

## Die Aufgaben 1 bis 6 sollen sich erst mal auf das AWT stützen. In der Regel wird das null-Layout (absolut Layout) verwendet.

Bitte nicht vergessen:

Window → Preferences → WindowBuilder → Swing → Code Generation → Lazy  
oder/bzw.

Window → Preferences → WindowBuilder → Swing → Code Generation → Init. Field

### 1. Aufgabe (AWT)

Üben Sie an diesem Beispiel, das Erstellen mit und ohne WindowBuilder, sowie das Arbeiten mit den jeweils vorgeschlagenen Modis im WindowBuilder

- Erstellen Sie ein *panel* mit zwei Textfeldern auf der Oberfläche. Das erste Textfeld soll eine Größe von 100 x 25 *Pixels* besitzen, eine Eingabe ist erlaubt. Das zweite Textfeld soll eine Größe von 80 x 25 *Pixels* besitzen, eine Eingabe ist hier *nicht* erlaubt. Das zweite Textfeld soll den Text »nur lesbar« enthalten. Achten Sie auf korrekte Syntax und Strukturierung.
- Integrieren Sie das *panel* in einen startbaren *Frame*.
- Beim Initialisieren soll der Inhalt des zweiten Textfeldes in das erste Textfeld geschrieben werden → dies soll über den Panel-Konstruktor geschehen.
- Erstellen Sie eine einzige startbare Klasse mit Panel und Frame. Konzipieren Sie die Klasse so, dass Sie sie testen können.

### 2. Aufgabe (SWING)

- Wiederholen Sie Aufgabe 1 jetzt mit entsprechenden SWING-Komponenten
- Gestalten Sie in einer weiteren Variante die Oberfläche direkt über einen JFrame.
- Verwenden und testen Sie die *DefaultCloseOperation*.

### 3. Aufgabe

Wiederholen Sie Experimente aus der Vorlesung. Erstellen Sie ein Panel. Setzen Sie den Layout-Manager der Oberfläche auf **null**. Platzieren Sie einen Button auf die Oberfläche. Beim Drücken soll der Text „Hallo Nr. X“ als Label auf der Oberfläche erscheinen. X steht für 1, 2, ... usw. Ändern Sie in einem zweiten Schritt den Button so, dass die Beschriftung am Anfang 0 und dann 1, 2, ... usw. ist. Verwenden Sie als Aktionsabhörer einen Standardadapter (externe Klasse).

Fügen Sie dem Button einen Mouse-Listener hinzu. Über den sollen die X-/Y-Koordinaten angezeigt werden (gleiches Label wie „Hallo Nr. X“), wenn Sie die Maus über die Button-Fläche bewegen. Verwenden Sie als Mouse-Listener eine innere Klasse.

Setzen Sie das Panel in einen startbaren Frame und überprüfen Sie die Funktionalität.

