MES programmieren

Für das, was wir im Weiteren tun, ist die Modellfabrik entwickelt worden. Sie sollen durch praktische Arbeit einen vertieften Einblick in die Leitstruktur moderner Industrieanlagen gewinnen. Unsere "Anlage" ist zwar mechanisch simpel, rechentechnisch, und was die Kommunikationsprotokolle angeht, aber identisch mit einer realen, sehr modernen Industrieanlage.

Lesen Sie bitte genau die Beschreibung im Portal :
https://portal.ts-muenchen.de/Portaldateien/Modellfabrik/Funktion.pdf

Die Anlage ist mechanisch nicht für Programmierübungen auf SPS-Ebene ausgelegt, sie kann durch fehlerhafte SPS-Programme leicht beschädigt werden. Also hier bitte keine Experimente !

Wir fangen zum Aufwärmen mit ein paar Wiederholungen zur Softwareentwicklung an, dann beschäftigen wir uns mit der Objektorientierung in der Industrieanwendung (auf MES-Systemen) und ein wenig (in Hinblick auf die agentenbasierten Systeme) auf der SPS.

Python

Lange waren die klassischen Programmiersprachen aus der IT (C++, Java) dominierend in der Industrietechnik.

Mit der radikalen Modernisierung der SPS-Technik, zum Teil auch durch den Einzug der Microcontroller in die Prozessebene (IoT), werden modernere Programmiersprachen aktuell.

Python ist als offene Sprache, die auf praktisch allen Betriebssystemen und auch auf den meisten SPS läuft, eine der Hochsprachen, die in Zukunft relevant sein werden.

Python ist eine Skriptsprache (Interpretersprache), was vor Allem bedeutet, daß keine Variablendeklarationen nötig sind.

Python installieren, IDLE

Python laden und installieren :
https://www.python.org/downloads/

Mit der Installation von Python 3 wird in Windows auch eine sehr praktische IDE installiert : IDLE

Mit der rechten Maustaste auf ein File, das .py als Endung hat, öffnet man IDLE.

Man kann auch mit großen Tools arbeiten, Visual Studio zum Beispiel, aber das hat kaum Vorteile. Aus der großen Zahl möglicher Editoren kann sich jeder aussuchen, was ihm gefällt …

Programmieren

Wie besprochen, machen wir ein etwas größeres Faß auf als unbedingt nötig, um neben den Themen Python und MES nochmal Gelegenheit zu bieten sich allgemein ein wenig mehr mit dem Programmieren anzufreunden.

Wir machen das in 4 Kapiteln :

1) Ein/Ausgabe, einfache Befehle
2) Verzweigungen (“if-Struktur”)
3) Schleifen (“while-Struktur”)
4) Was brauche ich : “if” oder “while” ??

Ein/Ausgabe, einfache Befehle

```
a = b + 1 ;
```

Rechnen : a = b + 1
a = a * 10
a = b / 10

Die Zuweisung des Datentyps erfolgt “implizit”, das heißt quasi im Moment der ersten Benutzung :

a = 12 : dieses a ist eine ganze Zahl (Integer)
a = 1.342 : dieses a ist eine Kommazahl (Float)
a = “oho” : dieses a ist eine Zeichenfolge (String)

Ein/Ausgabe :

```
a = input("Geben Sie eine Zahl ein")
print("Eingabe war : ", a)
```

Hier ist zu beachten, daß die "input" Anweisung in Python immer einen String liefert ! So ergibt z.b. diese Befehlsfolge :

```
a = input("Geben Sie eine Zahl ein")
a = a * 2
print("ergibt : ", a)
```

die Ausgabe : 22 (zweimal der String 2)

Wenn man eine Eingabe als Zahl weiterverarbeiten möchte, muß man das vorher konvertieren :

```
a = input("Geben Sie eine Zahl ein")
a = int(a)
a = a * 2
print("ergibt : ", a)
```

jetzt kommt 4 raus !

Übung 1

---

1. Aufgabe :

```
z1 = input("zeichen 1 : ")
z2 = input("zeichen 2 : ")
z = z1 + z2
print("zusammen : ",z)
```

2. Aufgabe :

```
z = input("zahl : ")
z = int(z)
z2 = z * 2
print("doppelt : ",z2)
z3 = z2 / 3
print("durch 3 : ",z3)
```

3. Aufgabe :

```
import math
z = input("zahl : ")
z = int(z)
sq = math.sqrt(z)
print("wurzel davon : ",sq)
```

4. Aufgabe :

```
import math
phi = input("zahl : ")
phi = float(phi)
rad = phi * math.pi / 180
sinus = math.sin(rad)
print("wert in bogenmaß : ",rad," sinus davon : ",sinus)
```