

# Grundlagen der Programmierung 2 SS 2005 - Lösung 4

## Lösung zu Aufgabe 13

Zu dieser Aufgabe gibt es keine Musterlösung.

## Lösung zu Aufgabe 14

Zum Löschen wird der `ChequeBook`-Klasse (dem *Model*) die Methode `removeEntry(int index)` hinzugefügt.

```
// Eintrag loeschen
void removeEntry(int index)
{
    // falls index außerhalb der Listengroesse, Methode beenden
    if (index < 0 || index >= entries) return;

    // Eintrag loeschen und andere Eintraege eins vorruecken
    for (int i = index + 1; i < entries; i++)
    {
        purposes[i-1] = purposes[i];
        amounts[i-1] = amounts[i];
    }
    entries -= 1;

    // Beobachter (View) informieren
    fireIntervalRemoved(this, index, index);
}
```

Diese Funktion wird dann vom Löschen Button wie folgt aufgerufen.

```
// Ereignisbehandlung für den "Loeschen"-Button
deleteButton.addActionListener(new ActionListener()
{
    public void actionPerformed (ActionEvent e)
    {
        listData.removeEntry(list.getSelectedIndex());
    }
});
```

Die vollständige Lösung befindet sich in der Datei `MoneyTracker.java`

## Lösung zu Aufgabe 15

a)

Folgendes ist beim Entwurf des Ereignisautomaten für den Taschenrechner zu beachten:

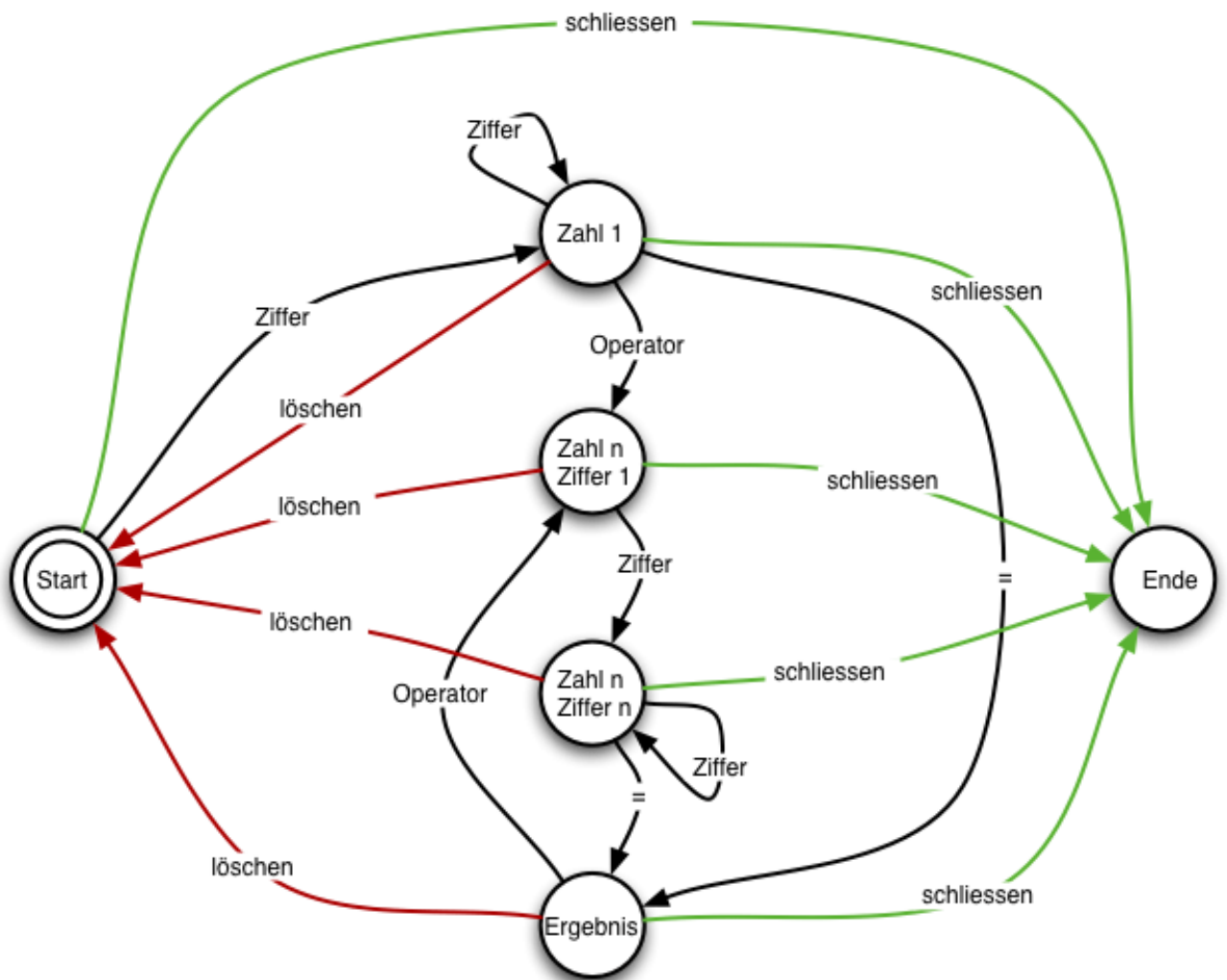
Regeln in Worten:

- alle Operatoren sind 2-stellig und
- jedes Zwischenergebnis muss mit '=' angefordert werden
- Das heißt, jede Eingabe beginnt mit *Operand Operator Operand '='* oder *Operand '='* darauf folgen beliebig oft *Operator Operand '='*

Als regulärer Ausdruck:

- $Operand ( Operator Operand '=' | '=' ) ( Operator Operand '=' )^*$
- Ein Operand besteht aus Ziffern, mindestens einer:  $Ziffer^+$  oder  $Ziffer Ziffer^*$
- Ein Operator ist '+' '-' '\*' '/'
- Eingabe von 'C' ist immer erlaubt und führt in den Initialzustand

Ereignisautomat des Taschenrechners:



- b) Bei der Implementierung des Automaten müssen folgende Teilaufgaben gelöst werden:
- Zustände codieren und aktuellen Zustand initialisieren
  - Rechenwerk: Operatoren codieren, speichern, Zwischenergebnisse
  - In der Reaktionsmethode Eingabe verarbeiten:  
Ziffern, Operatoren, '=', 'C'
  - In der Methode `toState`, Aktionen bei Erreichen von Zuständen programmieren, z. B. Buttons (de)aktivieren

Die entsprechende Java Implementierung ist in `Calculator.java` realisiert.