Patrick Felschen, Matr. Nr. 932056

Julian Voß, Matr. Nr. 934505

Aufgabenblatt 2

Aufgabe 1

Wird das Skript ausgeführt, wird der Nutzer über eine Konsolenausgabe aufgefordert, einen Pfad anzugeben. Diese Eingabe wird in eine Variable eingelesen. Nachfolgend wurden für jeden Eigenschafts-Test des Pfades Funktionen definiert. Mit dem „test“ Befehl lassen sich Pfade auf verschieden Eigenschaften überprüfen.

Parameter	**Funktion**
	Überprüft, ob Expression True ist, gibt True zurück, wenn Verzeichnis/Pfad existiert
-d	Gibt True zurück, falls der Pfad ein Verzeichnis ist
-f	Gibt True zurück, falls der Pfad eine Datei ist
-L	Gibt True zurück, falls der Pfad ein symbolischer Link (Verknüpfung) ist
-O	Gibt True zurück, falls der Nutzer Besitzer der Datei ist

Über den „stat“ Befehl können weitere Informationen ausgelesen werden. Durch den Parameter „-c“ kann ein bestimmtes Format angegeben werden, welches die Ausgabe haben soll. Das Format „%U“ gibt den Besitzer des angegebenen Pfades aus.

Die verschiedenen Funktionen können nach der Deklaration über den Befehl „<Methodenname> /bin/bash“ ausgeführt werden.

```
#!/bin/bash

echo "Bitte Pfad angeben: "
read DIR
echo "Angegebener Pfad: $DIR"

function pathExists {
        if test -a $DIR; then
                echo "Pfad exisitiert."
        else
                echo "Pfad existiert nicht."
        fi
}

function fileOrDir {
        if test -d $DIR; then
                echo "Es handelt sich um ein Verzeichnis."
        elif test -f $DIR; then
                echo "Es handelt sich um eine Datei."
        fi
}

function isSymbolicLink {
        if test -L $DIR; then
                echo "Es handelt sich um einen symbolischen Link."
        fi
}

function isOwner {
        if test -O $DIR; then
                echo "Aufrufer ist Besitzer"
        else
                echo "Aufrufer ist nicht Besitzer"
        fi
}

function printOwner {
        echo "Besitzer: " $(stat -c '%U' $DIR)
}

pathExists /bin/bash
fileOrDir /bin/bash
isSymbolicLink /bin/bash
isOwner /bin/bash
printOwner /bin/bash

exit 0
```

Aufgabe 2

Auf die im Befehl angegebenen Parameter kann im Skript über die „$@“ Liste zugegriffen werden. Für jeden Parameter läuft die „for“-Schleife einmal durch. Die einzelnen Elemente werden im Schleifendurchlauf in die Variable „DIR“ gespeichert.

```
for DIR in "$@"
do
        echo "Angegebener Pfad: $DIR"
        pathExists /bin/bash
        fileOrDir /bin/bash
        isSymbolicLink /bin/bash
        isOwner /bin/bash
        printOwner /bin/bash
done

exit 0
```

Aufgabe 3

Die letzten vier Zeichen des Pfades werden ausgelesen und überprüft, ob sie mit „.txt“ übereinstimmen. Ist dies der Fall, wird der Nutzer gefragt, ob die Datei geöffnet werden soll. Bestätigt der Nutzer dies mit der Eingabe „yes“ oder „y“, öffnet sich die Datei per „nano“ Befehl. Alternativ kann die Abfrage durch eine „no“ oder „n“ Eingabe abgebrochen werden. Bei anderen Eingaben wird die Abfrage wiederholt.

```
function checkExtension {
        if [ "${DIR: -4}" == ".txt" ]; then
                echo "Datei hat Endung .txt"
                echo "Soll die Datei geöffnet werden? (yes/no)"
                read command
                while [ "$command" != "y" ] && [ "$command" != "yes" ] && [ "$command" != "n"
] && [ "$command" != "no" ];
                do
                        echo "Soll die Datei geöffnet werden? (yes/no)"
                        read command
                done

                if [ "$command" == "y" ]; then
                        nano "$DIR"
                fi

        else
                echo "Datei hat keine .txt Endung"
        fi
}


for DIR in "$@"
do
        echo "Angegebener Pfad: $DIR"
        pathExists /bin/bash
        fileOrDir /bin/bash
        isSymbolicLink /bin/bash
        isOwner /bin/bash
        printOwner /bin/bash
        checkExtension /bin/bash
done

exit 0
```