# How you model and analysis the problem

All 4 accounts under the bank class, bank class contains general functions (deposit and withdraw) and initials(name, ac\_num, bal).

Assumptions:

Assume all accounts cannot deposit or withdraw negative value and non number value.

Current account: The number of cheque will increase when the user withdraw from this account, at the beginning the number of cheque issued is 0

Deposit account: The number of withdrawal cannot exceed the predefined maximum number, in this program the maximum number of withdrawal is 3. For fixed interest rate, assume every deposit keep in the account for a month. The defined interest rate is 2% for a year and the interest calculate per month for every deposit and will add to the balance.

Restricted account: The restricted account is a subclass of current account, every withdrawal will also be count and increase the number of cheque count in this account, the withdraw amount cannot exceed the predefined value. In this program, the predefine value is 10000.

Overdue account: The overdue account is a subclass of current account, every withdrawal will also be count and increase the number of cheque count in this account, the balance of this account cannot smaller than the overdue limit. In this program, the limit is 100.

1. user enter the account number
2. system will check if it is a valid account number
3. if yes, the system will ask the user to proceed further action or not
4. if yes, the system will ask the user to choose deposit or withdrawal
5. then, the system will ask the user to input the amount to withdraw or deposit
6. the system will show the balance after process
7. the system will ask if the user wants further action
8. if no, the system will save all data into the file for next time use

# 1.2 UML class diagram

bank()

name: string

ac\_num: string

bal: float

deposit(): void

withdraw(): void

current()

name: string = Current account

ac\_num: string= 111

bal: float

cheque: int

withdraw\_from\_ac(): void

deposit()

name:string = Deposit account

ac\_num: string= 222

bal:float

counter: int

withdraw\_from\_ac(): void, count:int

deposit(): void, interest\_rate\_pa:float

restricted()

name:string = Restricted account

ac\_num: string= 333

bal: float

cheque:int

withdraw\_restricted(): void, max\_withdraw\_limit:int= 10000

overdue()

name: string= Account with Overdraft Facility

ac\_num:string = 444

bal:float

cheque: int

withdraw\_from\_ac(): void, max\_overdue\_limit:int =100

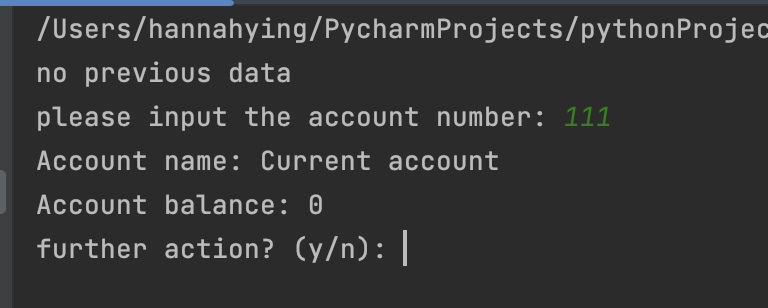
# 1.3 How you test your program

* Test Case group 1: Check results on entering valid data
* Test Case group 2: Check results on entering Invalid data or unexpected data
* Test Case group 3: Check response when enter empty data

