

Übung 1 – Medientechnik

Aufgabe 1:

Erstellen eines einfachen Taschenrechners in Java mit einer Swing-GUI nach dem Model-View-Controller-Prinzip.



Das Modell und der Controller dienen bereits als fertige Codevorlage und sind auf den Webseiten zu dieser Übung zu finden.

Die GUI soll ein Ergebnisfeld, ein Eingabefeld und mehrere Schaltknöpfe enthalten. Durch Klicken auf den „add“-Button soll die Zahl, die im Eingabefeld eingegeben wurde, jeweils zum Ergebnis dazugezählt werden. Durch einen Klick auf den „exit“-Button oder durch Anklicken des Fenster-Schließicons soll das Programm beendet werden.

Um die View zu erstellen gehen Sie wie folgt vor:

1. Erstellen Sie eine Klasse View1, die ein einfaches Hauptfenster der Größe 200x150 erzeugt.
2. Platzieren Sie die Dialogelemente indem Sie drei Panels erstellen. Im obersten Panel soll das Ergebnis erscheinen, in der Mitte erfolgt die Eingabe und im untersten Panel sollen die Buttons angeordnet sein. Wählen Sie dazu einen geeigneten Layout-Manager.
3. Versetzen Sie das Panel mit dem Textfeld zur Eingabe mit einem vorgestellten Label „Eingabe:“ und das Panel mit dem Textfeld für das Ergebnis mit dem Label „Ergebnis:“. Das Ergebnis-Textfeld soll nicht veränderbar sein und bei Programmaufruf den Wert „0“ enthalten.

Der Start des Taschenrechners soll durch eine Klasse Calculator erfolgen.

Aufgabe 2:

Erstellen Sie eine zweite View (als Klasse View2), die die bereits im Modell vorgesehenen Funktionen zur Subtraktion und Multiplikation ebenfalls in einer GUI zur

Verfügung stellt. Überlegen Sie sich hierfür ein geeignetes Layout (vergrößern Sie gegebenenfalls das Hauptfenster!).

Beachten Sie dabei, dass auch der Controller an die geänderte GUI angepasst werden muss! Erstellen Sie dazu eine Klasse Controller2 (kopieren Sie dazu die Klasse Controller und ergänzen bzw. passen Sie sie an die neue View2 an).

Fügen Sie ausserdem dem Taschenrechner noch die Funktion „Division“ hinzu.

Aufgabe 3 (HAUSAUFGABE):

Erstellen Sie nach der Model-View-Controller-Architektur eine GUI mit einer virtuellen Tastatur (d.h. für jeden einzugebenden Buchstaben einen Button), über die ein Text eingegeben werden kann, der während der Eingabe in einer darüberliegenden Textzeile erscheint. Das Textfeld soll nicht editierbar sein.

Fügen Sie ausserdem noch einen „clear“- und einen „exit“-Button hinzu. Über den „clear“-Button soll es möglich sein, die Textzeile ganz zu löschen.

Die Zeichen der virtuellen Tastatur sollen von „A“ bis „Z“ und „a“ bis „z“ gehen.

Optional können auch Eingabefelder implementiert werden, über die der Benutzer zunächst einen ASCII-Start- und Endwert angibt, und einen „create keyboard“-Button, bei dessen Klick dann entsprechend der Angaben die Tastatur dynamisch aufgebaut wird.

Abgabetermin: Montag, 25.04.2005

Die Aufgabe ist über das Online-Abgabesystem (zu finden auf den Webseiten zur Vorlesung auf <http://mimuc.de/mt>) abzugeben. Dazu bitte sämtliche Dateien in ein ZIP-Archiv packen und dieses hochladen. Die abgegebenen Java-Dateien müssen im Linuxpool kompilieren und ohne spezielle Umgebungen (also nicht nur in eclipse/Netbeans!) lauffähig sein. Bitte gegebenenfalls eine kurze Textdatei mit Anleitung zum Kompilieren, Starten und Bedienen mitliefern.

Diese Aufgabe kann nicht in Gruppen bearbeitet werden!

Allgemeine Hinweise zur Hausaufgaben-Regelung:

Es gibt drei Übungsblätter, die jeweils eine Hausaufgabe mit Java-Programmierung beinhalten. Davon müssen 2 abgegeben und als korrekt befunden sein.

Die angegebenen Abgabetermineine müssen eingehalten werden. Es gilt jeweils 0 Uhr (Ende) des angegebenen Tages.

Bei den drei praktischen Übungen (Laborexperimenten) gibt es ebenfalls jeweils eine Hausaufgabe, die eine Nachbereitung der vor Ort erstellten Inhalte darstellt. Hier sind alle 3 abzugeben.

Sind diese Voraussetzung alle erfüllt kann an der Klausur am Ende des Semester teilgenommen werden.

Bei Fragen oder Problemen bitte jeweils in den Übungen ansprechen, oder per email an

Siegfried.Wagner@medien.ifl.lmu.de oder

Nora.Zelhofer@stud.ifl.lmu.de oder

Marc.Breisinger@stud.ifl.lmu.de