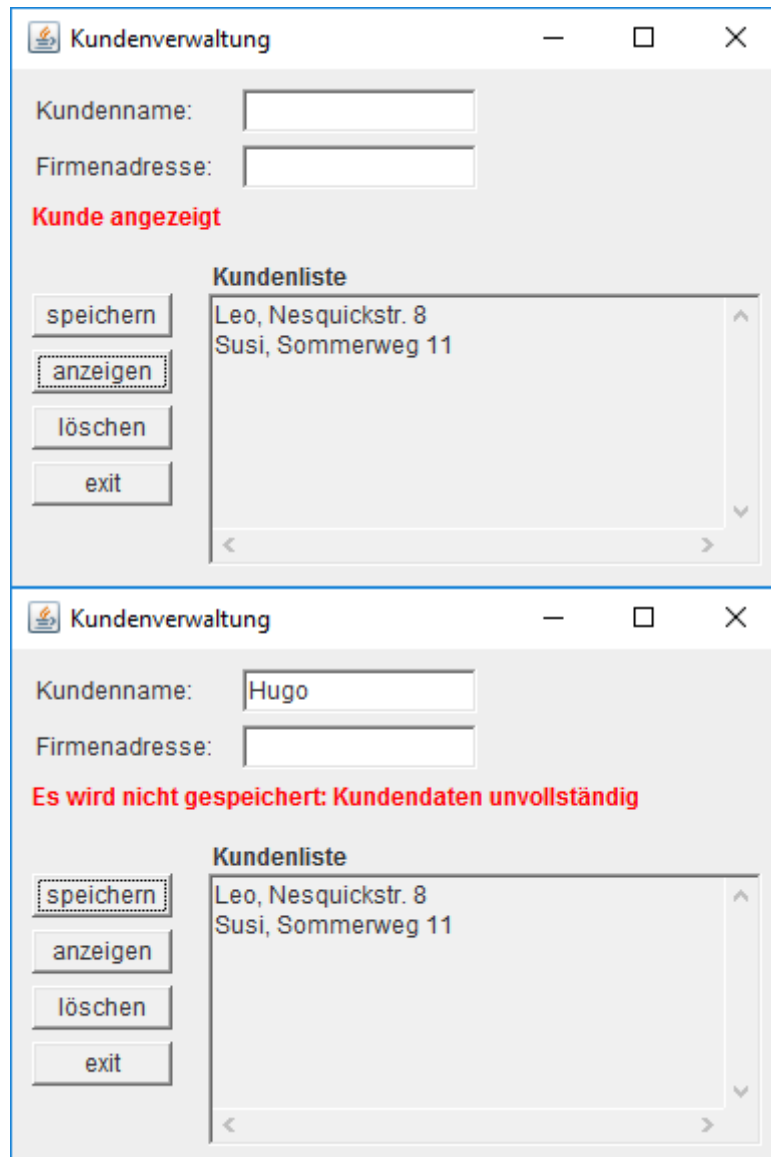
## 4. Aufgabe

Erstellen Sie eine GUI-Applikation Kundenverwaltung mit zwei Textfeldern (Kundenname, Firmenadresse), einer TextArea, einem Speicher-Knopf, einem Anzeige-Knopf, einem Lösch-Knopf und einem Exit-Knopf.

- Beim Drücken des Speicher-Knopfes wird mit dem Inhalt der Textfelder ein Objekt der Klasse Kunde erstellt, die Textfelder werden zurückgesetzt. Das geht gilt nur, wenn alle Textfelder mehr als 2 Zeichen enthalten. Gilt das nicht erfolgt eine entsprechende Statusmeldung.
- Beim Drücken des Anzeige-Knopfes wird das aktuelle Kundenobjekt in der TextArea dargestellt.
- Beim Drücken des Lösch-Knopfes wird der aktuelle Inhalt der textArea gelöscht. Die TextArea darf nicht editierbar sein.
- Beim Drücken des Exit-Knopf wird das Programm beendet.

Grundsätzlich erfolgt zu jeder Operation (Ausnahme EXIT) eine Statusmeldung – in Rot.

Verwenden Sie für jeden dieser vier Knöpfe einen eigenen ActionListener mit jeweils anderer Adapterart: äußerer, innerer, GUI/Container-Klasse selbst, anonymer innerer Adapter. Zu allen Steuerelementen soll es geeignete Label/Führungslabel geben, die ihren Verwendungszweck beschreiben.



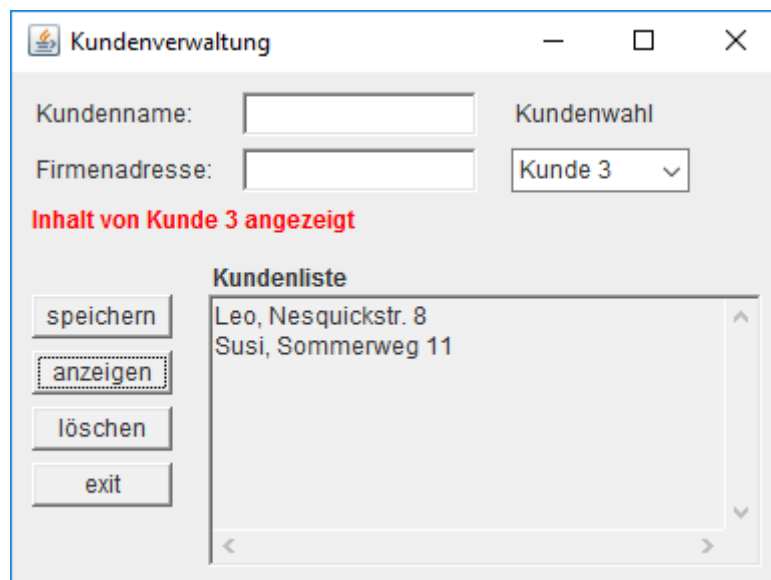
## 5. Aufgabe

Modifizieren Sie Aufgabe 4 so, dass nur noch ein ActionListener-Objekt (nehmen Sie die GUI selbst) für die Buttons existiert. Ergänzen die die Anwendung um einen Window-Listener, über dem das Programm ebenfalls beendet werden kann.