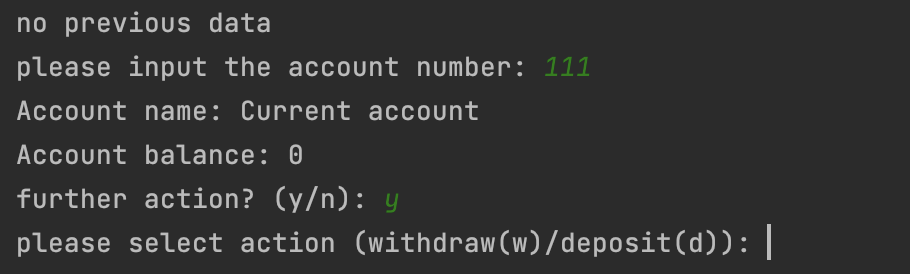
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** |
| **1.1** | **Input valid account number** | 1. **Run the program** 2. **Input account number** | **111/222/333/444** | 1. **Show corresponding account name** 2. **Show corresponding account number and account balance** 3. **Ask the user for further action** |
| **1.2** | **Further action: yes** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** | 1. **111/222/333/444** 2. **y** | **Ask the user to choose deposit or withdraw** |
| **1.3** | **Further action: no** | 1. **Run the program** 2. **Input valid account number** 3. **Type “n” in further action** | 1. **111/222/333/444** 2. **n** | **Print “finish”** |
| **1.4** | **Further input: deposit** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** | 1. **111/222/333/444** 2. **y** 3. **d** | **Ask the user to enter amount for deposit** |
| **1.5** | **Further input: withdraw** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** | 1. **111/222/333/444** 2. **y** 3. **w** | **Ask the user to enter amount for withdrawl** |
| **1.6** | **Current account: deposit** | 1. **Run the program** 2. **Input “111” account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Enter valid deposit amount** | 1. **111** 2. **y** 3. **d** 4. **500** 5. **n** | **Print “deposit to Current account new balance is 500”** |
| **1.7** | **Current account: withdraw** | 1. **Run the program** 2. **Input “111” account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Enter valid withdraw amount** | 1. **111** 2. **y** 3. **w** 4. **100** 5. **n** | **Show the no. of cheque issued, print “**  **withdraw to Current account new balance is 400"** |
| **1.8** | **Deposit account:deposit** | 1. **Run the program** 2. **Input “222” account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Enter valid deposit amount** | 1. **222** 2. **y** 3. **d** 4. **500** 5. **n** | **Show the interest per month**  **Print “deposit to Deposit account new balance is 500.83“** |
| **1.9** | **Deposit account:withdraw** | 1. **Run the program** 2. **Input “222” account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Enter valid withdraw amount** | 1. **222** 2. **y** 3. **w** 4. **100** 5. **n** | **Show the number of withdrawal**  **Print “withdraw to Deposit account new balance is 400.83"** |
| **1.10** | **Restricted account:deposit** | 1. **Run the program** 2. **Input “333” account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Enter valid deposit amount** | 1. **333** 2. **y** 3. **d** 4. **50000** | **Print “deposit to restriced account new balance is 50000.00”** |
| **1.11** | **Restricted account: withdraw** | 1. **Run the program** 2. **Input “333” account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Enter valid withdraw amount** | 1. **333** 2. **y** 3. **w** 4. **10000** 5. **n** | **Show the number of cheque issued, Print “withdraw to Restricted account new balance is 40000.00”** |
| **1.12** | **Overdue account: deposit** | 1. **Run the program** 2. **Input “444” account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Enter valid deposit amount** | 1. **444** 2. **y** 3. **d** 4. **500** 5. **n** | **Print “deposit to overdue account new balance is 500.00”** |
| **1.13** | **Overdue account: withdraw** | 1. **Run the program** 2. **Input “444” account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Enter valid withdraw amount** | 1. **444** 2. **y** 3. **w** 4. **600** | **Show the number of cheque issued, Print “withdraw to overdue account new balance is -100.00”** |
| **1.14** | **Store data** | 1. **Run the program** 2. **Input any valid account number** 3. **Type “y” in further action** 4. **Type d/w in further input** 5. **After a few deposit/ withdrawal** 6. **Type “n” in further action** | 1. **444** 2. **y** 3. **d** 4. **500** 5. **n** | **bank\_ac.dat will be updated** |
| **1.15** | **Load data** | 1. **Run the program** | **N.A.** | **The data in bank\_ac.dat will be loaded and printed out** |
| 2.1 | **Input invalid account number** | 1. **Run the program** 2. **Input invalid account number** | 1. **555** | **Print no such account** |
| 2.2 | **Further action: invalid** | 1. **Run the program** 2. **Input valid account number** 3. **Type invalid value in further action** | 1. **444** 2. **5** | **Print “invalid”** |
| 2.3 | **Further input: invalid** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type invalid in further input** | 1. **444** 2. **y** 3. **y** | **Print “invalid”** |
| 2.4 | **Current account: invalid deposit (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type negative value in the amount** | 1. **111** 2. **y** 3. **d** 4. **-10** | **Print “cannot deposit negative value”** |
| 2.5 | **Current account: invalid deposit (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type non integer value in the amount** | 1. **111** 2. **y** 3. **d** 4. **y** | **Print “invalid”** |
| 2.6 | **Current account: invalid withdraw (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type negative value in the amount** | 1. **111** 2. **y** 3. **w** 4. **-60** | **Print “cannot withdraw negative value”** |
| 2.7 | **Current account: invalid withdraw (larger than the balance)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type a larger value than the balance in the amount** | 1. **111** 2. **y** 3. **w** 4. **1000** | **Print “insufficient balance”** |
| 2.8 | **Current account: invalid withdraw (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type “w” in the amount** | 1. **111** 2. **y** 3. **w** 4. **w** | **Print “invalid”** |
| 2.9 | **Deposit account: invalid deposit (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type negative value in the amount** | 1. **222** 2. **Y** 3. **D** 4. **-10** | **Print “cannot deposit negative value”** |
| 2.10 | **Deposit account: invalid deposit (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type non integer value in the amount** | 1. **222** 2. **Y** 3. **D** 4. **d** | **Print “invalid”** |
