TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP MÔN HỌC

Kiểm thử và đảm bảo chất lượng phần mềm

BÀI TẬP: KIỂM THỬ DÒNG ĐIỀU KHIỂN

Sinh viên: Đỗ Hữu Hoàng Tùng

Mã sinh viên: 22026551

1. Đặc tả bài toán

Chương trình mô phỏng đơn giản quá trình giao dịch trong ngân hàng, nhận 3 thành phần chính là đầu vào:

1. balance: Số tiền hiện tại có trong tài khoản
2. transaction_type: Loại giao dịch, có thể nhận 1 trong 2 giá trị:
    a. deposit: Giao dịch bỏ tiền vào tài khoản
    b. withdrawal: Giao dịch rút tiền khỏi tài khoản
3. amount: Số tiền sử dụng vào giao dịch

Tùy vào loại giao dịch, số balance trong tài khoản sẽ được thêm vào hoặc bớt đi 1 lượng tương ứng với amount. Chương trình có 1 số ràng buộc nhất định như sau:

1. amount và balance đều phải là 1 số tự nhiên
2. transaction_type chỉ có thể nhận 1 trong 2 giá trị là deposit và withdrawal
3. Do vấn đề an ninh, chỉ cho phép số lượng amount tối đa trong mỗi giao dịch là 1 triệu
4. Nếu transaction_type là withdrawal thì amount không được lớn hơn balance

2. Cài đặt chương trình

Sử dụng ngôn ngữ Python để viết hàm cho chương trình

```
def bank_transaction(balance: int, transaction_type:str, amount:int) ->
int:
    """
    Perform a bank transaction (deposit or withdrawal) on the current
amount.

    Parameters:
    balance (int): The current amount in the bank account.
    transaction_type (str): The type of transaction ('deposit' or
'withdrawal').
    amount (int): The amount to deposit or withdraw.

    Returns:
    int: The new amount in the bank account after the transaction.
```

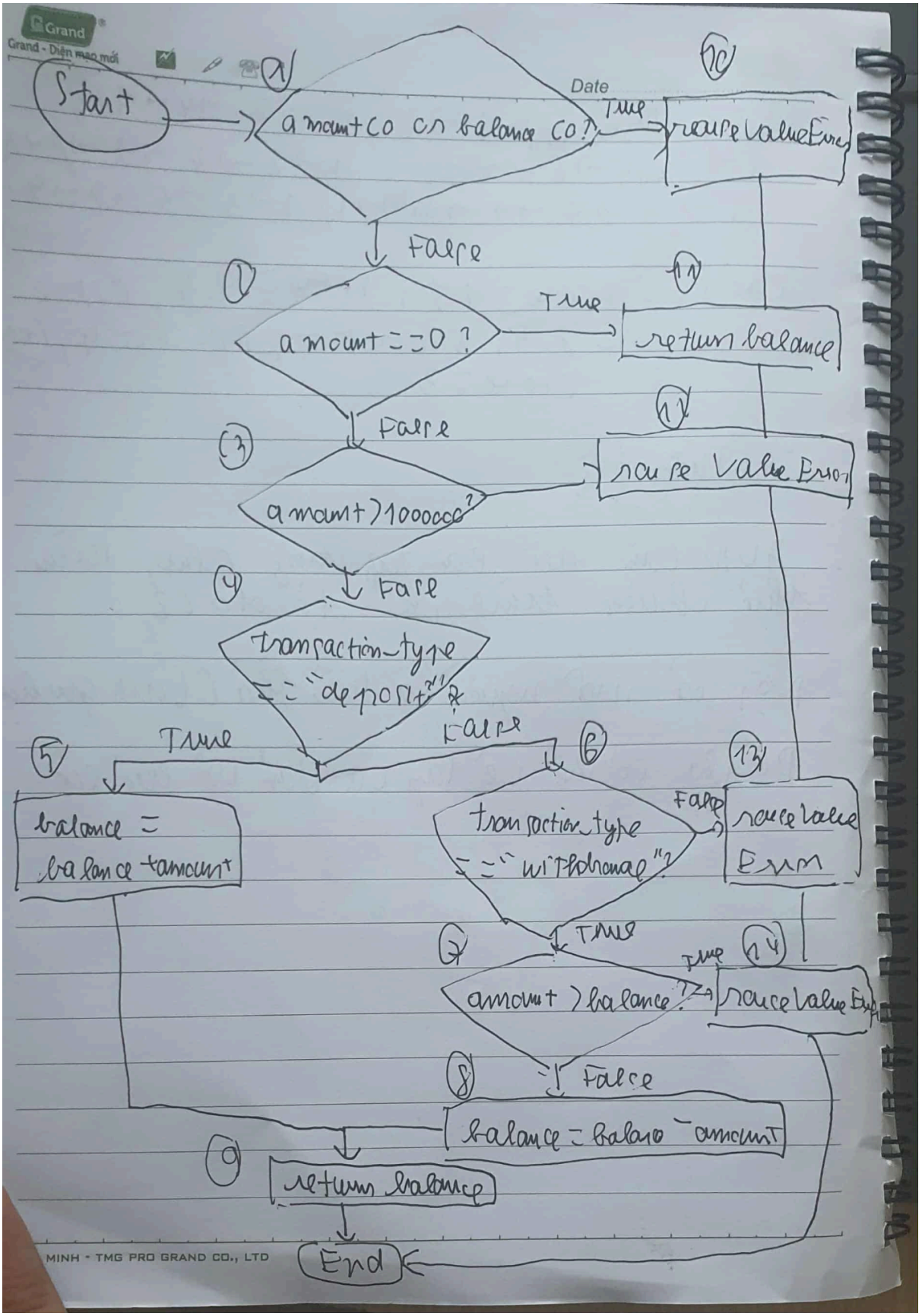
```
    """
    if amount < 0 or balance < 0:
        raise ValueError("Input not valid.")
    if amount == 0:
        return balance
    if amount > 1000000:
        raise ValueError("Input not valid.")
    if transaction_type == 'deposit':
        balance = balance + amount
    elif transaction_type == 'withdrawal':
        if amount > balance:
            raise ValueError("Input not valid.")
        balance = balance - amount
    else:
        raise ValueError("Input not valid.")
    return balance
```

