

Informatikprojekt – Aufgabenstellung

Dateischlüssel anhand von Messdaten auslesen



Zu identifizierender Parameter: Versuchsart

Szenario

- Es wurden diverse Versuchsreihen zur Prüfung des Sicherungsverhaltens von Schraubverbindungen durchgeführt.
- Die Prüfungen wurden nach einem festgelegten Verfahren durchgeführt, welches im Anhang erklärt wird.
- Die Messdaten wurden als CSV-Dateien abgespeichert. In den Dateinamen sind in verschlüsselter Form diverse Versuchsparameter abgespeichert. Der Dateiname ist immer gleich aufgebaut, d. h. die einzelnen Versuchsparameter sind in den Ihnen vorliegenden Daten immer an der selben Stelle im Dateinamen gespeichert.

Szenario – Dateischlüssel

Aufbau eines Dateinamens:

“xx_IP23_AA_BB_CC_DD_EE_FF_GG_HH_II_JJ_KK_LL_MM.csv“

- Die rot markierten Zeichen beziehen sich auf Ihre Gruppennummer und haben nichts mit den Versuchsparametern zu tun.
- Jeder im Dateinamen verschlüsselte Versuchsparameter wird durch 2 Zeichen dargestellt, im obigen Beispiel mit den Platzhaltern "AA" bis "MM" symbolisiert.

Szenario – Dateischlüssel – enthaltene Versuchsparmeter

Im Dateinamen sind folgende Parameter verschlüsselt:

- Festigkeitsklasse / Material von Schraube und Mutter
- Klemmlängenverhältnis
- Nenngroße
- Oberflächenbeschichtung
- Ordner-Nr. der Versuchsreihe
- Prüffrequenz
- Schmierung
- Sicherungselement
- Verschiebeweg
- Versuchsart
- Versuchs-Nr. innerhalb einer Versuchsreihe
- Versuchsteil
- Vorspannungskraft-Ausnutzung

Hinweise zur Verschlüsselung:

- Links sind alle im Dateischlüssel enthaltenen Parameter aufgelistet.
- Die Liste ist **alphabetisch sortiert**, was allerdings **nicht** der tatsächlichen Abfolge im Dateinamen entspricht.
- Einige der Parameter sind bei allen Messdateien konstant, andere variieren.
- Teilweise besteht ein Zusammenhang zwischen den einzelnen Versuchsparmetern.

Szenario – Dateischlüssel – Voraussetzung für Übersetzung

Zur „Übersetzung“ des Dateinamens in die tatsächlichen Versuchsp Parameter müsste eine Zuordnungsvorschrift vorliegen, sodass alle Zeichen im Dateinamen basierend auf ihren Positionen und ihren Ausprägungen den tatsächlichen Versuchsp Parametern zugeordnet werden können. Ein beispielhafter Ausschnitt aus einer solchen Zuordnungsvorschrift könnte wie folgt aussehen:

Platzhalter	Indexposition im Dateinamen	Verschlüsselter Versuchsp Parameter	Ausprägungen des Schlüssels (Symbole)	Bedeutungen dieser Ausprägungen (also dieser Symbole)
„AA“	8, 9	Nenng röße	L2	Nenng röße M4
			iO	Nenng röße M5
			9z	Nenng röße M6
„BB“	11, 12	Versuchsart	gT	Einstellversuch
			M9	Nachweisversuch
...

Die konkrete Ausprägung eines Schlüssels kann aus Zahlen, Buchstaben oder einer Kombination daraus bestehen.

Aufgabenstellung: Python-Programm entwickeln

Hauptziele des Programms:

- Identifizieren Sie anhand der zur Verfügung gestellten Datensätze die **Position** des Dateischlüssels für die Verschlüsselung des Parameters **Versuchsart** in den Dateinamen.
- Für **diesen Parameter** sollen die Bedeutungen aller in den Beispiel-Datensätzen auftretenden **Ausprägungen des Schlüssels** identifiziert werden. Gesucht ist also die Zuordnung, welche Symbol-Kombination für welchen Versuchsteil charakteristisch ist.