| 2.11 | **Deposit account: invalid withdraw (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type negative value in the amount** | 1. **222** 2. **Y** 3. **W** 4. **-20** | **Print “cannot withdraw negative value”** |
| 2.12 | **Deposit account: invalid withdraw (larger than the balance)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type a larger value than the balance in the amount** | 1. **222** 2. **Y** 3. **W** 4. **501** | **Print “insufficient balance”** |
| 2.13 | **Deposit account: invalid withdraw (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type “w” in the amount** | 1. **222** 2. **Y** 3. **W** 4. **w** | **Print “invalid”** |
| 2.15 | **Deposit account: invalid deposit (exceed the number of withdrawal)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type valid amount** 6. **Repeat step 3-5 for 3 times** | 1. **222** 2. **Y** 3. **W** 4. **30** 5. **Y** 6. **W** 7. **10** 8. **Y** 9. **W** 10. **20** 11. **Y** 12. **W** 13. **10** | **Print "Exceed maximum number of time of withdrawal"** |
| 2.16 | **Restricted account: invalid deposit (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type negative value in the amount** | 1. **333** 2. **Y** 3. **D** 4. **-8** | **Print “cannot deposit negative value”** |
| 2.17 | **Restricted account: invalid deposit (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type non integer value in the amount** | 1. **333** 2. **Y** 3. **D** 4. **d** | **Print “invalid”** |
| 2.18 | **Restricted account: invalid withdraw (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type negative value in the amount** | 1. **333** 2. **Y** 3. **W** 4. **-100** | **Print “cannot withdraw negative value”** |
| 2.19 | **Restricted account: invalid withdraw (larger than the balance)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type a larger value than the balance in the amount** | 1. **333** 2. **Y** 3. **W** 4. **501** | **Print “insufficient balance”** |
| 2.20 | **Restricted account: invalid withdraw (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type “w” in the amount** | 1. **333** 2. **Y** 3. **W** 4. **W** | **Print “invalid”** |
| 2.22 | **Restricted account: invalid deposit (exceed the restricted amount)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type “10001” in the amount** | 1. **333** 2. **Y** 3. **W** 4. **10001** | **Print "Exceed maximum withdrawal limit"** |
| 2.23 | **Overdue account: invalid deposit (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type negative value in the amount** | 1. **444** 2. **Y** 3. **D** 4. **-5** | **Print “cannot deposit negative value”** |
| 2.24 | **Overdue account: invalid deposit (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “d” in further input** 5. **Type non integer value in the amount** | 1. **444** 2. **Y** 3. **D** 4. **r** | **Print “invalid”** |
| 2.25 | **Overdue account: invalid withdraw (negative value)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type negative value in the amount** | 1. **444** 2. **Y** 3. **W** 4. **-40** | **Print “cannot withdraw negative value”** |
| 2.26 | **Overdue account: invalid withdraw (larger than the overdue balance)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type a larger value than the balance in the amount** | 1. **444** 2. **Y** 3. **W** 4. **101** | **Print “insufficient balance”** |
| 2.27 | **Overdue account: invalid withdraw (non interger input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type “y” in further action** 4. **Type “w” in further input** 5. **Type “w” in the amount** | 1. **444** 2. **Y** 3. **W** 4. **t** | **Print “invalid”** |
| **3.1** | **Input empty account number** | 1. **Run the program** 2. **Input account number** | **Press enter** | **Print “no such account”** |
| **3.2** | **Further action: empty** | 1. **Run the program** 2. **Input valid account number** 3. **Type empty data in further action** | 1. **111** 2. **Press enter** | **Print “invalid”** |
| **3.3** | **Further input: empty** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Empty further input** | 1. **111** 2. **Y** 3. **Press enter** | **Print “invalid”** |
| **3.4** | **Current account:deposit (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type d in further input** | 1. **111** 2. **Y** 3. **D** 4. **Press enter** | **Print “invalid”** |
| **3.5** | **Current account:withdraw (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type w in further input** 5. **Type empty value in amount** | 1. **111** 2. **Y** 3. **W** 4. **Press enter** | **Print “invalid”** |
| **3.6** | **Deposit account:deposit (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type d in further input** 5. **Type empty value in amount** | 1. **222** 2. **Y** 3. **D** 4. **Press enter** | **Print “invalid”** |
| **3.7** | **Deposit account:withdraw (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type w in further input** 5. **Type empty value in amount** | 1. **222** 2. **Y** 3. **W** 4. **Press enter** | **Print “invalid”** |
| **3.8** | **Restricted account:deposit (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type d in further input** 5. **Type empty value in amount** | 1. **333** 2. **Y** 3. **D** 4. **Press enter** | **Print “invalid”** |
| **3.9** | **Restricted account: withdraw (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type w in further input** 5. **Type empty value in amount** | 1. **333** 2. **Y** 3. **W** 4. **Press enter** | **Print “invalid”** |
| **3.10** | **Overdue account: deposit (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type d in further input** 5. **Type empty value in amount** | 1. **444** 2. **Y** 3. **D** 4. **Press enter** | **Print “invalid”** |
| **3.11** | **Overdue account: withdraw (empty input)** | 1. **Run the program** 2. **Input valid account number** 3. **Type y in further action** 4. **Type w in further input** 5. **Type empty value in amount** | 1. **444** 2. **Y** 3. **W** 4. **Press enter** | **Print “invalid”** |

