



Praktikum zu LV 7328 - Ruby: **Übung 10**

GUI-Entwicklung mit FXRuby



Organisatorisches



- Arbeitsverzeichnis:

`~/lv/ruby/10/`

- Dateinamen:

`10-fxlotto-a.rb` # Abzugeben (Teil A, nach 1. Woche)

`10-fxlotto-b.rb` # Abzugeben (Teil B, nach 2. Woche)

`10-lotto.rb` # Zu benutzen, aus Übung 03 abgeleitet

- Werkzeuge:

`ruby` # Der Interpreter

`emacs` # mit Ruby-Mode. Auch X-Emacs ok

`scite, irb, ri` # Optionale Tools, wie üblich

- Vorlagen:

`10-fxlotto.rb`



Überblick zur Aufgabe



- Die Aufgabe: **GUI-Steuerung für die Lottoziehung**
Grundlage: Ergebnisse der Aufgabe 03
- Die Aufgabe besteht aus zwei Teilen:
 - A: Einfache graphische Benutzeroberfläche**
Basisfenster, Menü, Startknopf
Ausgabe der Lottoziehungsergebnisse per MessageBox.
 - B: Komfortable graphische Benutzeroberfläche**
Dreiteiliges Fenster: Links und rechts Kontrollen,
unten Ausgabefenster
Optionen: Wahl des Ziehungsverfahrens, Wahl der
Zahlen (zu ziehen, insgesamt), Zusatzzahl ja/nein

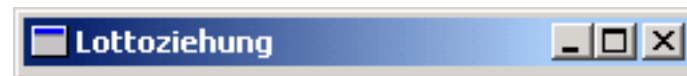


Die Aufgabe



Teil A, Vorbereitungen:

- **Legen Sie eine Datei "10-lotto.rb" an**
Diese Datei soll nur die Implementierung der Klasse "Lotto" (incl. aller Methoden) aus Aufgabe 03 enthalten.
- Kopieren Sie dazu `03-lotto.rb` nach `10-lotto.rb` und entfernen Sie alle unnötigen Zeilen.
- Kopieren Sie die Vorlage `10-fx1otto.rb` aus dem Dozentenverzeichnis in Ihren Arbeitsbereich als `10-fx1otto-a.rb`.
- **Testen Sie `10-fx1otto-a.rb`.** Fehler sollten nicht auftreten . Es sollte ein leeres Fenster erscheinen, das nur aus einer Titelzeile besteht:



- Beseitigen Sie eventuelle Fehlermeldungen



Teil A, Vorgaben:

- Fügen Sie eine Menüleiste hinzu, mit **Menüpunkten "Datei" und "Hilfe"**.
- Menüpunkt "Datei" soll als einzigen **Eintrag "Beenden"** enthalten. Bei Aktivierung soll das Programm beendet werden.
- Menüpunkt "Hilfe" soll den einzigen **Eintrag "Über Lottoziehung"** enthalten. Bei Aktivierung soll eine Mitteilung mit einem Copyright-Vermerk in einem Unterfenster ("FXMessageBox") erscheinen und durch Klick auf einen "OK"-Knopf wieder verschwinden.
- Das Hauptfenster soll einen **Knopf "Ziehung starten"** enthalten. Bei Aktivierung soll das Ergebnis der **Lottoziehung per MessageBox angezeigt** werden.
- Beachten Sie das Model-View-Controller Design; halten Sie es ein!