Aufgabenstellung: Python-Programm entwickeln

Weitere Anforderungen an das Programm:

- Die wesentlichen Ergebnisse (**Beantwortung der Ziele**) sollen dem Nutzer nach dem Programmdurchlauf in Form einer **automatisiert generierten Excel-Datei** vorliegen. Dabei soll in einem Tabellenblatt eine Auflistung der Rohdateinamen mit Zuordnung zu der Ausprägung des zu ermittelnden Versuchsparameters vorliegen. In der Datei soll ein **weiteres Tabellenblatt** als Legende dienen, aus der hervorgeht:
 - der ermittelte Versuchsparameter
 - die Indexposition der Verschlüsselung dieses Parameters im Dateinamen
 - die Bedeutung der identifizierten Ausprägungen

Allgemeine Vorgaben zur Programmerstellung

- Halten Sie Ihr Programm so **allgemein wie möglich** zur Verwendung auf andere Dateien mit ähnlichem Aufbau
- Fehler sollen abgefangen werden (Fehlervermeidung)
- Der Algorithmus bzw. die **Teilalgorithmen** sind sinnvoll in **Funktionen** umzusetzen.
- Über eine „Hauptfunktion“ sollen der Programmablauf gewährleistet und die dafür notwendigen Funktionen aufgerufen werden.
- Bewertet werden unter anderem Aufbau und Übersichtlichkeit der Funktionen.
- **Dokumentationen im Code** (Kommentare, Docstrings bzw. Programmkopf) müssen klar und verständlich sein.
- Achten Sie auf erkennbare Benennungen von Prozeduren, Variablen, etc. (tipps_python_programmierung.html)
- Ergebnisse sollen soweit möglich **numerisch und graphisch** ausgegeben werden.
- Die Ergebnisausgabe muss nachvollziehbar und reproduzierbar sein.
- Die Rohdaten sollen vor dem automatisierten Auswerten nicht manuell verändert werden.
- Ergebnisse sollen im **Kurzbericht** kritisch bewertet werden. Wie aussagekräftig sind sie?

Hilfreiche Zusatzmodule (1/2)

Sie können beliebige Zusatzmodule verwenden. Eine nach Funktionalitäten gegliederte, sehr umfangreiche Liste von Zusatzmodulen und Libraries finden Sie beispielsweise unter <https://awesome-python.com>. Folgend eine Auflistung von gängigen Modulen für verschiedene Aufgabenbereiche:

- os (Verschiedene Betriebssystemschnittstellen, z. B. Ordneraktionen)
- sys (Systemspezifische Parameter & Funktionen)
- time (Zeitzugriff und Umrechnungen)

- numpy (Grundlegendes Paket für effizientes wissenschaftliches Rechnen)
- pandas (Leistungsstarke Datenstrukturen, Datenanalysewerkzeuge)

- seaborn (Datenvisualisierungen, v.a. auch für Statistik)
- matplotlib (Grundlegendes Paket zur 2D- und 3D-Visualisierung von Daten)
- folium ((Interaktive) Visualisierung von Satellitenkarten/Straßenkarten)

Hilfreiche Zusatzmodule (2/2)

Sie können beliebige Zusatzmodule verwenden. Eine nach Funktionalitäten gegliederte, sehr umfangreiche Liste von Zusatzmodulen und Libraries finden Sie beispielsweise unter <https://awesome-python.com>. Folgend eine Auflistung von gängigen Modulen für verschiedene Aufgabenbereiche:

- SciPy (Umfangreiche Tools für Mathematik, Wissenschaft, Technik)
- SymPy (Symbolisches Rechnen)
- tkinter (Grundlegendes Modul zur Erstellung von graphischen Bedienoberflächen)
- csv (Lesen & Schreiben von CSV-Dateien)
- openpyxl (Lesen & Schreiben von Excel-Dateien)
- python-docx (Lesen & Schreiben von Word-Dateien)
- python-pptx (Lesen & Schreiben von PowerPoint-Dateien)
- PyPDF2 (Aufteilen, Zusammenführen und Umwandeln von PDF-Seiten)

Abgabe als ein einziges ZIP-Archiv

Dateiname des ZIP-Archives: **IP_SS23_XX.zip** [XX steht für die Team Nr.]

Inhalt des ZIP-Archives:

- **Bericht** in Präsentationsform als PDF-Datei. Dateiname: **IP_SS23_XX_Bericht.pdf**
- **Programm** als py-Datei. Dateiname: **IP_SS23_XX_Programm.py**
- **Programmablaufplan** als pdf-Datei (Allgemeinverständlich formuliert ohne Quellcode). Dateiname: **IP_SS23_XX_PAP.pdf** [optional/zusätzlich: Programmablaufplan als pap-Datei]
- **Excel-Auswertungsdatei** : Dateiname: **IP_SS23_XX_Ergebnis.xlsx**

Bericht in Präsentationsform: max. 12-17 Folien:

- Beteiligte Teammitglieder mit Aufgabenbereich
- Kurzerläuterung Problemstellung (max. 2 Folien)
- Kurzbeschreibung der Herangehensweise (Algorithmen bzw. Teilalgorithmen und verwendete Kriterien erklären)
- Ergebnisdarstellung und (kritische) Bewertung
- Zeigen Sie einen mit Python generierten, exemplarischen Plot über den Verlauf der Vorspannkraft einer beliebigen Datei

Aufgabenstellung: Python-Programm entwickeln

Tipps für den Beginn der Arbeit (1/3):

- Verschaffen Sie sich einen Überblick über die Verläufe der relevanten Informationen in Ihren Datensätzen.
- Für wichtige Hinweise hinsichtlich des Ablaufes der Versuche siehe separate Datei bzgl. Erklärung des Schraubenprüfstandes.