# Screenshots show your test results

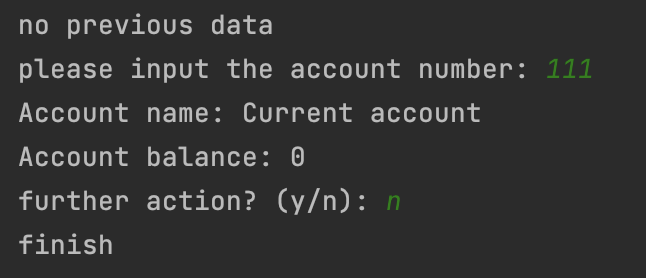
1.1



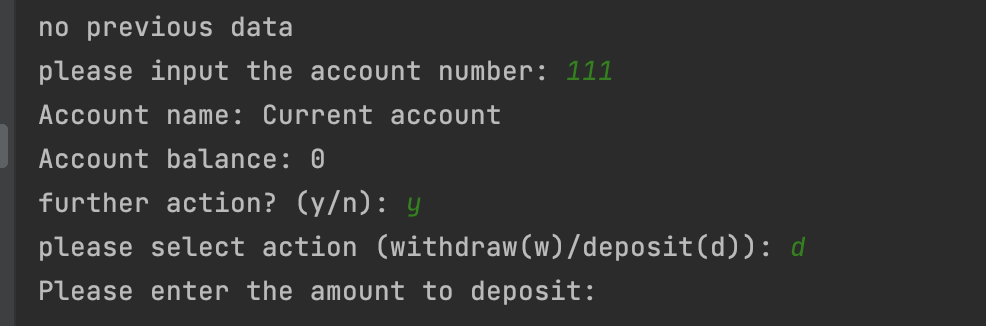
1.2



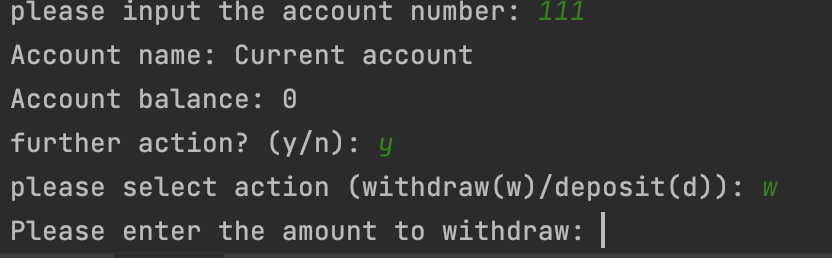
1.3



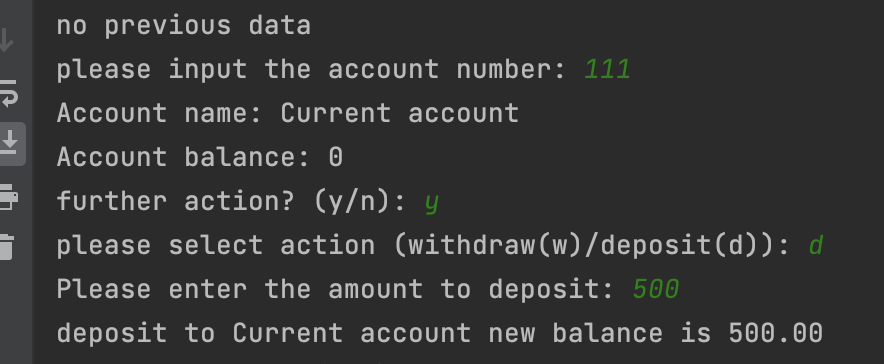
1.4



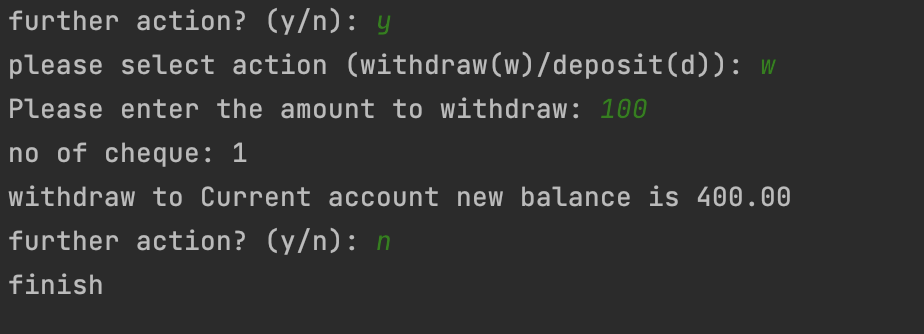
1.5



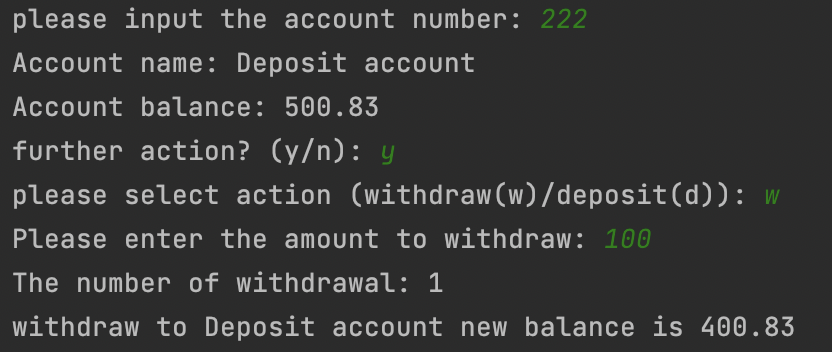
1.6



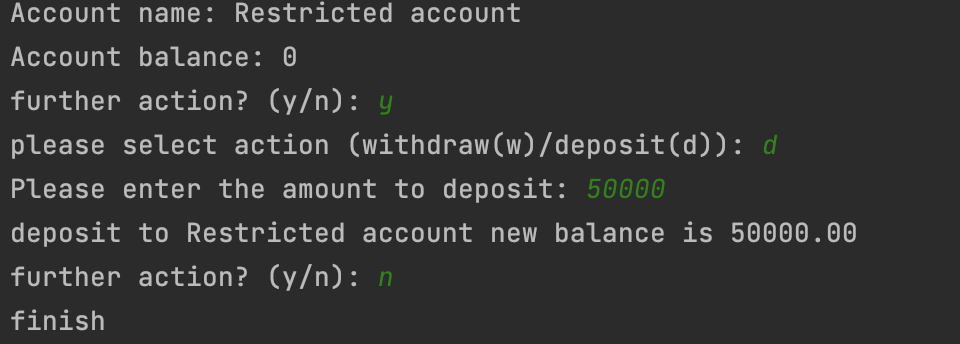
1.7



