
Funktionale Programmierung

<http://proglang.informatik.uni-freiburg.de/teaching/functional-programming/2013/>

Übungsblatt Abschlussprojekt: Grafischer Taschenrechner

Do, 2014-02-13

Abgabe: Fr, 2014-03-07

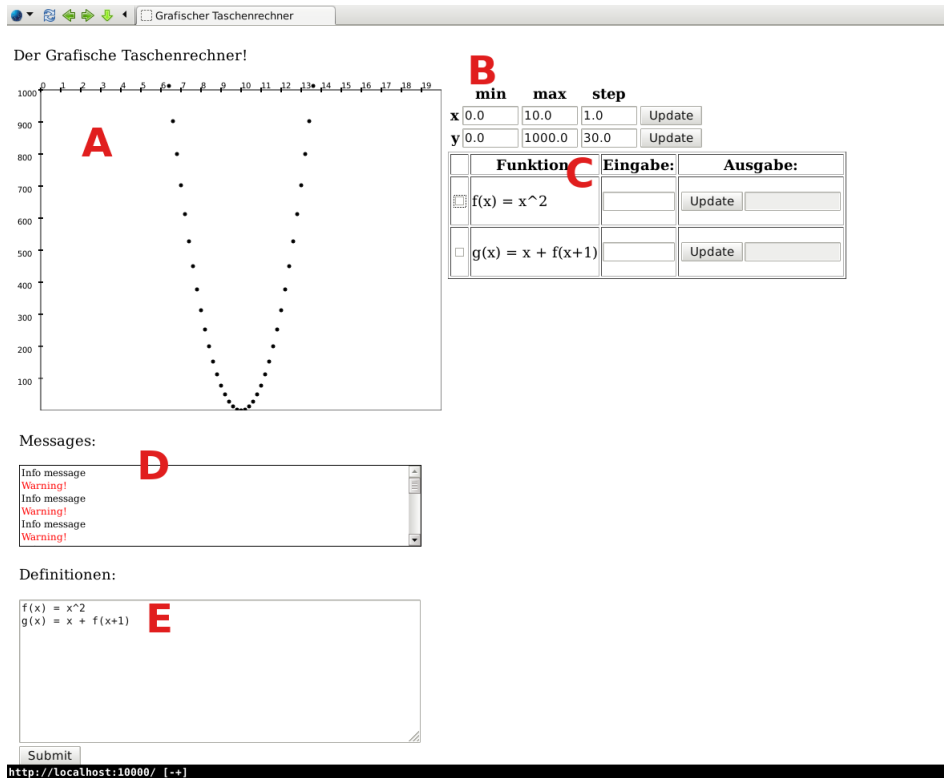
Errata:

- 2014-02-14: Hinweis zu `threepenny-gui` Version hinzugefügt.

Hinweise:

- Die Bearbeitung des Projekts ist **freiwillig**. Beim erfolgreichen Implementieren der Basisversion (s.u.) erhalten Sie einen Bonus von einer Notenstufe auf die Note Ihrer bestandenen mündlichen Prüfung. Beim erfolgreichen Implementieren der erweiterten Version erhalten Sie zwei Notenstufen Bonus.
- Die Benutzeroberfläche des Taschenrechners wird mit `threepenny-gui` implementiert (siehe auch Blatt 6). Die notwendigen GUI-Elemente sind allerdings schon im Modul `Widgets.hs` (verfügbar auf der Vorlesungshomepage) gekapselt; die Beispiele in `Calculator.hs` sollten zur Bewältigung des Projekts ausreichen. Falls es allerdings Probleme mit den Widgets geben sollte, oder sie Funktionalität vermissen, melden Sie sich bitte sofort beim Assistenten.
- Das Modul `Widgets.hs` funktioniert nur mit `threepenny-gui` Version 0.3.* und *nicht* mit `threepenny-gui` Version 0.4.*. Um zu testen welche Version installiert ist, führen Sie den Befehl `ghc-pkg list threepenny-gui` in einer Shell aus. Falls die falsche Version installiert ist, löschen Sie am Besten das Verzeichnis `$HOME/.ghc` und führen Sie den Installationsbefehl `cabal install threepenny-gui-0.3.0.1` aus.

Beschreibung der Funktionalität und Benutzeroberfläche: Auf der Vorlesungshomepage finden Sie die Module `Calculator.hs` und `Widgets.hs`. Wenn Sie die `main` Funktion in `Calculator.hs` aufrufen und die Adresse `localhost:10000` in ihrem Browser öffnen, müssten Sie ein Mock-up der gewünschten Benutzeroberfläche sehen. Der Beispielpcode in `Calculator.hs` demonstriert wie die Widgets verwendet werden können.



Der Rechner erlaubt dem Benutzer das Eingeben von einfachen Funktionen (wie $f(x) = x^2$). Diese können dann punktweise ausgewertet oder in einem Plot-Fenster dargestellt werden.

- Das Plot-Fenster **A** stellt ausgewählte Kurven dar.
- Mit den Eingabefeldern in **B** kann der Plotbereich festgelegt werden. Anklicken von **Update** passt die dargestellten Kurven entsprechend an.
- Die eingegebenen Funktionen sind in **C** aufgelistet. Wenn das Häkchen in der ersten Spalte angewählt ist, soll die Funktion im Plotfenster dargestellt sein. In dem Eingabefeld in der zweiten Spalte kann ein x -Wert angegeben werden; durch Anklicken von **Update** in der dritten Spalte wird dann der entsprechende Funktionswert im Ausgabefeld angezeigt.
- Im Nachrichtenfeld **C** werden Eingabefehler und andere Nachrichten dem Benutzer mitgeteilt.
- Im Editor **E** kann der Benutzer neue Funktionen definieren. Durch Anklicken von **Submit** werden die zu Verfügung stehenden Funktionen in **C** entsprechend der Eingabe neu gesetzt.

Aufgabe 1 (Basisversion)

Implementieren Sie die oben beschriebene Funktionalität, aber ignorieren Sie den einstellbaren Plotbereich (d.h. wählen Sie einen festen Bereich). Der Benutzer soll seine Funktionen durch Addition, Subtraktion, Multiplikation und Potenzierung mit positiven Exponenten (+, -, *, ^) ausdrücken können. Fangen Sie fehlerhafte Eingaben ab und geben Sie entsprechenden Fehlermeldungen im Nachrichtenfenster.

Aufgabe 2 (Erweiterte Version)

Fügen Sie zur Basisversion die Operationen Division (/) und Potenzierung mit negativen Werten hinzu. Es soll außerdem möglich sein, sich in den Definitionen auf andere definierte Funktionen zu beziehen. Zum Beispiel sollte eine Editor-Eingabe wie die folgende funktionieren:

$$f(x) = x^2$$

$$g(x) = x + f(x+1)$$

Erlauben Sie zusätzlich das Einstellen des Plotbereichs. Achten Sie dabei darauf, dass die Kurven auch beim „heranzoomen“ erkennbar dargestellt werden (sofern Sie im Plotbereich liegen).