LU11.L03 - Mocks mit unittest.mock

Musterlösung zu LU11.A03 - Mocks mit unittest.mock.

test_absences.py

```
""" Unit Tests für absences mit unittest.mock """
from unittest.mock import Mock, patch

import pytest

from absences import notify_absences, daily_check

PIA = {'name': 'Pia', 'email': 'pia@example.test', 'absences': 4}
MAX = {'name': 'Max', 'email': 'max@example.test', 'absences': 3}


@pytest.fixture(name='mailer')
def fixture_mailer():
    """ Ein Mock anstelle des echten Mailservers """
    return Mock()


def test_notify_one_student(mailer):
    """ Nur Pia hat mehr als 3 Absenzen """
    assert notify_absences([PIA, MAX], mailer) == 1
    mailer.send.assert_called_once_with(
        'pia@example.test', 'Absenzen', 'Guten Tag Pia, Sie haben 4
Absenzen.'
    )


def test_notify_limit(mailer):
    """ Grenzwert: genau 3 Absenzen lösen keine E-Mail aus """
    assert notify_absences([MAX], mailer) == 0
    mailer.send.assert_not_called()


def test_notify_several(mailer):
    """ Mehrere Lernende erhalten je eine E-Mail """
    other = {'name': 'Cem', 'email': 'cem@example.test', 'absences': 7}
    assert notify_absences([PIA, other], mailer) == 2
    assert mailer.send.call_count == 2
    assert mailer.send.call_args_list[1].args[0] == 'cem@example.test'


def test_daily_check(mailer):
    """ load_students wird durch einen Stub mit festen Daten ersetzt
```

```
"""
    with patch('absences.school_data.load_students', return_value=[PIA,
MAX]):
        assert daily_check(mailer) == 1


def test_daily_check_no_database(mailer):
    """ Datenbankfehler mit side_effect simulieren """
    with patch('absences.school_data.load_students',
side_effect=ConnectionError):
        assert daily_check(mailer) == -1
    mailer.send.assert_not_called()
```

Ergebnis

```
$ pytest -q
.....
5 passed

$ pylint --max-line-length=120 absences.py school_data.py test_absences.py
Your code has been rated at 10.00/10
```

Erläuterungen

- Der Mailer ist ein **Mock**: Geprüft wird nicht ein Rückgabewert, sondern ob `send` richtig aufgerufen wurde.
- `load_students` wird durch einen **Stub** ersetzt: `return_value` liefert feste Testdaten, `side_effect` simuliert den Ausfall der Datenbank.
- Weil `absences.py` das Modul mit `import school_data` einbindet, funktioniert der Pfad `absences.school_data.load_students` ebenso wie `school_data.load_students`. Hätte `absences.py` `from school_data import load_students` verwendet, müsste der Pfad `absences.load_students` lauten.

M450-LU11, M450-C1A

Marcel Suter

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

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

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