

## 4.1.2 Fit For Prüfung

Montag, 30. Januar 2023 14:57

### 4.1.2. Fit for Prüfung

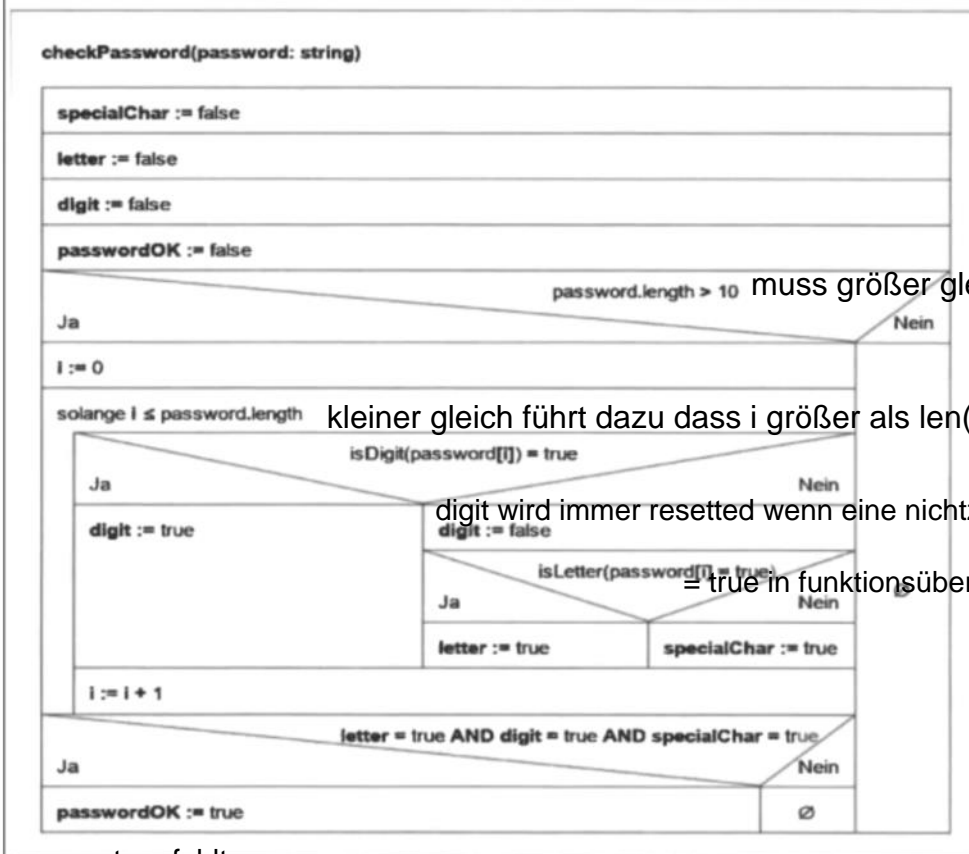
- c) Die Online-Terminvergabe soll nur über einen Login möglich sein. Dazu müssen die Benutzer einen Account erstellen und ein Passwort vergeben, welches folgende Kriterien erfüllen muss:

- Es muss mindestens zehn Zeichen lang sein.
- Es muss mindestens einen Buchstaben (letter), eine Ziffer (digit) und ein Sonderzeichen (specialChar) enthalten.

Es liegt ein fehlerhaftes Struktogramm zur Überprüfung des Passwortes vor. Der Rückgabewert soll wahr oder falsch sein.

Finden Sie drei Fehler und geben Sie eine Korrektur an.

6 Punkte



muss größer gleich sein

kleiner gleich führt dazu dass i größer als len(password) wird

digit wird immer resettet wenn eine nichtzahl mit drin ist

= true in funktionsübergabe

**Prüfung AP1 2022**

Sie planen, eine eigene Lösung für eine automatisierte Konfiguration der Standardarbeitsplätze zu programmieren. Aus einer Datenbank werden alle zu konfigurierenden PCs ausgelesen. Danach wird für jeden PC aus der Datenbank die zu installierende Software abgefragt und auf dem PC installiert.

Es gibt die folgenden Variablen:

PCNr                   Ganzzahl – Laufvariable

SoftwareNr   Ganzzahl – Laufvariable

Es gibt die folgenden Felder (Array)

