Anwendung kann man mit `System.exit(0)` herbeiführen.

Aufgabe 2 „Body-Mass-Index (BMI)“

Der Body-Mass-Index (BMI) ist ein Index, der das Verhältnis von Körpergewicht zur Größe angibt. Die Formel für den BMI lautet:

$$\text{BMI} = (\text{Körpergewicht in kg}) / (\text{Größe in m})^2.$$

Folgende Tabelle zeigt die Klassifikation der BMI-Werte in Abhängigkeit des Geschlechts.

Klassifikation	Männer	Frauen
Untergewicht	<20	<19
Normalgewicht	20-25	19-24
Übergewicht	25-30	24-30
Adipositas	30-40	30-40
massive Adipositas	>40	>40

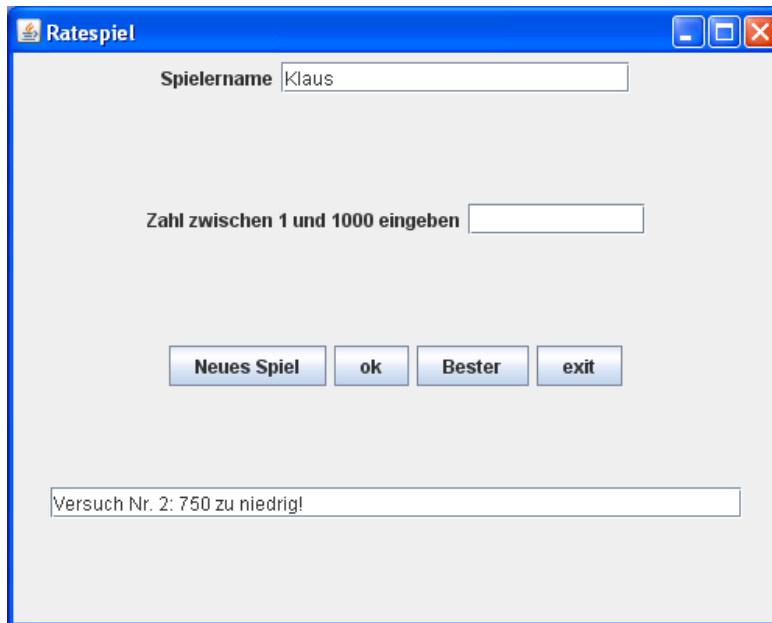
- Schreiben Sie eine JFrame-basierte Java-Applikation, in der man die Körpergröße und das Gewicht eingeben kann und die dann den BMI berechnet.
- Erweitern Sie Ihre Applikation derart, dass der Benutzer eine Einschätzung seines BMI anhand obiger Tabellen erhält.



Aufgabe 3 „Zahlen-Ratespiel“

(Dies war die Klausur vom Sommersemester 2003)

Schreiben Sie eine Java-Applikation, die ein Ratespiel implementiert. Bei diesem Ratespiel muss der Benutzer eine zufällig erzeugte Zahl zwischen 1 und 1000 erraten. Als Hinweis bekommt er angezeigt, ob er zu hoch oder zu niedrig getippt hat. Die Anzahl der Versuche, die er benötigt, wird mitgezählt. Wenn er die Zahl erraten hat, wird sein Name und die Anzahl der Versuche an eine Ergebnisdatei angehängt. Eine mögliche graphische Oberfläche zu diesem Ratespiel sieht wie folgt aus:



Der Benutzer muss folgendes über der Oberfläche eingeben können:

- Spielername
- Nächste getippte Zahl
- Zwei Buttons zum Erzeugen eines neuen Spiels und zum Beenden der Applikation.
- Die Prüfung, ob die aktuelle Eingabe höher, niedriger oder gleich der gesuchten Zahl ist, soll entweder über einen „ok“-Button oder durch ein Return im Textfeld erfolgen. (Eine der beiden Lösungen ist zu realisieren).

Im Ausgabefeld muss stehen:

- ob der Tipp zu hoch, zu niedrig oder richtig war.
- Wie viele Versuche bisher gemacht wurden.

Wurde richtig getippt, muss an eine Ergebnisdatei der Name des Spielers und die Anzahl der Versuche angehängt werden.

Zusatzaufgabe:

Erweitern Sie die Applikation um einen Button „Bester“. Bei Anklicken des Buttons soll das Ergebnisfile gelesen werden. Aus dem Ergebnisfile soll der bisher beste Spieler (d.h. der Spieler mit den wenigsten Versuchen) ermittelt und im Ausgabefeld ausgegeben werden.

