[CS 11 25.1] Lab 7g – Safe Banking

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Problem Statement

You are trying to establish your own bank (for some reason), and you need it to be very safe!

Task Details

Your task is to implement a **dataclass** named `Account`:

- It should have two attributes: `account_id` (a `str`) and `balance` (an `int`).
- It should have two methods: `withdraw` and `deposit`.

Both methods should not change the `balance` value and should raise a `ValueError` if either of the following happens:

- A negative amount is withdrawn or deposited.
- The balance would become negative after (hypothetically) performing the withdrawal or deposit.

Name your file `lab07g.py` and your testing file `test_lab07g.py`.

Restrictions

- The following symbols are allowed: `iter`, `next`, `map`, `filter`
- The following imports are allowed:
  - `count`, `islice`, `chain`, `takewhile`, `starmap` and `zip_longest` from `itertools`.
  - `cache`, `lru_cache`, `total_ordering`, `partial`, `reduce` and `wraps` from `functools`.
  - `randint`, `randrange and choice` from `random`.
  - `Fraction` from `fractions`.
  - `dataclass` from `dataclasses`.
  - `contextmanager` from `contextlib`.
  - `Enum`, `auto` from `enum`.
- Anonymous functions are allowed.
- Inner functions are allowed.
- Classes, dataclasses, and enums are allowed.
- Recursion is allowed.
- Loops are allowed.
- Generators and comprehensions are allowed.
- Your source code must have at most 800 bytes.

Example Testing

Here's an example testing file. **Important:** You need to run `pytest`, otherwise, the tests won't run! Please fill in the TODO as well.

Copy

```
# pyright: strict

from lab07g import Account

def test_Account():
    account = Account(account_id="JJV", balance=0)

    account.deposit(11)
    assert account.balance == 11
    account.withdraw(10)
    assert account.balance == 1

    try:
        account.withdraw(-2)
    except ValueError:
        print("This should be printed out")

    try:
        account.withdraw(2)
    except ValueError:
        print("This should also be printed out")

    # TODO add more tests here
```

Constraints

- The `withdraw` and `deposit` methods will be called a total of at most $350{,}000$ times.
- The initial balance is between 0 and 10^{20}.
- Each argument to `withdraw` or `deposit` is between -10^{20} and 10^{20}.
- Each name is a string of between 0 and 5 English letters.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 130 ❤️ points if you solve all test cases where:
  - `ValueError` will never need to be raised.
- You get 120 🔴 points if you solve all test cases.

Submit solution [CS 11 25.1] Lab Exercise 7

My submissions

Points: 250 (partial)
Time limit: 6.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Clarifications

Report an issue

No clarifications have been made at this time.