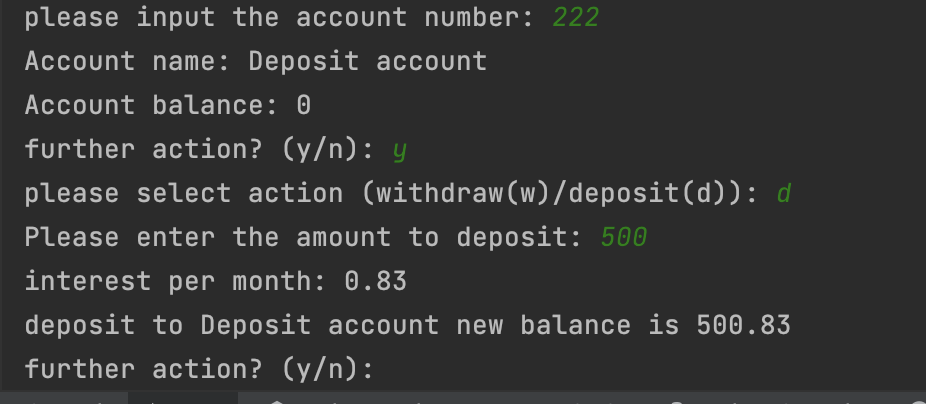
1.8



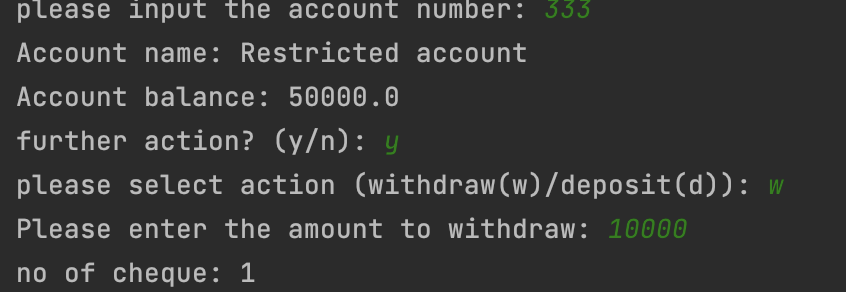
1.9



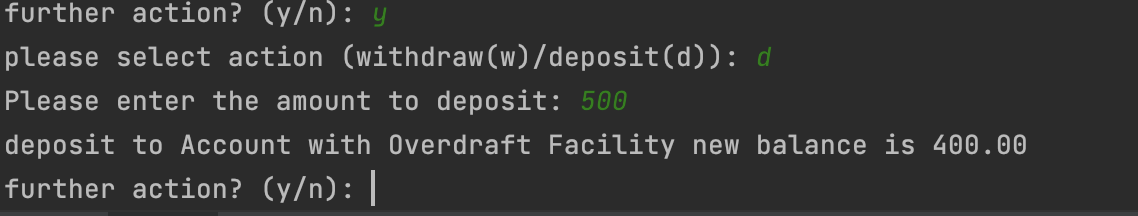
1.10



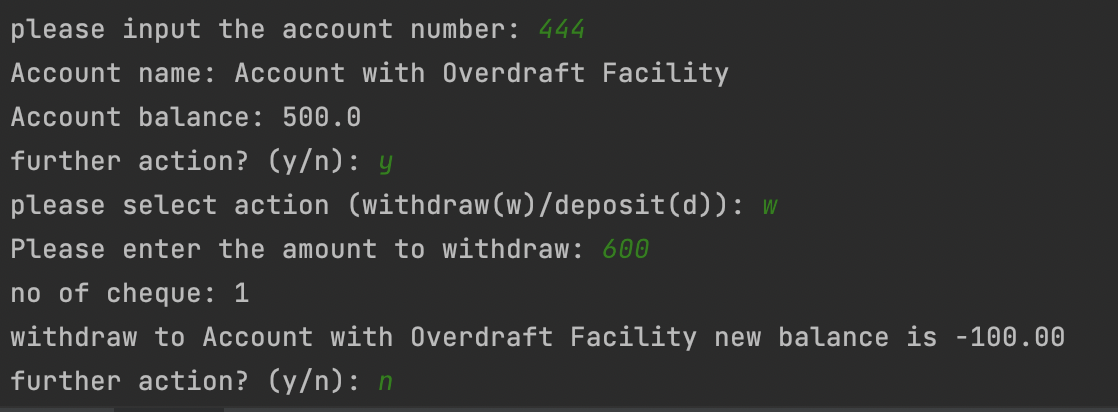
1.11



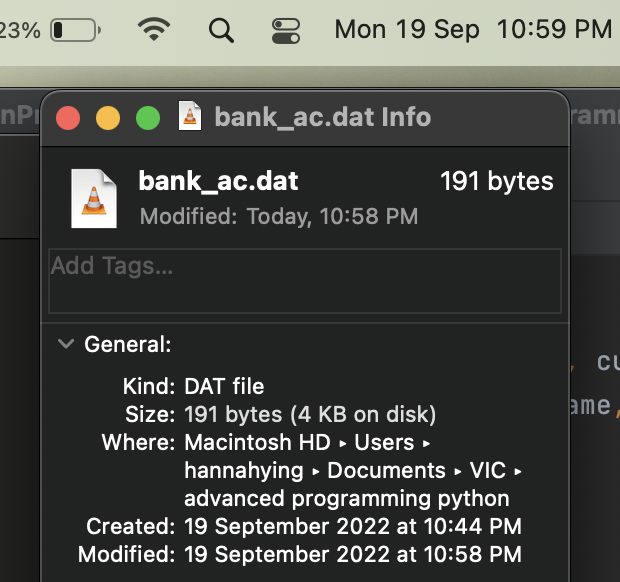
1.12



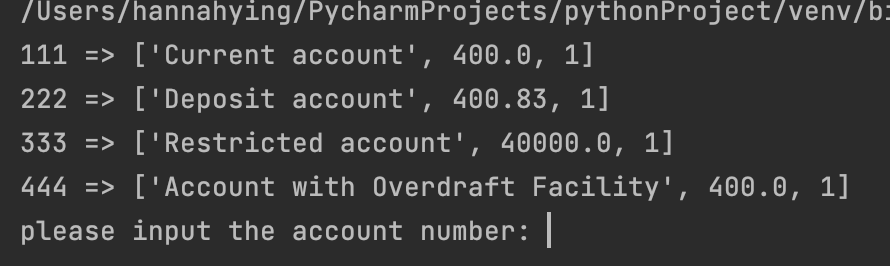
1.13



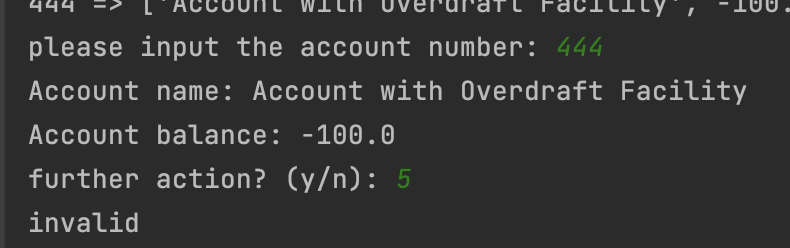
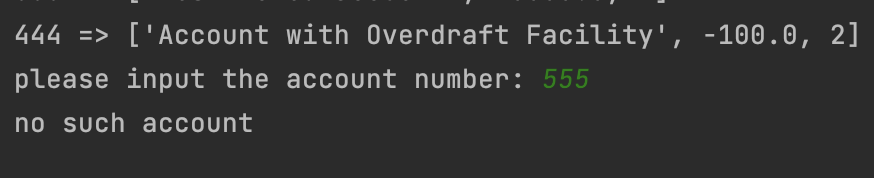
1.14