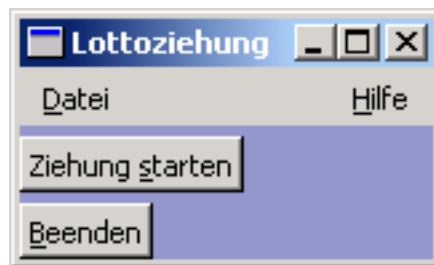


Die Aufgabe

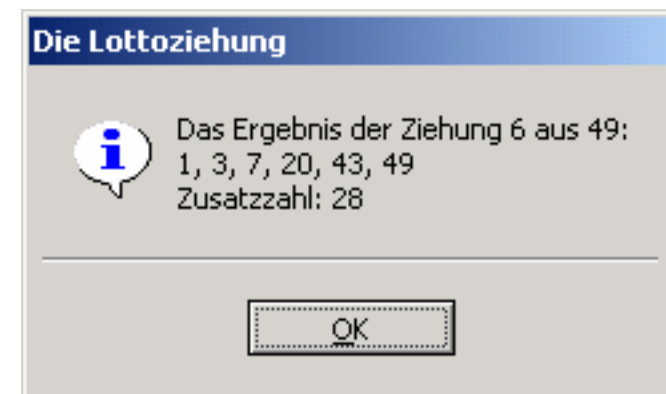


Teil A, Hinweise:

- Verwenden Sie die Code-Beispiele aus der Vorlesung zur Realisierung vom Menü, von Knöpfen und MessageBox-Ausgaben. Hier einige Beispiel-Anzeigen:



- (Einbau des Knopfes "Beenden" ist optional.)
- Ordnen Sie den Widgets Aktivitäten zu, wie in den Vorlesungsbeispielen.
- Achten Sie auf **Bedienbarkeit auch per Keyboard!** Tooltips sind optional.



Organisatorisches

- Für die Erfüllung von Teil A gibt es einen Punkt, für die von Teil B einen weiteren.
 - Jeder Aufgabenteil wird separat abgegeben!
 - Dies ist eine Doppel-Übung, d.h. es gibt 2 Wochen Bearbeitungszeit.
- Hinweise zu Teil B
 - Kopieren Sie: `cp 10-fxlotto-a.rb 10-fxlotto-b.rb`, bearbeiten Sie die neue Datei und geben Sie auch diese ab.
 - Teil B erfordert mehr Arbeit als Teil A und setzt die erfolgreiche Bearbeitung von Teil A voraus.
 - Er erreicht schon ein realistisches Niveau einer Benutzeroberfläche.
 - Die Arbeitsweise für Teil B ist selbständiger als für Teil A: Während Teil A durch Übertragung der Vorlesungsbeispiele bereits lösbar ist, werden für B mehrere neue Widgets benötigt. Die Einzelheiten zu deren Funktionsweisen sind selbständig zu recherchieren.



Teil B, Vorgaben



- Das **Hauptfenster** wird **dreigeteilt**:
 - "Oben links", "oben rechts" und "unten"
 - Nur der untere Teil soll sich bei "resize"-Aktionen verändern lassen
- Oben links:
 - Gruppe (FXGroupBox) von drei FXRadioButton-Objekten, zur Auswahl des Ziehungsverfahrens.
 - Darunter: "Ziehung starten"-Knopf und "Beenden"-Knopf
- Oben rechts:
 - Gruppe von 2 FXComboBox-Objekten, zur Auswahl der Anzahl jeweils zu ziehender Zahlen (möglich: 2, 4, 6, 8) und der Grundmenge, aus der gezogen wird (möglich: 10, 49, 99).
Voreinstellungen: 6, 49
 - Ein FXCheckBox-Objekt zur Wahl "Mit Zusatzzahl"
 - Je ein FXLabel-Objekt vor den Combo-Boxen mit Text nach Vorlage.
- Unten: Ein FXText-Objekt zur Anzeige der Ergebnisse



Teil B, Hinweise



- Vorlage
 - Siehe nebenstehende Abbildung
- Informationsquellen
 - Quellcode der FXRuby-Beispiele aus dem "User guide" (URL siehe Vorlesung)
 - Wer am eigenen PC entwickelt: Quellcode aus der Windows-Installation von Ruby
- Ziel des Teils B
 - Es kommt weniger auf den exakten Nachbau der nebenstehenden Oberfläche an, als vielmehr auf den Erwerb der Fähigkeiten, sich eigenständig Kenntnisse über weitere FXRuby-Widgets anzueignen.

