Proaktive Einführung in BASH, PHP und Programmierung / programmier_projekt_PEBPP

Wir bauen eine Adressverwaltung!

Warum?

Weil eine Adressverwaltung alle Anforderungen an eine Programmiersprache abdeckt, um diese kennenzulernen.

Idee

Um einen Einstieg zu geben, dem möglichst alle Leute folgen können, werden hier die Vorgaben beschrieben damit Du in Eigenarbeit, Stück für Stück, ein lauffähiges Programm schreiben kannst. Gedacht ist dabei, im Wechsel selber zu programmieren und das jeweilige Ergebnis dann beim LUG-Treffen vorzustellen und weiter zu optimieren. Lösungen zu erarbeiten und sich so in die Basics der Programmierung vorzutasten. Hier werden die Schritte nach und nach veröffentlicht, und vielleicht auch das ein oder andere Programm vorgestellt, oder es gar in Entwicklung in einem Pad zu zeigen. Das Pad kann auch gut zum gemeinsamen Arbeiten und/oder Austausch genutzt werden.

Eine erste Version wird in BASH gebaut, wenn diese fertig ist werden wir sie nochmals in PHP (Konsole only) bauen.

Die Idee dahinter ist, das Du sowohl die grundsätzliche LINUX-System Programmierung, wie auch programmieren im allgemeinen und durch das Umsetzten des fertigen Programmes ebenfalls erste Schritte in Richtung Web-Programmierung kennen lernst.

Werkzeuge

- Pad / Zum Austausch oder gemeinsamen Arbeiten.
    - ABS.PDF (2.6MB) / Kompendium zu BASH, das Nachschlagewerk für alle Fälle
- BASH / Die zu benutzende Shell
- PHP (5-7) / der PHP-Interpreter
- http://php.net / Die Informationsquelle zu PHP
- Dateiverwaltung und Editor: https://www.midnight-commander.org/
- Editoren: mcedit, joe, kwrite, kate

In eigener Sache: Wer mal hier reinlesen möchte, meldet sich bitte bei mir für Zugang.

Teil 1 / Vorgaben zum Adressbuch

Ein Adress-Programm in BASH habe ich schon in den Neunzigern geschrieben es ist 5382 Zeichen (incl. Kommentaren) groß. Das sollte also zu schaffen sein. 😉 — *Franke 2016-10-13 06:26*

Das Programm soll folgendes können:

- Menu
    - Eingabe

    - Ausgabe
    - Sortieren
    - Suchen
    - Loeschen
    - UNDO
    - Source
    - Hilfe
    - ENDE
- Verarbeitet werden Adress- oder Freiformdaten, b.B.: Vorname, Nachnahme, Strasse, Telefon, Email, Keywords, Variablen,...
- Daten werden nur als Textonly (ASCII) verarbeitet und gespeichert.
- Kommentare im Programm sind Pflicht.
- Jedes Programm enthält einen Versionsschlüssel in der Form:
    - $VER: PROGRAMMNAME 0.0.1.5 (2016-10-13) (2016-01-01) PROGRAMMIERER $
    - $_VERDESC: NAME major-release.minor-release.major-bugfix.minor-changes DT-STAMP-CHANGE DT-STAMP-CREATE CREATOR-STRING $
- Jedes veröffentlichte Programm trägt einen Dateinamen nach folgenden Schema (Sonst können wir hier die Programme nicht zuordnen oder vergleichen):
    - PROGRAMMNAME.0.0.1.5.PROGRAMMIERER.DT-STAMP-CHANGE.bash
    - fkn_adr.0.0.1.2.franke.20161012.bash
- Speichert auch für euch jede Version, dann könnt Ihr nachvollziehen was Ihr gemacht habt.
- Es ist sinnvoll zum jeweiligen Programm auch eine Demo Datei mit Dummy-Einträgen (also keine existierende Adressen etc.) zu haben, diese sollte dann auch dem Programm zuzuordnen sein, also besser ebenfalls nach dem Namensschema anlegen: fkn_adr.0.0.1.2.franke.20161012.db
- Sendet mir nach jedem Treffen eure jeweilige Version damit ich die hier reinstellen kann, dann kann jeder vom anderen lernen.

Viel Erfolg!

Programm Versionen

Ein erstes Rumpfprogramm seht Ihr hier:

adressprogramm_in_bash.0.0.0.1.Franke.bash

```
#!/bin/bash
################################################
# $VER: adressprogramm_in_bash 0.0.0.1 (20020712/Franke) (19971001) FKN-
Systems/Ng $##
# Adress-Demo in Bash-Script ######################
################################################

# DECLARATIONS / VARIABLES / FUNCTIONS / ... ##################################


# START ###############################

clear
echo
echo "Willkommen zum ..."
echo
echo "dieses Programm ist ..."
```

```
echo
echo "Es dient dem Erfassen, suchen, loeschen von Datensaetzen aller Art"
echo "------------------------------------------------------------------------"

# MAIN ##############################

# ...

exit 1
```

Testschnipsel für Teil: 1

_preb_case.test

```
#!/bin/bash


# Variablen Zuweisung:

KEY='value'
Variablenname='Ein Text'
FOO='BAR'

echo $FOO
echo ${FOO} # das sollte IMMER so geschrieben werden

echo 'BAR ${FOO} FOO' # einfache Ticks entwerten
echo "BAR ${FOO} FOO" # doppelte Ticke entwerten NICHT


select VAR in AA BB CC
do

    case ${VAR} in

        AA) ####################
            echo 'FOO'
            ;;
        BB) ####################
            echo 'BAR'
            ;;
        *) ######################
            echo 'DEFAULT'
            ;;
    esac

done
```

Etappen

1. Implementiert ein Auswahlmenu und seht die einzelnen Programmfunktionen vor.
    - Dazu ist z.B. eine case-Auswahl geeignet

2. Ausbau der einzelnen Programmfunktionen.
3. ...