## 6. Aufgabe

Erweitern/Modifizieren Sie das Programm aus der Aufgabe 4 folgendermaßen:

- Die GUI-Klasse soll nun drei Kunden verwalten können.
- Zum Speichern der Kundendaten gibt es nun ein Kombinationsfeld. Mit diesem Kombinationsfeld können die Kundendaten als Objekt mit der Nummer 1, 2 oder 3 gewählt werden. Es existieren auch drei Kundenreferenzen (am besten ein Feld).
- Die Druckknöpfe »Speichern« und »Ausgeben« beziehen sich nun auf den jeweils gewählten Kunden



## 7. Aufgabe (SWING)

Modifizieren Sie die Aufgabe 5 nun so, dass alle Buttons (bei addActionListener) die actionPerformed-Methode der GUI-Klasse via Methoden-Referenz ansteuern.

## 8. Aufgabe

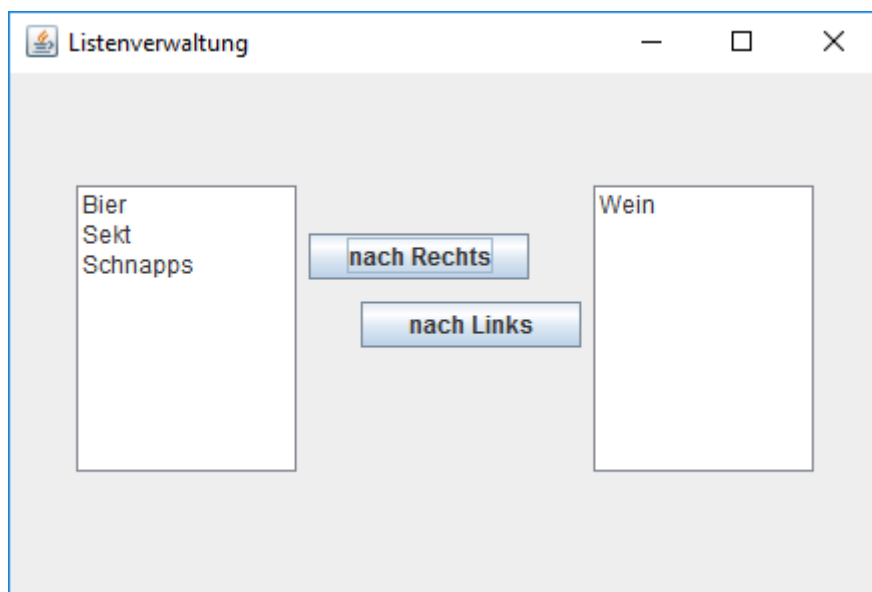
### Lernziel

*Java-Ereignisse: Training der Event-Feinmotorik*

### Aufgabenstellung

Erstellen Sie ein GUI-Applikation mit zwei Listen (AWT) und zwei Buttons. Setzen Sie per Programm mindestens drei Listeneinträge in die erste Liste. Selektieren Sie Listenelemente und verschieben Sie dieses via Button 1 in die zweite Liste. Das Ganze soll auch umgekehrt möglich sein (via Button 2).

- Finden und verwenden Sie die richtigen Listenmethoden. Bitte AWT-Listen verwenden.
- Testen Sie die Stabilität Ihres Programms (was passiert z.B. wenn kein Element selektiert ist und sie drücken einen Button)
- Gestalten Sie Ihr Programm so um, dass nur dann ein nach XXX-Button benutzt werden kann, wenn ein entsprechendes Listenelement selektiert ist.
- Beim Schließen des Fensters soll das Programm beendet werden.
- Verwenden Sie zum Schluss wo möglich Lambda-Ausdrücke



## 9. Aufgabe

Erstellen Sie eine eigene GUI-Klasse `SicherBeenden`, die folgenden Mechanismus bereitstellt:

Beim Schließen des Fensters soll das Programm komplett beendet werden. Allerdings soll dies vorsichtshalber noch einmal bestätigt werden (via popup-Fenster). Testen Sie diesen Mechanismus.