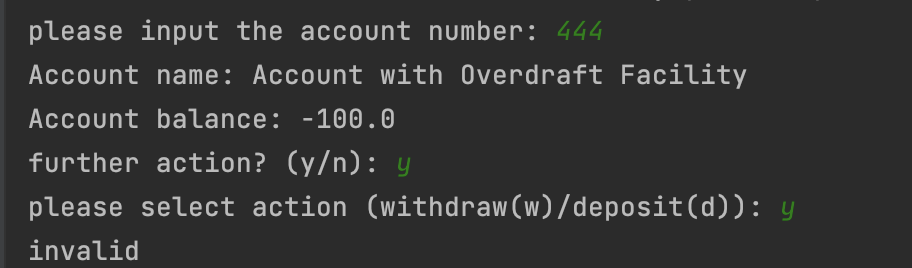
1.15



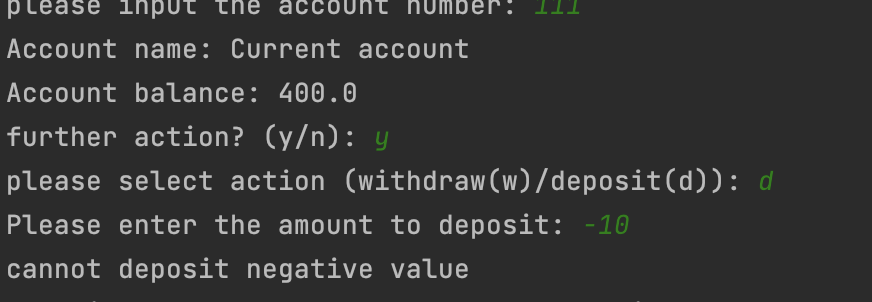
2.1

2.2

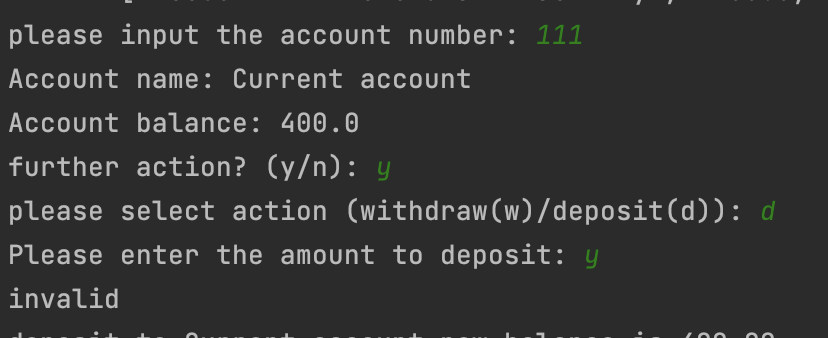
2.3



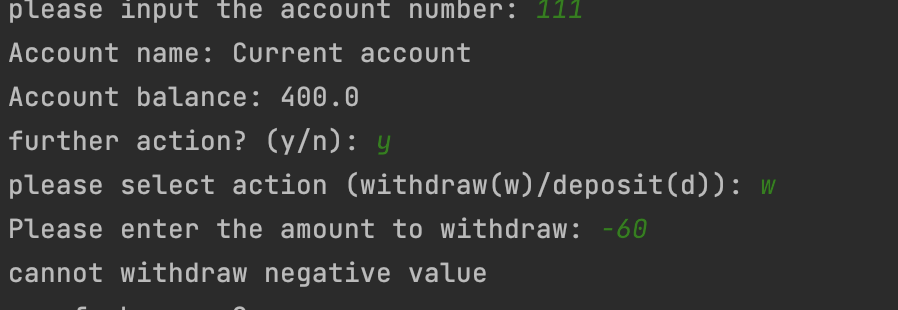
2.4



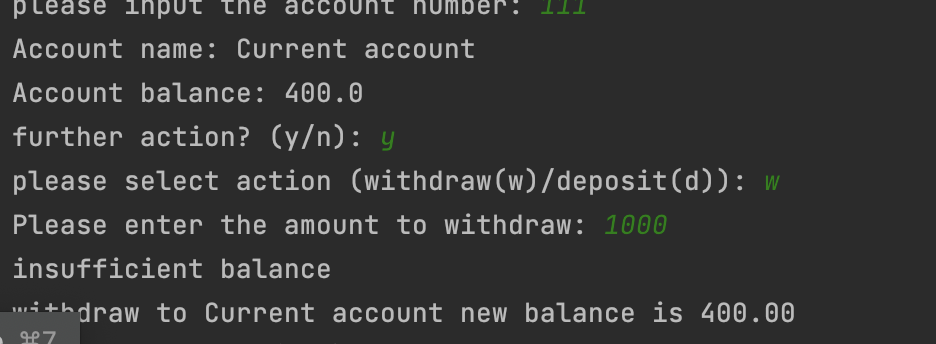
2.5



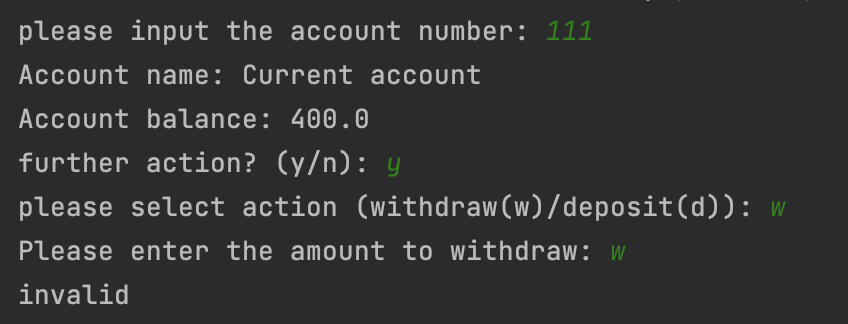
2.6



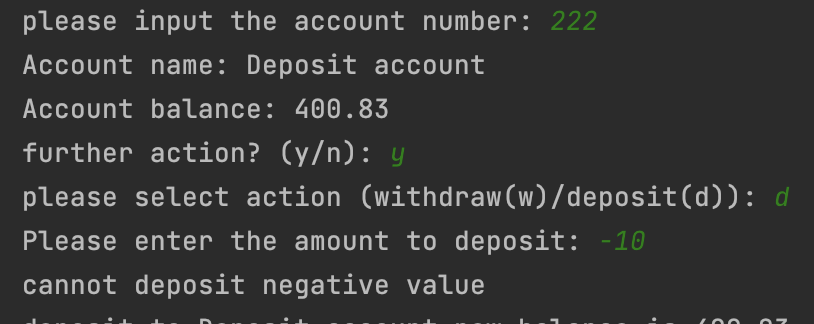
2.7



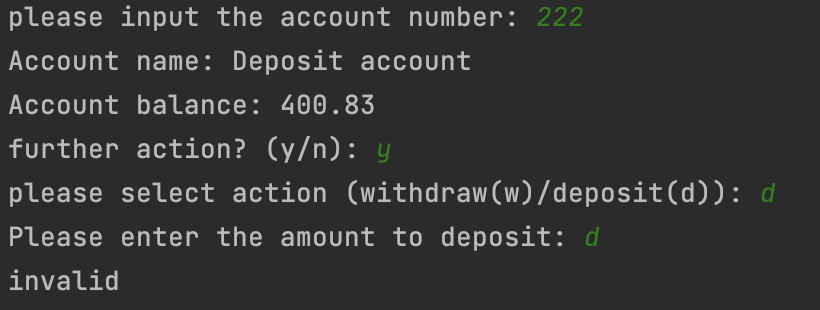
2.8



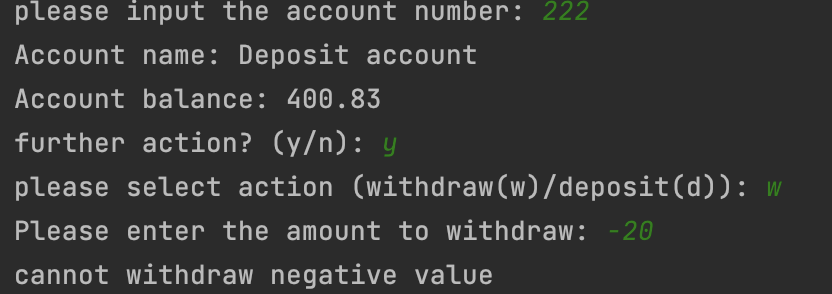
2.9