1. Đoạn code chương trình giao dịch ngân hàng

3. Kiểm thử dòng dữ liệu

3.1 Đồ thị dòng điều khiển

Dựa vào mã nguồn trên, ta vẽ được đồ thị dòng điều khiển như sau:



Hình 1. Đồ thị dòng điều khiển của chương trình

Chương trình gồm có 3 biến là amount, balance và transaction_type, dựa vào đồ thị ta xác định được def, p-use và c-use của các biến:

- Biến amount: def(amount) = None, p-use(amount) = {1, 2, 3, 7}, c-use(amount) = {5, 8}
- Biến balance: def(balance) = {5, 8}, p-use(balance) = {1, 7}, c-use(balance) = {11, 5, 8, 9}
- Biến transaction_type: def(transaction_type) = None, p-use(transaction_type) = {4, 6}, c-use(transaction_type) = None

Như vậy, do trong chương trình không tồn tại câu lệnh def của 2 biến amount và transaction type, với độ đo c-uses ta chỉ có thể xây dựng được các con đường của biến balance như sau:

Biến	Du-pair	Def-clear path	Complete path
Balance	(5, 9)	5 -> 9	1(F) -> 2(F) -> 3(F) -> 4(T) -> 5 -> 9
	(8, 9)	8 -> 9	1(F) -> 2(F) -> 3(F) -> 4(F) -> 6(T) -> 7(F) -> 8 -> 9

Bảng 1. Phân tích đường đi theo độ phủ all-uses

Từ đó, ta sinh ra được 2 ca kiểm thử

STT	Input	Expected Output
1	balance: 1000 transaction_type: deposit amount: 5000	6000
2	balance: 1000 transaction_type: withdrawal amount: 500	500

Bảng 2. Bảng kiểm thử theo dòng dữ liệu

```
class TestBankTransactionDataFlowAllUsesMethod(unittest.TestCase):
    def test_1(self):
        self.assertEqual(bank_transaction(1000, 'deposit', 5000), 6000)

    def test_2(self):
        self.assertEqual(bank_transaction(1000, 'withdrawal', 500),
500)
```

2. Đoạn code xây dựng các ca kiểm thử theo dòng dữ liệu

3.3 Kết quả kiểm thử

Sử dụng công cụ kiểm thử được tích hợp sẵn Pyunit của Python để tạo và chạy các ca kiểm thử vừa được sinh ra

```
..
----------------------------------------------------------------------
Ran 2 tests in 0.000s

OK
```

Hình 2. Kết quả kiểm thử

STT	Input	Expected Output	Actual Output	Status
1	balance: 1000 transaction_type: deposit amount: 5000	6000	6000	Passed
2	balance: 1000 transaction_type: withdrawal amount: 500	500	500	Passed

Bảng 2. Bảng báo cáo kiểm thử theo dòng dữ liệu

4. Nhận xét và kết luận

Từ kết quả kiểm thử trên, có thể thấy rằng với phương pháp kiểm thử theo dòng dữ liệu theo tiêu chí all_uses, chương trình đã vượt qua

toàn bộ các ca kiểm thử. Tuy nhiên, chỉ có biến amount có thể áp dụng được phương pháp này vì đây là biến duy nhất có khối lệnh def, và chỉ cần hai ca kiểm thử là đã đạt 100% độ phủ. Điều này cho thấy phương pháp kiểm thử theo dòng dữ liệu có thể chưa thực sự phù hợp với chương trình này.

Link GitHub: dhht0810/KTPM