LU11.A03 - Mocks mit unittest.mock (Zusatz)

Testen Sie eine Funktion, die E-Mails versendet und auf eine Datenbank zugreift,
ohne dass Mailserver oder Datenbank vorhanden sind.
Sozialform: Einzelarbeit · **Zusatzaufgabe** (Advanced), keine Abgabe

Theorie: LU11b - Mock

Ausgangslage

Die Schule will Lernende mit mehr als 3 Absenzen automatisch per E-Mail benachrichtigen. Erstellen Sie in PyCharm ein neues Projekt mit diesen zwei Dateien:

school_data.py

```
""" Zugriff auf die Schuldatenbank (im Test nicht verfügbar) """


def load_students():
    """
    Lädt alle Lernenden mit ihren Absenzen aus der Datenbank
    :return: Liste von Dictionaries mit name, email und absences
    """
    raise ConnectionError('Keine Verbindung zur Schuldatenbank')
```

absences.py

```
""" Benachrichtigt Lernende mit zu vielen Absenzen """
import school_data

MAX_ABSENCES = 3


def notify_absences(students, mailer):
    """
    Sendet allen Lernenden mit mehr als MAX_ABSENCES Absenzen eine E-
Mail
    :param students: Liste von Dictionaries mit name, email und
absences
    :param mailer: Objekt mit der Methode send(to, subject, text)
    :return: Anzahl versendeter E-Mails
    """
    count = 0
    for student in students:
        if student['absences'] > MAX_ABSENCES:
```

```
            mailer.send(
                student['email'],
                'Absenzen',
                f'Guten Tag {student["name"]}, Sie haben
{student["absences"]} Absenzen.'
            )
            count += 1
    return count


def daily_check(mailer):
    """
    Lädt die Lernenden aus der Datenbank und versendet die
Benachrichtigungen
    :param mailer: Objekt mit der Methode send(to, subject, text)
    :return: Anzahl versendeter E-Mails, -1 wenn die Datenbank nicht
erreichbar ist
    """
    try:
        students = school_data.load_students()
    except ConnectionError:
        return -1
    return notify_absences(students, mailer)
```

Auftrag

Schreiben Sie in `test_absences.py` die folgenden Tests. Verwenden Sie `Mock` für den Mailer und `patch` für `load_students`.

Nr	Test	Prüfen Sie
1	Von zwei Lernenden hat eine Person 4 Absenzen, die andere 3.	Rückgabewert 1; `send` wurde **genau einmal** mit Adresse, Betreff und Text der richtigen Person aufgerufen.
2	Eine Person hat genau 3 Absenzen (Grenzwert).	Rückgabewert `0`; `send` wurde **nicht** aufgerufen.
3	Zwei Personen haben zu viele Absenzen.	`send` wurde zweimal aufgerufen, der zweite Aufruf ging an die zweite Person.
4	`daily_check` mit einer Datenbank, die zwei Lernende liefert	`load_students` wird mit `patch` durch einen Stub ersetzt. Rückgabewert stimmt.
5	`daily_check`, wenn die Datenbank nicht erreichbar ist	`load_students` löst mit `side_effect` einen `ConnectionError` aus. Rückgabewert `-1`; keine E-Mail versendet.

Den Mailer erstellen Sie am besten in einer Fixture. Achten Sie bei `patch` auf den richtigen Pfad: `absences.py` importiert das Modul `school_data`.

M450-LU11, M450-C1A

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu11/aufgaben/mock

Last update: **2026/11/23 09:26**