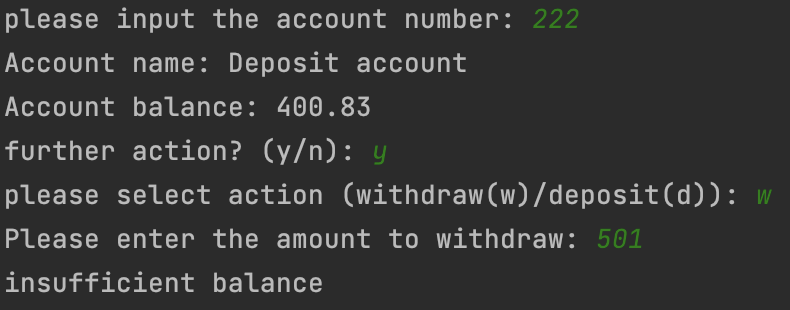
2.10



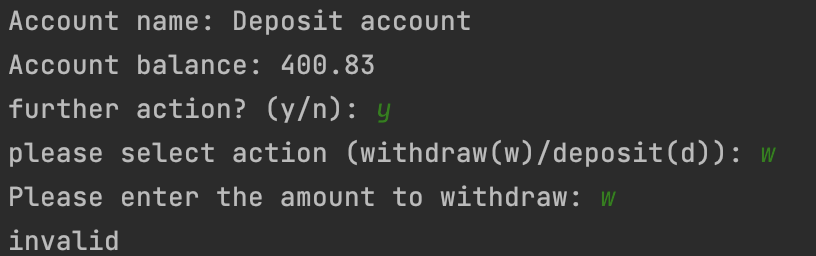
2.11



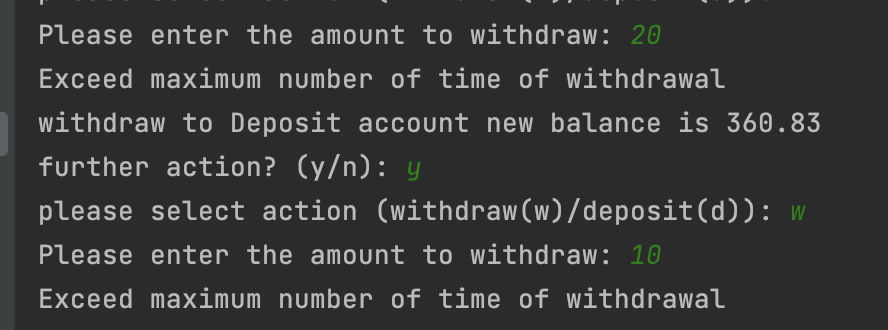
2.12



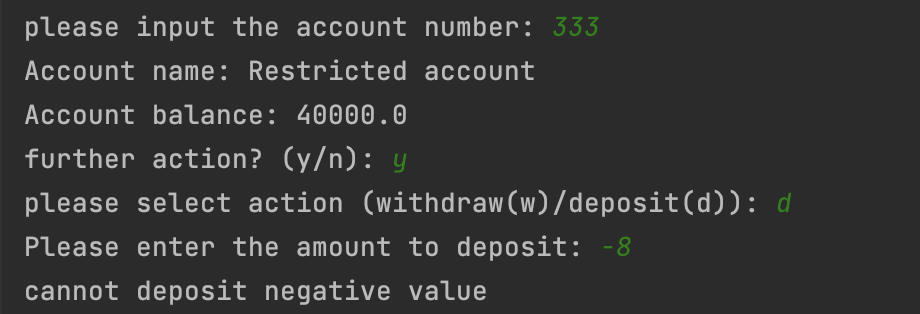
2.13



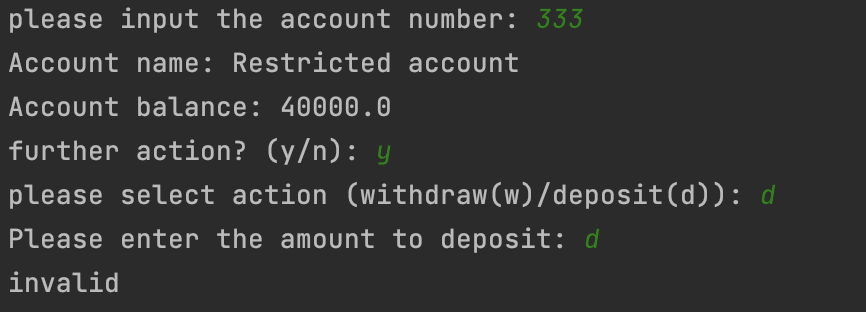
2.14



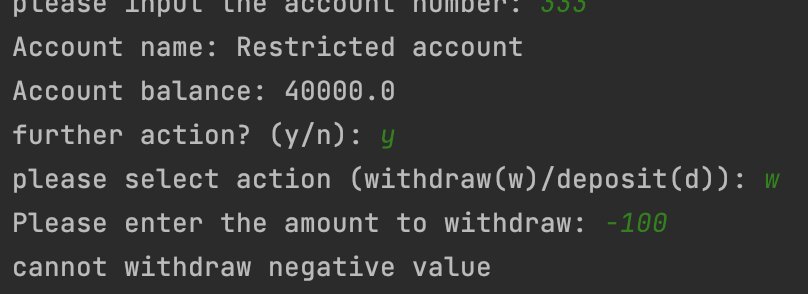
2.15



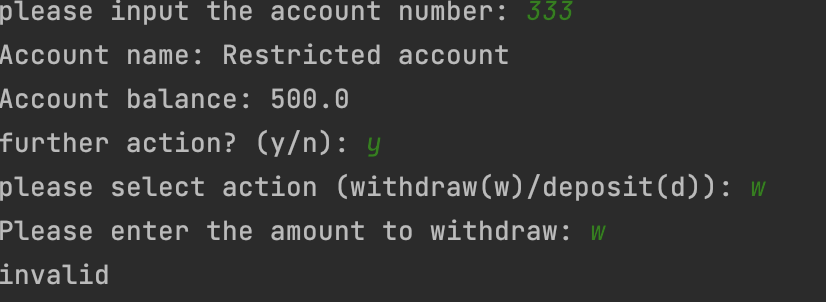
2.16



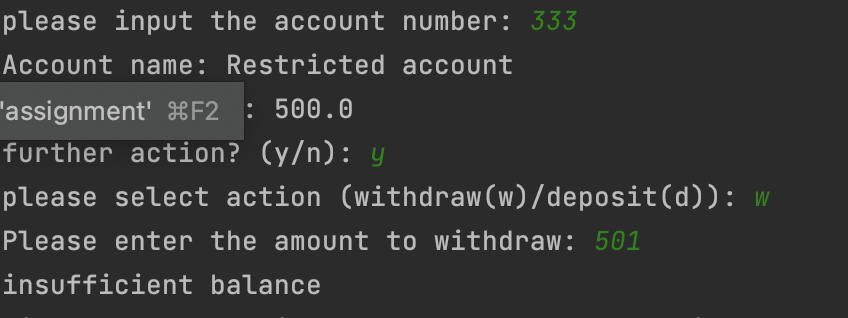
2.17



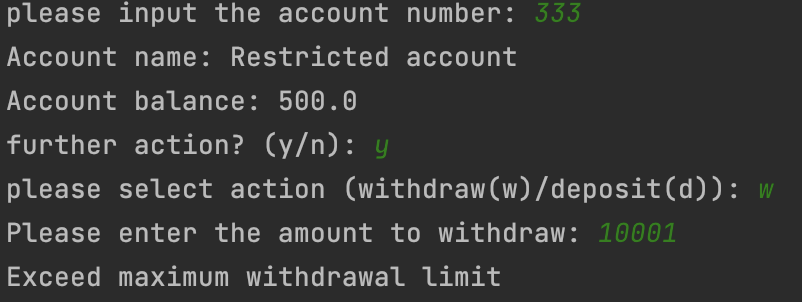
2.18