Aufgabenstellung: Python-Programm entwickeln

Tipps für den Beginn der Arbeit (2/3):

- Zur Programmentwicklung (Ausprobieren) genügt es oftmals Ausschnitte der Rohdaten zu verwenden. Dadurch reduziert sich die Rechenzeit erheblich. Für das endgültige Ergebnis sollen alle Daten verwendet werden.
- Um Ergebnisse manuell zu sichten, können interaktive Diagramme hilfreich sein. In der IDE Spyder ist das z.B. für Plots mit dem Zusatzmodul matplotlib möglich, indem die Plots statt inline in einem separaten Fenster angezeigt werden. Um das einzustellen, kann im Programmcode (oder einmal in der Konsole) der magic command „%matplotlib qt“ verwendet werden (ohne Anführungszeichen). Die Ausgabe innerhalb des Plot-Explorers von Spyder kann mit dem Befehl „%matplotlib inline“ aktiviert werden.

Aufgabenstellung: Python-Programm entwickeln

Beispielhafte Ergebnis-Datei:

Rechts: Tabellenblatt „Legende“

Unten: Tabellenblatt „Zuordnung“

	A	B
1	Gruppe 99 - Zusammenfassung:	
2		
3	Zuzuordnender Versuchsparameter:	Versuchsteil
4	Key-Platzhalter (Buchstaben):	KK
5	Positionen des Keys im Dateinamen:	[38, 39]
6		
7		
8		
9	Legende für die verschiedenen Ausprägungen dieses Keys:	
10		
11	Ausprägung	Bedeutung
12	AV	Anziehversuch
13	RV	Vibrationsversuch
14	LV	Losdrehversuch

Nr.	Dateiname	AA	BB	CC	DD	EE	FF	GG	HH	II	JJ	KK	LL	MM	Zugeordneter_Versuchsteil
0	YY_IP21_08_--_A2_17_50_ev_40_a2_12_EV_RV_01_02.csv	08	--	A2	17	50	ev	40	a2	12	EV	RV	01	02	Vibrationsversuch
1	YY_IP21_08_--_A2_17_50_ev_40_a2_12_EV_RV_01_03.csv	08	--	A2	17	50	ev	40	a2	12	EV	RV	01	03	Vibrationsversuch
2	YY_IP21_08_SP_A2_17_50_ev_40_a2_12_NV_RV_01_04.csv	08	SP	A2	17	50	ev	40	a2	12	NV	RV	01	04	Vibrationsversuch
3	YY_IP21_08_SP_A2_17_50_ev_40_a2_12_NV_RV_01_06.csv	08	SP	A2	17	50	ev	40	a2	12	NV	RV	01	06	Vibrationsversuch
4	YY_IP21_10_--_A2_17_50_ev_10_a2_12_EV_RV_01_04.csv	10	--	A2	17	50	ev	10	a2	12	EV	RV	01	04	Vibrationsversuch
5	YY_IP21_10_--_A2_17_50_ev_10_a2_12_EV_RV_01_05.csv	10	--	A2	17	50	ev	10	a2	12	EV	RV	01	05	Vibrationsversuch
6	YY_IP21_10_SP_A2_17_50_ev_10_a2_12_NV_RV_01_02.csv	10	SP	A2	17	50	ev	10	a2	12	NV	RV	01	02	Vibrationsversuch
7	YY_IP21_10_SP_A2_17_50_ev_10_a2_12_NV_RV_01_06.csv	10	SP	A2	17	50	ev	10	a2	12	NV	RV	01	06	Vibrationsversuch
8	YY_IP21_12_--_88_17_50_ev_40_ge_12_EV_RV_01_06.csv	12	--	88	17	50	ev	40	ge	12	EV	RV	01	06	Vibrationsversuch
9	YY_IP21_12_--_88_17_50_ev_50_ge_12_EV_AV_01_06.csv	12	--	88	17	50	ev	50	ge	12	EV	AV	01	06	Anziehversuch
10	YY_IP21_12_--_88_17_50_ev_50_ge_12_EV_AV_01_09.csv	12	--	88	17	50	ev	50	ge	12	EV	AV	01	09	Anziehversuch

11.05.23

Link zu den Rohdaten

Die Rohdaten zu Ihrer Aufgabe finden Sie unter folgendem Link:

<https://th-koeln.sciebo.de/s/t6xOf4muu0JAGjF>