

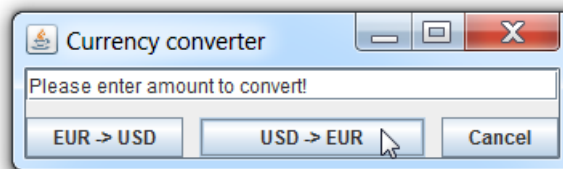
**Bereich: Grafische Benutzeroberflächen (UI), Events (1)****Währungsumrechner (2)****Package:** `de.dhbwka.java.exercise.ui.event`**Klasse:** `CurrencyCalculator`**Aufgabenstellung:**

Erweitern Sie die Aufgabe `CurrencyCalculator` des Aufgabenblatts „Grafische Benutzeroberflächen (2)“ 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 1,09 multipliziert werden, wenn „EUR → USD“ gedrückt wurde;
- durch den Faktor 1,09 dividiert werden, wenn „USD → EUR“ gedrückt wurde.

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

**Hinweis:**

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

## Bereich: Grafische Benutzeroberflächen (UI), Events (1)

### Body-Mass-Index (BMI)

**Package:** de.dhbwka.java.exercise.ui.event

**Klasse:** BMICalculator

#### Aufgabenstellung:

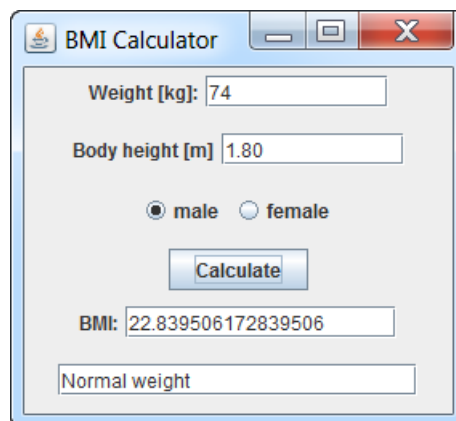
Der Body-Mass-Index (BMI) ist ein Index, der das Verhältnis von Körpergewicht zur Größe angibt. Die Formel für den BMI lautet:

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

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

Klassifikation	Männer	Frauen
Untergewicht (Short weight)	<20	<19
Normalgewicht (Normal weight)	20-25	19-24
Übergewicht (Overweight)	25-30	24-30
Adipositas (Adiposity)	30-40	30-40
massive Adipositas (Massive Adiposity)	>40	>40

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



#### Hinweis:

In der Klasse `JRadioButton` gibt es eine (geerbte) Instanzmethode `boolean isSelected()`, die den Zustand eines Feststell-Buttons zurückliefert: `true` wenn der Button ausgewählt ist, `false` wenn nicht.

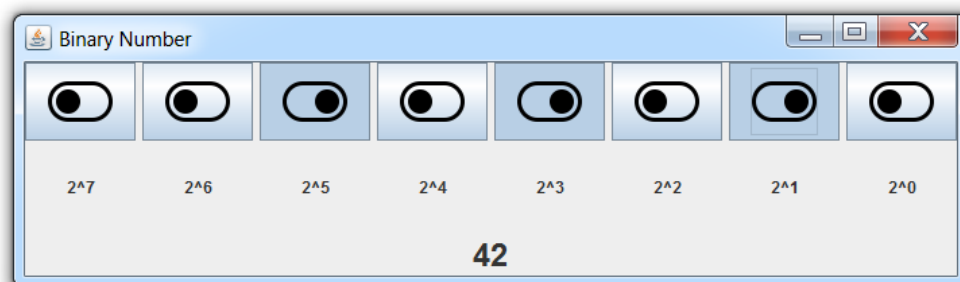
**Bereich: Grafische Benutzeroberflächen (UI), Events (1)****Binäre Zahlen****Package:** de.dhbwka.java.exercise.ui.event**Klasse:** BinaryNumber**Aufgabenstellung:**

Schreiben Sie zur Bestimmung des Dezimalwerts einer Binärzahl eine Benutzeroberfläche, welche das Setzen von 8 Bits ermöglicht.

Nutzen Sie hierbei zum Setzen eines virtuellen Bits `JToggleButton`s. Ist der Button gedrückt, soll das Bit für den Wert als gesetzt gelten.

Zur besseren optischen Veranschaulichung nutzen Sie die bereitgestellten Icons, um den Zustand des jeweiligen `JToggleButton`s herauszuheben (siehe Hinweise). Beschriften Sie auch jeweils unterhalb der `JToggleButton`s, welche 2-er Potenz das Bit repräsentiert.

Im unteren Bereich soll zusätzlich ein Label den aktuellen Dezimalwert anzeigen, der sich durch Einstellen der virtuellen Bits ergibt. Selbstverständlich soll dieser Wert beim Drücken eines Buttons aktualisiert werden.

**Hinweise:**

Kopieren Sie die Dateien `on.png` und `off.png` in Ihr Projektverzeichnis. (Eine Download-Möglichkeit für diese Dateien finden Sie auf der Homepage der Vorlesung bei dieser Aufgabenstellung.) Die Icons für die verschiedenen Zustände der `JToggleButton`s können Sie wie folgt setzen (ImageIcon: Klasse im Package `javax.swing`)

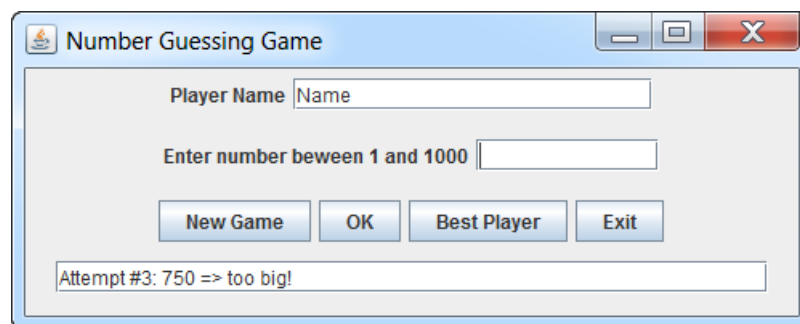
```
ImageIcon imgOff = new ImageIcon("off.png");
ImageIcon imgOn = new ImageIcon("on.png");
toggleButton.setIcon(imgOff); // icon if not selected (default icon)
toggleButton.setSelectedIcon(imgOn); // icon if selected
```

Analog zu `JRadioButton` erbt auch `JToggleButton` die Methode `boolean isSelected()` zur Bestimmung des Zustands (siehe Hinweise in BMI-Aufgabe).

Icon-Quelle: <https://www.iconfinder.com/iconsets/essential-app-1>

**Bereich: Grafische Benutzeroberflächen (UI), Events (1)****Zahlenraten (2)****Package:** de.dhbwka.java.exercise.ui.event**Klasse:** NumberGuess**Aufgabenstellung:**

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, werden sein Name und die Anzahl der Versuche an eine Ergebnisdatei angehängt. Eine mögliche grafische 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.

*Hinweis: Siehe auch die Aufgabe „Zahlenraten“ des Aufgabenblatts „Kontrollstrukturen“, bei der die Ein-/Ausgabe über das Konsolenfenster erfolgt.*

**Zusatzaufgabe:**

Erweitern Sie die Applikation um einen Button „Best Player“. 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.