PCListe[]           Stringliste mit den Namen der PC

SoftwareListe[]   Stringliste mit den Namen der Software

Es stehen Ihnen die folgenden Funktionen zur Verfügung:

getPC() – liefert eine Liste von PC-Namen aus der Datenbank

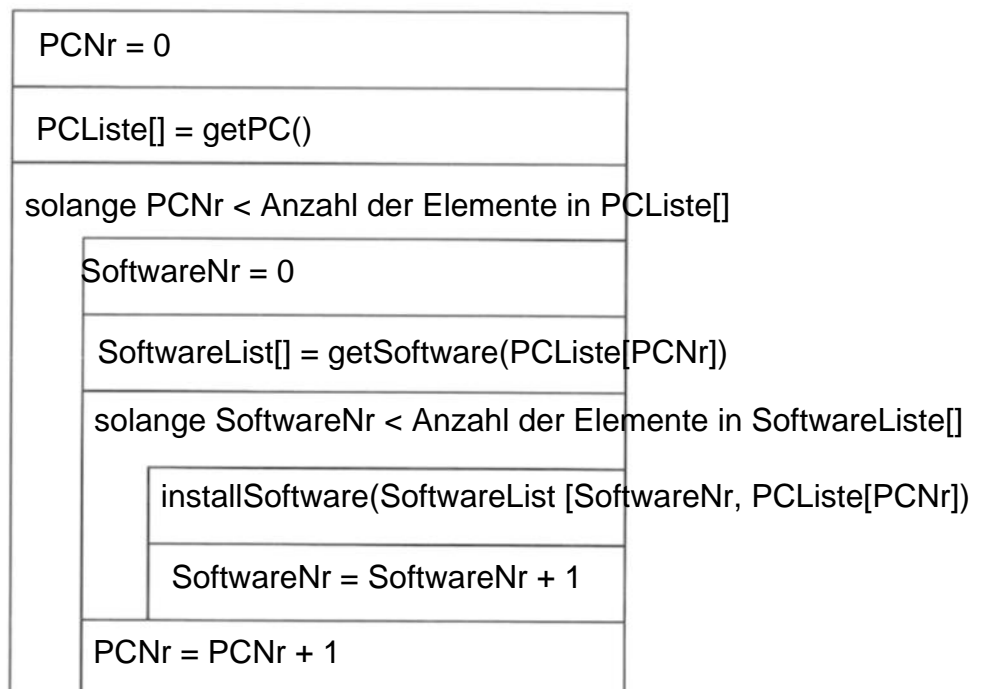
getSoftware(String) – liefert zu dem angefragten PC eine Liste der zu installierenden Software

installSoftware(String, String) Installiert die im ersten String angegebene Software auf dem im zweiten String übergebenen PC

Tragen Sie die Anweisungen folgerichtig in das nebenstehende Struktogramm ein.

9 Punkte

1. installSoftware(SoftwareListe [SoftwareNr], PCListe[PCNr])
2. Solange SoftwareNr < Anzahl der Elemente in SoftwareListe []
3. PCListe[] = getPC()
4. PCNr = PCNr + 1
5. PCNr = 0
6. SoftwareListe[] = getSoftware(PCListe[PCNr])
7. SoftwareNr = 0
8. SoftwareNr = SoftwareNr + 1
9. Solange PCNr < Anzahl der Elemente in PCListe[]



Die abgefragten Produktionsdaten werden über eine entsprechende API an die Steuerung der Walzanlage übergeben. Die Auftragsdaten werden im Array `result[]` mit dem Index 0 bis 3 gespeichert. Sie sollen jetzt an die Steuerung der Walzanlage durch eine von Ihnen zu erstellende Funktion übergeben werden. Gehen Sie von einem Array `result[]` aus, bei dem im Index 0 die Breite, im Index 1 die Länge, im Index 2 die Dicke und im Index 3 die Anzahl der zu produzierenden Wellpappen stehen.

Erstellen Sie die Funktion „**launchTask(result[])**“.

Zur Kommunikation mit der Steuerung der Walzanlage stehen Ihnen die folgenden API-Funktionen zur Verfügung:

**setRollerDim(int,int,int)** – Übergeben wird Breite, Länge und Dicke der Wellpappe.

**rollerStart()** – Startet einen Auftrag von einem Stück. Es wird eine Wellpappe mit den gesetzten Parametern erzeugt.

Die Walzanlage verfügt über einen Notausschalter. Sie darf nur laufen, wenn der Notaus nicht ausgelöst ist.

Der Status des Notausschalters kann mit der Funktion **bool getEmergencyStop()** abgefragt werden, der „true“ liefert wenn der Notaus ausgelöst ist und „false“ wenn der Notaus nicht ausgelöst ist.

Ergänzen Sie das gegebene Struktogramm durch die entsprechenden Befehle zur Produktion der geforderten Anzahl von Wellpappen (siehe Index 3) in den angegebenen Maßen (siehe Index 0, 1 und 2). 7 Punkte

<b>launchTask(result[])</b>
<b>int i = 0</b>
<b>bool emergencyStop = getEmergencyStop()</b>
<b>setRollerDim(result[0], result[1], result[2])</b>
<b>while not emergencyStop and i &lt; result[3]</b>
<b>i += 1</b>
<b>rollerStart()</b>
<b>emergencyStop = getEmergencyStop()</b>