

Übung 2 – Medientechnik

Aufgabe:

Erstellen Sie mit der Entwicklungsumgebung Netbeans und den darin integrierten visuellen Werkzeugen eine GUI analog zum Counter-Beispiel aus der Vorlesung.

- Starten Sie in einem Terminal-Fenster: netbeans-3.6
- Gehen Sie auf File->Mount Filesystem... und wählen Sie Local Directory. Wählen Sie nun ein Verzeichnis in ihrem Home, in welchem Sie ihre Netbeans-Project-Dateien ablegen möchten. Dies sollte nun links unter Filesystem zu sehen sein.
- Wählen Sie das Icon für New... bzw. File->New... und wählen Sie als Template unter Java GUI Forms (ausklappen) JFrame Form. Vergeben Sie einen geeigneten Namen, wie z.B. CounterGUI
- Nun sollte bereits ein leeres JFrame zu sehen sein. Rechts ist die Palette unter der Swing und AWT Elemente zu finden sind (iconifiziert). Darunter befindet sich der Properties-Editor. Wählen Sie das gesamte JFrame aus (anklicken) und betrachten Sie die verschiedenen Properties.
- Wenn Sie auf den Reiter über dem JFrame, der nicht [Form] beinhaltet, klicken, dann sehen Sie den aktuell (generierten) Code. Der hellblau unterlegte Teil wurde dabei durch den visuellen Editor erstellt und ist nicht direkt änderbar. Allerdings lässt sich die Code-Generierung im visuellen Editor unter dem Punkt Code (beim Properties-Fenster) beeinflussen.
- Wählen Sie nun JPanel aus der Palette und klicken Sie in den oberen Teil des JFrame. Da ein JFrame automatisch das BorderLayout besitzt, wird das JPanel mit dem Constraint NORTH versehen. Sie finden dies im Properties-Editor unter Layout. Ändern Sie dort den Wert und beobachten Sie, was passiert. Stellen Sie anschließend wieder zurück auf NORTH und fügen Sie noch ein zweites JPanel im Süden ein.
- In das obere JPanel setzen Sie zunächst ein JLabel und dann ein JTextField. Beachten Sie dabei, dass das Default-Layout eines JPanels das FlowLayout ist. Der Text des JLabel soll nun auf „Counter value“ gesetzt werden. Dies kann entweder im Properties-Editor oder durch einen (Doppel-)Klick auf das JLabel geschehen. Das JTextField setzen sie auf nicht-editierbar und den Text „0“.
- In das untere JPanel setzen Sie drei JButtons „Count“, „Reset“ und „Exit“.
- Wählen Sie den Count-Button und dann im Properties-Editor Events und klicken Sie auf das <none> in der Zeile von actionPerformed. Es erscheint ein Default-Eintrag, den Sie mit Return bestätigen. Die Ansicht wechselt daraufhin auf den Code und der Cursor steht an der Stelle, an der dieser Event behandelt wird. Fügen Sie hier einfach ein System.out.println(„Button pressed“) ein.
- Über das Icon „Test Form“ über dem JFrame können Sie die GUI optisch testen.
- Durch Aufruf von Build->Execute können Sie den Code ausführen. Hier sollte nun beim Klicken auf „count“ im unteren Fensterbereich von Netbeans die entsprechende Ausgabe zu sehen sein.

Eine korrekte (vollständige) Implementierung des Counters ist hier nicht von Interesse.

- Betrachten Sie den Code und versuchen Sie ihn zu verstehen. Erklären Sie, wie die GUI erstellt wird und wie die Ereignisbehandlung funktioniert.
- Experimentieren Sie mit verschiedenen Layout-Managern, indem Sie rechts im Inspector auf das [JFrame] mit der rechten Maustaste klicken und im Kontextmenü Set Layout auswählen. Die Einstellungen sind dabei in der Regel über den Properties-Editor durchzuführen. Einzige Ausnahme bildet das GridBagLayout, das über das Kontextmenü ein Customize anbietet, wodurch ein extra GridBagLayout Customizer geöffnet wird.
- Der GridBagLayout-Manager ist der komplizierteste aber auch „mächtigste“. Der NullLayout-Manager dagegen ist der einfachste und erlaubt exakte Positionierung von Elementen. Davon sollte allerdings im Hinblick auf Usability Abstand genommen werden. Das Verhalten der verschiedenen Layout-Manager ist durch lange Erfahrung im Design von Benutzerschnittstellen (Design-Guidelines) geprägt und in den meisten Fällen zu berücksichtigen statt zu versuchen, es zu umgehen.

Eine gute und verständliche Übersicht über die verschiedenen Layout-Manager findet sich auch in der Online-Edition des Buches „Java Programmierhandbuch und Referenz für die Java-2-Plattform“, Standard Edition, 3. Auflage 2002, Stefan Middendorf, Reiner Singer, Jörn Heid, im Kapitel 8.8 „Die Layout-Manager“ auf http://www.dpunkt.de/java/Programmieren_mit_Java/Oberflaechenprogrammierung/71.html

- Zum Abschluss sollen die verschiedenen Look-and-Feel-Varianten ausprobiert werden. Ohne spezielle Angaben entspricht das Look-and-Feel meistens der jeweiligen Plattform. Um ein anderes Look-and-Feel nutzen zu können, muss folgender Code zu Beginn der Main-Klasse hinzugefügt werden:

```
try {
    javax.swing.UIManager.setLookAndFeel( LookAndFeelName );
}
catch (Exception e) {
    System.out.println("LookAndFeel: Class not found!");
}
```

Dabei ist *LookAndFeelName* durch den String für das entsprechende Look-And-Feel zu ersetzen. Im Linuxpool der Uni kann man hier einmal folgende ausprobieren:
"javax.swing.plaf.metal.MetalLookAndFeel"
"com.sun.java.swing.plaf.motif.MotifLookAndFeel"
"com.sun.java.swing.plaf.gtk.GTKLookAndFeel"

Das plattformspezifische Look-And-Feel entspricht auf dieser Plattform dem CrossPlatformLookAndFeel, so dass hierbei keine Unterschiede zu sehen wären.