

Grundlagen der Programmierung 2 SS 2005 - Aufgabenblatt 4

Ausgabe: 06.05.05

Aufgabe 13 (Entwurf eines Ereignis-Automaten)

In der Vorlesung wurde vorgestellt, wie man aus Ereignisfolgen den dazugehörigen endlichen Automaten zeichnet. Schauen Sie sich die Screenshots eines einfachen Bankautomaten an und entwerfen Sie den dazugehörigen endlichen Automaten. Eine entsprechende lauffähige Implementierung des Bankautomaten stellt die kompilierte Datei `Bankautomat.class` dar. Das Java Programm kann durch die Anweisung `'java Bankautomat'` ausgeführt werden.



1.

Zunächst gibt man die Karte ein. *Weiter* führt zum nächsten Bild, *Abbrechen* wieder zum ersten.



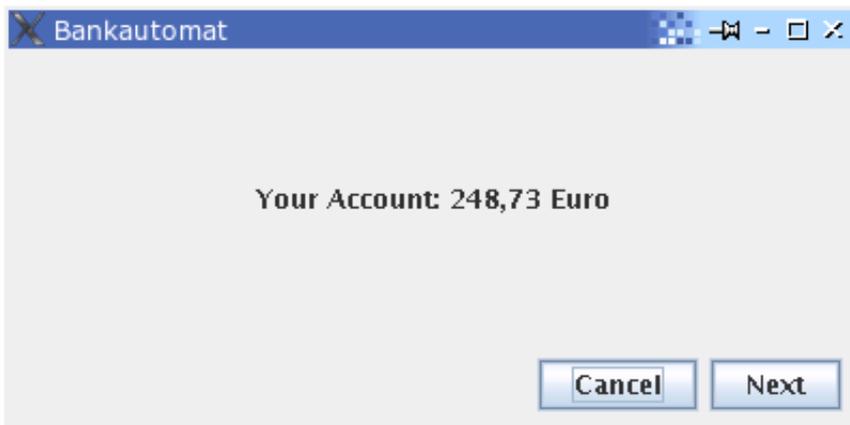
2.

Nun muss man die PIN-Nummer eingeben (1234 ist korrekt). Die Richtige PIN-Nummer führt zum nächsten Bild, eine falsche immer wieder zur Eingabe (Man kann beliebig oft eine falsche PIN-Nummer eingeben.). *Abbrechen* führt zum ersten Bild.



3.

Im Hauptmenü wählt man zwischen *Auszahlung* und *Kontostand*. Der *Kontostand* führt zum 4. Bild, die *Auszahlung* zum 5.



4.

Dieses Bild zeigt den aktuellen Kontostand. *Weiter* führt wieder in das Hauptmenü (3. Bild) und *Abbrechen* führt zum siebten Bild.



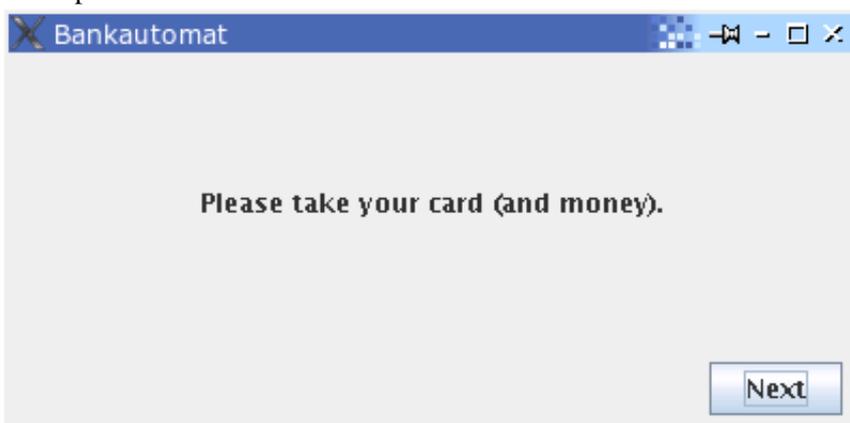
5.

Hier gibt man ein, wieviel Geld man abheben möchte. *Weiter* führt zum nächsten Bild, *Abbrechen* zum Hauptmenü.



6.

Dies ist eine Sicherheitsabfrage vor der Auszahlung. *Weiter* veranlasst die Auszahlung, *Abbrechen* führt ins Hauptmenü.



7.

Dies ist ein Hinweifenster. *Weiter* führt zum Start.

Aufgabe 14 (Model/View-Paradigma)

Erweitern Sie das in Kapitel 8 der Vorlesung vorgestellte Programm `MoneyTracker.java` um die Möglichkeit, Einträge aus der Liste zu löschen.

Dazu müssen Sie zuerst in der Model-Klasse eine entsprechende Methode ergänzen und anschliessend in der Fenster-Klasse einen Button für diese Funktionalität hinzufügen. Achten Sie dabei darauf Model und View jederzeit konsistent zu halten.

Aufgabe 15 (Implementierung eines Ereignis-Automaten)

a)

Modellieren Sie einen Ereignis-Automaten für den in der Abbildung dargestellten und in `Calculator.java` bereits teilweise implementierten Taschenrechner. Der Automat bzw. das Programm soll in den jeweiligen Zuständen nur sinnvolle Button-Selektionen erlauben. Jederzeit kann unter Verwendung des Buttons 'C' in den initialen Zustand gewechselt werden. Durch die Benutzung des Fenster-Schließ-Button kann die Applikation jederzeit beendet werden. Weiterhin ist in der Abbildung das Aussehen der GUI Komponenten in den jeweiligen Zustandsschritten bei der Berechnung von '12+23=' dargestellt. Dabei fordert man mit dem Zeichen '=' die Anzeige des Ergebnisses an. Alle angegebenen Operationen bestehen aus zwei Operanden, ein Zwischenergebnis kann unmittelbar mit einem weiteren Operand verknüpft werden. Stellen Sie sicher, dass die Operanden mindestens eine Ziffer enthalten.

Eingaben wie '+9=', '9+=', '9+-9=' oder '9+9-9=' sind für die Berechnung nicht erlaubt. Auch soll die Regel Punktrechnung vor Strichrechnung *nicht* beachtet werden. Die Eingabe von '10+9=-8=' würde ein korrektes Ergebnis mit der Ausgabe '11' liefern.

Der Taschenrechner hat folgende Tastenbelegung:

- Ziffern: 0,1,2,3,4,5,6,7,8,9
- Operatoren: +,-,*,/
- Ergebnis anzeigen: =
- alle Eingaben verwerfen: C

b)

In diesem Aufgabenteil soll der in `Calculator.java` bereits teilweise implementierte Taschenrechner um die Implementierung eines Ereignis-Automaten (entsprechend dem bereits in Aufgabenteil (a) modellierten) erweitert werden. Bei der Implementierung ist darauf zu achten, dass alle Buttons, die in den einzelnen Zuständen nicht benötigt werden, deaktiviert sind. Bitte beachten Sie bei Ihren Tests, dass die Integer-Werte in Java nur eine Größe von 32 Bit besitzen. Sie brauchen den Automaten nicht auf Überlaufbehandlung zu erweitern.