Aufgabe 4 „Hütchenspiel“ (Nachklausur vom Sommersemester 2003)

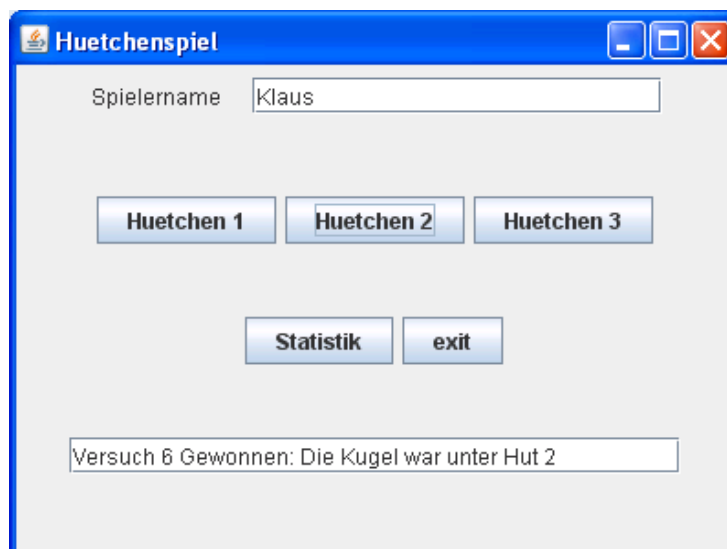
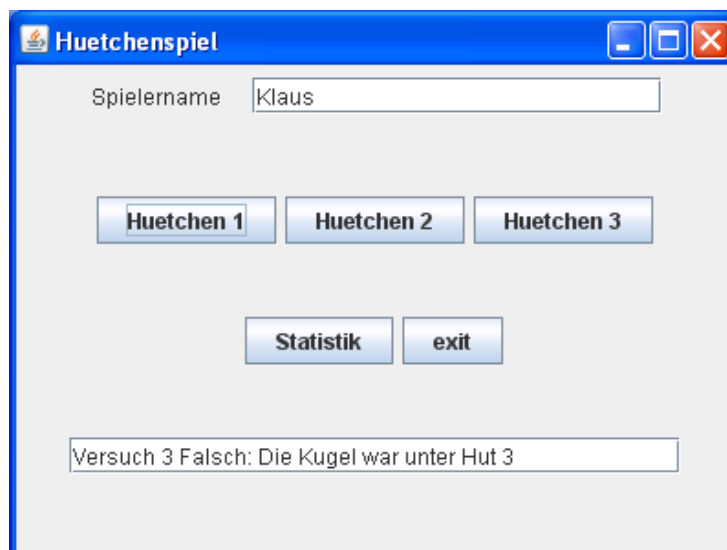
Schreiben Sie eine Java-Applikation, die das bekannte Hütchenspiel implementiert!

Beim Hütchenspiel gibt es drei Hütchen. Unter einem der Hütchen liegt eine Kugel. Der Spieler muss erraten, unter welchem Hütchen die Kugel liegt.

Die drei Hütchen sollen in der Java-Applikation durch drei Buttons realisiert werden. Bei jedem Spielzug wird zufällig die Kugel unter ein Hütchen gelegt.

Wurde ein Hütchen-Button angeklickt, so bekommt der Spieler in einem Textfeld angezeigt, ob er richtig geraten hat und wie viele Versuche er bisher hatte. Hat er richtig geraten wird die Anzahl der Versuche und der Name des Spielers in ein Ergebnisfile angehängt.

Eine mögliche graphische Oberfläche zu diesem Hütchenspiel sieht wie folgt aus:



Der Benutzer muss folgendes über der Oberfläche eingeben können:

- Spielername
- Drei Buttons, die die Hütchen realisieren. Nach jedem Spielzug wird die Kugel zufällig unter einem Hütchen verteilt. (Verwenden Sie dazu z.B. eine zufällige erzeugte Integerzahl zwischen 1 und 3)
- Ein Button zum Beenden der Applikation.

Im Ausgabefeld muss stehen:

- Wie viele Versuche bisher gemacht wurden.
- Ob das richtige Hütchen getippt wurde oder nicht.
- Wurde richtig getippt muss an eine Ergebnisdatei der Name des Spielers und die Anzahl der Versuche angehängt werden.

Zusatzaufgabe:

Erweitern Sie die Applikation um einen Button „Statistik“. Bei Anklicken des Buttons soll das Ergebnisfile gelesen werden. Aus dem Ergebnisfile soll der Durchschnitt der bisherigen Versuche ermittelt und ins Ausgabefeld ausgegeben werden.

Hinweis:

Objekte der Klasse `StringTokenizer` im Paket `java.util` ermöglichen es, Zeichenketten in einzelne Teilzeichenketten, so genannte Tokens, zu zerlegen. Dabei versteht man unter einem Token jeweils eine zusammenhängende Folge von Zeichen, die von einem Trennzeichen eingegrenzt sind. Mit Hilfe eines `StringTokenizer`-Objekts können die aus dem Ergebnisfile eingelesenen Zeilen einfach in einzelne Token zerlegt werden.