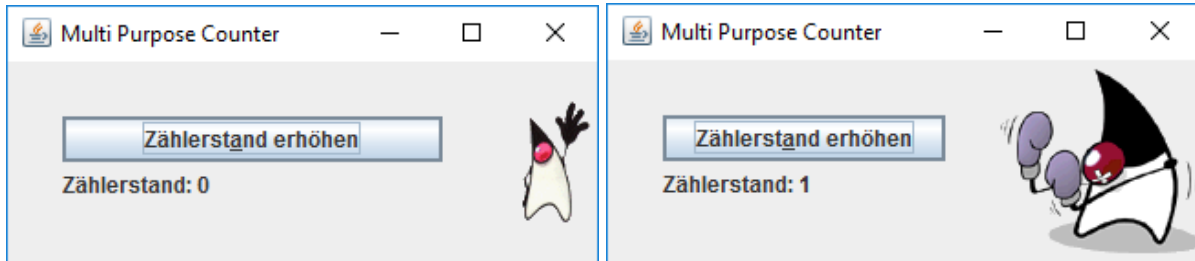
Erstellen Sie dann das Programm Kundenverwaltung (Aufgabe 6) neu: komplett auf SWING-Basis. Diese Klasse soll von der GUI-Klasse `SicherBeenden` erben. Führen Sie dann noch die folgende Modifikation durch:

Durch Drücken der Return-Taste soll es möglich sein, von Eingabefeld zu Eingabefeld zu springen  
Testen Sie Ihr Programm

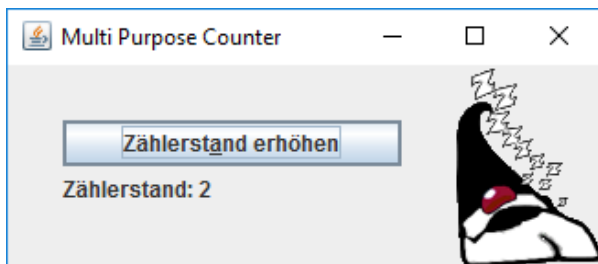
## 10. Aufgabe

Für Freaks oder solche, die es werden wollen.

Erstellen Sie ein Swing-Programm, das als allgemeiner Zähler taugt, mit zwei Label (mit einem Text bzw. Bild als Inhalt) sowie einen Befehlsschalter. Das folgende Bild zeigt den Zähler beim Start und nach dem ersten Schritt.

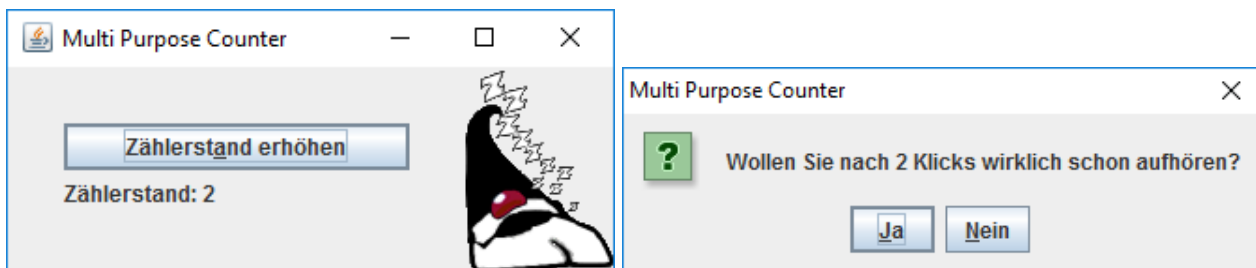


Es stehen insgesamt 3 Icons zur Verfügung – siehe nächstes Bild.



Die Funktionalität ist folgende:

- Wenn der Button gedrückt/ausgelöst wird, erhöht sich der Zählerstand und das Icon wird gewechselt (nach dem dritten kommt wieder das erste)
- Wird die Return-Taste gedrückt wird der Button als Default-Button ausgelöst.
- Wird „ALT a“ gedrückt wird der Button ebenfalls ausgelöst.
- Wird das Programm beendet kommt es zu einer Sicherheitsabfrage – siehe nächstes Bild.
- Die einzelnen Komponenten sind mit „Tooltips“ versehen.



Die drei Icons (duke.gif, fight.gif, snooze.gif) stehen im Lehre-Ordner zur Verfügung. Zur Anschauung gibt es ein ausführbares jar-File der Anwendung. Möglicherweise muss man etwas googeln oder nachlesen – und viel probieren.