LU04.L02 - Skip und xfail

Lösung zu LU04.A02 - Skip und xfail. Der Code stammt aus dem Branch `solution` des Repositorys m450-lu04-a02-calculator.

test_main.py

test_main.py

```python
"""Unit Tests für den Taschenrechner."""
import sys

import pytest

from calculator import divide


def test_success():
    """Zwei Zahlen werden korrekt dividiert."""
    result = divide(6, 4)
    assert result == pytest.approx(1.5)


@pytest.mark.xfail(raises=TypeError, reason='divide prüft die Eingaben nicht')
def test_invalid():
    """Buchstaben statt Zahlen."""
    divide('a', 2)


def test_zero_division():
    """Division durch 0 löst einen ZeroDivisionError aus."""
    with pytest.raises(ZeroDivisionError):
        divide(6, 0)


@pytest.mark.skip(reason='Dieser Test wird absichtlich übersprungen')
def test_mark_skip():
    """Wird immer übersprungen."""
    assert False


@pytest.mark.skipif(sys.version_info > (3, 8), reason='nur für Python 3.8 und älter')
def test_mark_skip_conditional():
    """Wird ab Python 3.9 übersprungen."""
```

```
    assert False


def test_pytest_skip():
    """Wird innerhalb der Funktion übersprungen."""
    if True:  # pylint: disable=using-constant-test
        pytest.skip('Die Bedingung ist erfüllt, der Test wird
übersprungen')
    assert False
```

Ausgabe

```
$ pytest -v -rsxX
test_main.py::test_success PASSED
test_main.py::test_invalid XFAIL (divide prüft die Eingaben nicht)
test_main.py::test_zero_division PASSED
test_main.py::test_mark_skip SKIPPED (Dieser Test wird absichtlich
übersprungen)
test_main.py::test_mark_skip_conditional SKIPPED (nur für Python 3.8 und
älter)
test_main.py::test_pytest_skip SKIPPED (Die Bedingung ist erfüllt, der Test
wird übersprungen)
==================== 2 passed, 3 skipped, 1 xfailed ====================
```

Erläuterungen

- **test_success:** Bei Fliesskommazahlen immer `pytest.approx` verwenden.
- **test_invalid:** `divide('a', 2)` löst einen `TypeError` aus, weil `divide` die Eingaben nicht prüft. Mit `raises=TypeError` gilt der Test nur dann als erwarteter Fehlschlag, wenn genau diese Exception auftritt. Eine andere Exception macht den Test rot.
- **test_zero_division:** Hier ist die Exception das **korrekte** Verhalten. Deshalb `pytest.raises` und nicht `xfail` (siehe LU04d).
- **test_mark_skip_conditional:** Mit einer aktuellen Python-Version ist die Bedingung erfüllt und der Test wird übersprungen.
- **test_pytest_skip:** `pytest.skip()` bricht den Test sofort ab. Das `assert False` danach wird nie ausgeführt.

M450-LU04

Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu04/loesungen/skipif?rev=1794900590

Last update: **2026/11/17 08:29**