

Erste Sonderübung im SoSe2020

Liebe Teilnehmer der Sonderübung,

aufgrund der aktuellen Lage müssen wir - Thomas Klütz und Nadine Schärmann - euch auf diesem Wege begrüßen. In der erste Woche gibt es zwei Aufgaben, die ihr selbstständig zu Hause lösen könnt. Habt ihr dazu Fragen, dann wartet nicht, schreibt uns eine E-Mail oder kontaktiert uns über den Chatraum für Übungen im Portal Matrix.

Ziel dieser ersten Übung ist es, dass ihr wieder reinkommt ins Programmieren und erkennt was ihr möglicherweise vom letzten Semester wiederholen müsst.

Wir möchten ab der nächsten Woche (trotz Feiertag!) die Sonderübung nicht nur hochladen, sondern wir planen eine kleinen Podcast oder ein kurzes Video zu den Inhalten jeder Woche.

Nach Ostern beginnen wir das neue Thema Pointer/Zeiger, welches uns für die nächsten Wochen begleiten wird.

Alle Materialien werden hier <https://gitlab.mn.tu-dresden.de/wwalter> hochgeladen und auch Lösungsvorschläge für die Programme findet ihr immer in der darauffolgenden Woche.

Unsere E-Mail-Adressen:

thomas.kluetz@mailbox.tu-dresden.de

nadine.schaermann@mailbox.tu-dresden.de

Passwortchecker

Das folgende Programm soll überprüfen, ob ein Passwort gewissen (Sicherheits-)Anforderungen genügt. Die Maximallänge eines Passworts beträgt dabei 20 Zeichen. Dies darf bei allen Funktionen/Subroutinen verwendet werden.

Schreibe ein Modul *Checker*, das folgende Unterprogramme enthält:

- eine Funktion *keineLZ* mit einem Rückgabewert vom Typ Logical und einem Parameter vom Typ Character, die testet, ob das eingegebene Passwort (außer am Ende) keine Leerzeichen enthält.
- eine Funktion *sicherGenug* mit einem Rückgabewert vom Typ Logical und einem Parameter vom Typ Character, die testet ob das eingegebene Passwort mindestens 1 Großbuchstaben, mindestens 1 Kleinbuchstaben und mindestens eine Ziffer enthält.
- eine Funktion *langGenug* mit einem Rückgabewert vom Typ Logical und einem Parameter vom Typ Character, die testet ob das eingegebene Passwort mindestens 8 Zeichen lang ist. Auf das Vorhandensein von inneren Leerzeichen muss dabei nicht geachtet werden, da dies durch die Funktion *keineLZ* kontrolliert wird.
- eine Funktion *Kreativ* mit einem Rückgabewert vom Typ Logical und einem Parameter vom Typ Character, die testet, ob das eingegebene Passwort ein sehr verbreitetes, unkreatives Passwort (z.B. Passwort1234, 123456789Abc, Hallo123) ist. Dazu wird zuerst die Datei "BekanntePasswörter.txt" geöffnet. Aus ihr werden die beliebten Passwörter der Reihe nach ausgelesen und mit dem eingegebenen Passwort verglichen. Stimmt das Passwort mit einem Eintrag der Datei überein, so wird false zurückgegeben, ansonsten true.

Schreibe außerdem ein Hauptprogramm. In diesem wird zuerst ein Nutzernamen (maximal 20 Zeichen) eingelesen. Danach wird in einer Schleife solange ein Passwort und eine Passwortbestätigung eingelesen, bis alle der folgenden Anforderungen erfüllt sind:

- Passwort und Nutzernamen stimmen nicht überein.
- Das Passwort enthält keine inneren Leerzeichen.
- Das Passwort ist mindestens 8 Zeichen lang.
- Das Passwort genügt den Sicherheitsanforderungen.
- Das Passwort ist kreativ genug.
- Passwort und Passwortbestätigung stimmen überein.

Es sind jeweils entsprechende Fehlermeldungen auszugeben.

Caesar-Code

Schreibe eine Funktion, die jeden Buchstaben eines Wortes in eine Zahl entsprechend seiner Stelle im Alphabet umwandelt. Alle Zeichen außer Buchstaben sollen unverändert bleiben.

Beispiel:

"Hallo W5lt!"

"8 1 12 12 15 23 5 12 20!"