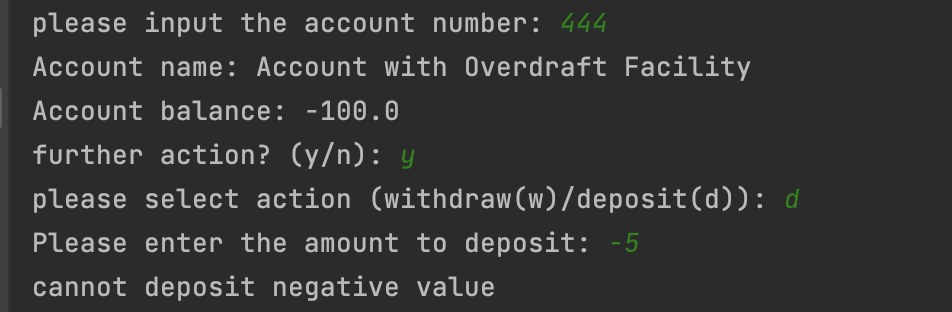
2.19



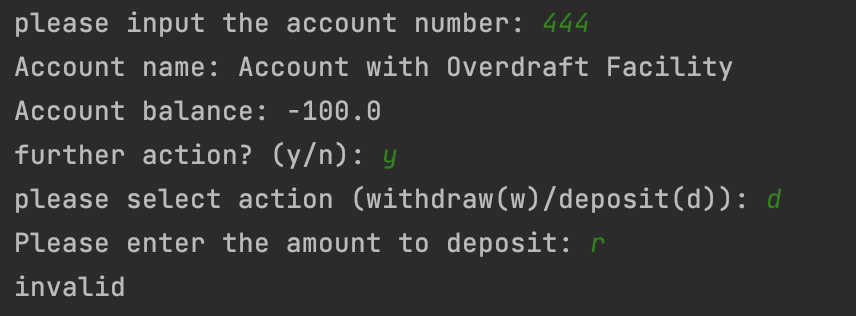
2.22



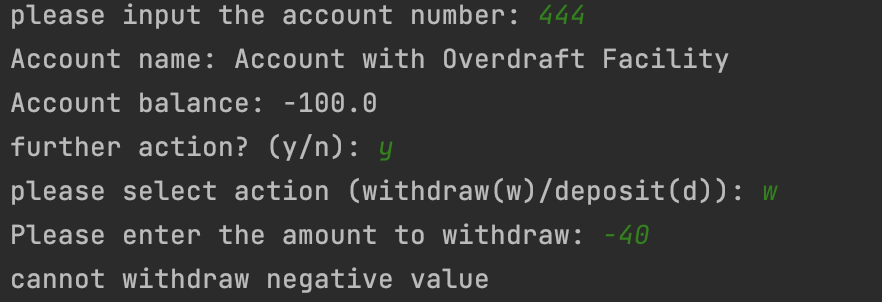
2.23



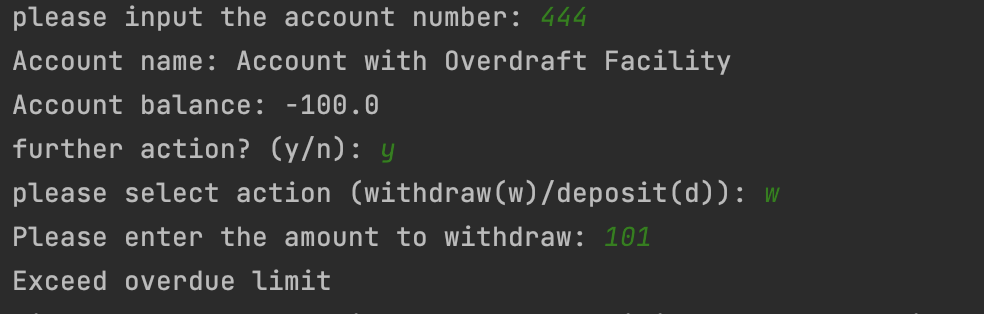
2.24



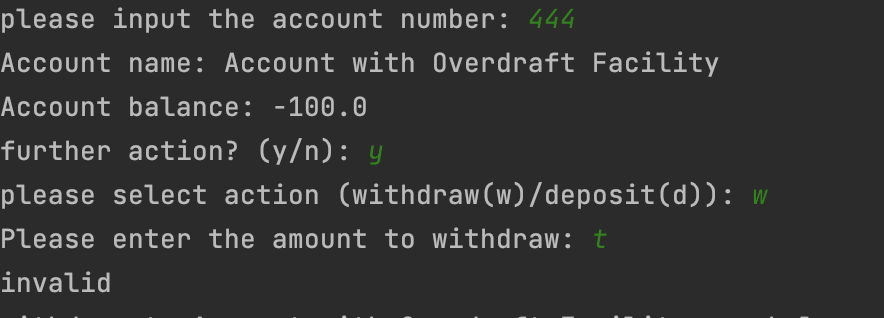
2.25



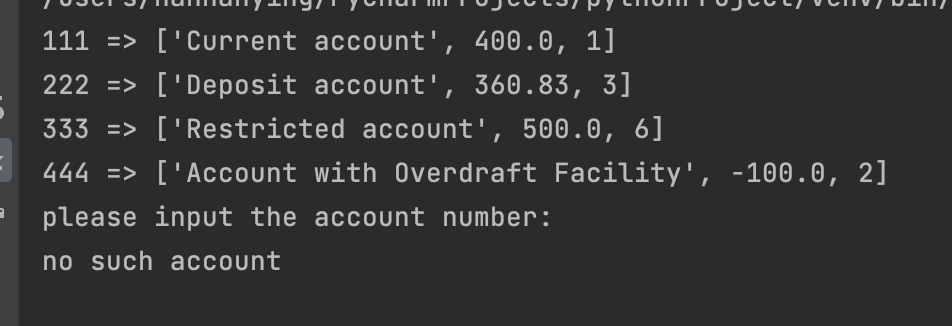
2.26



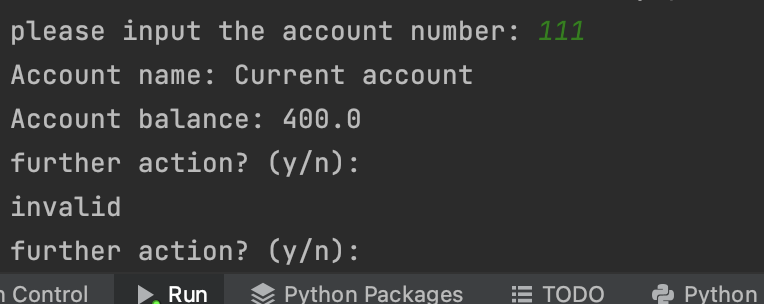
2.27



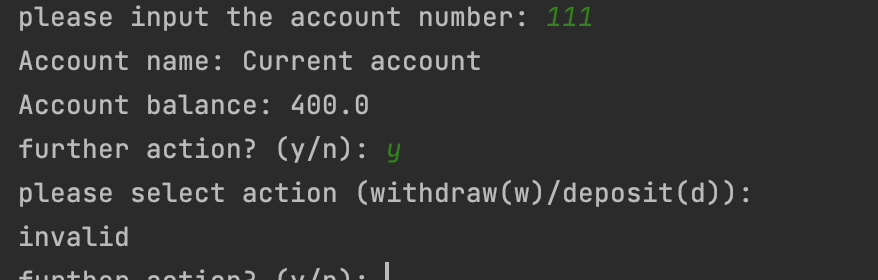
3.1



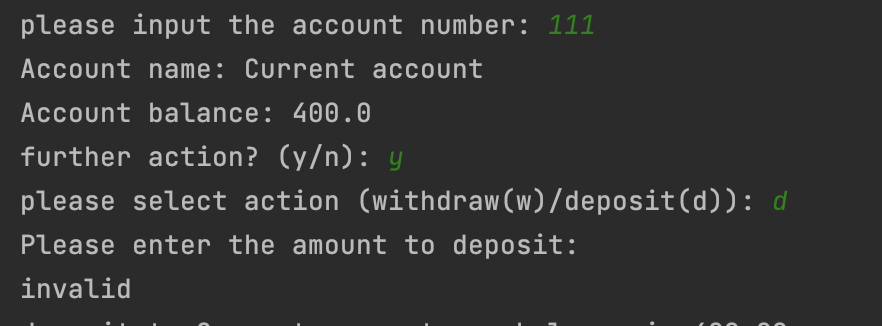
3.2



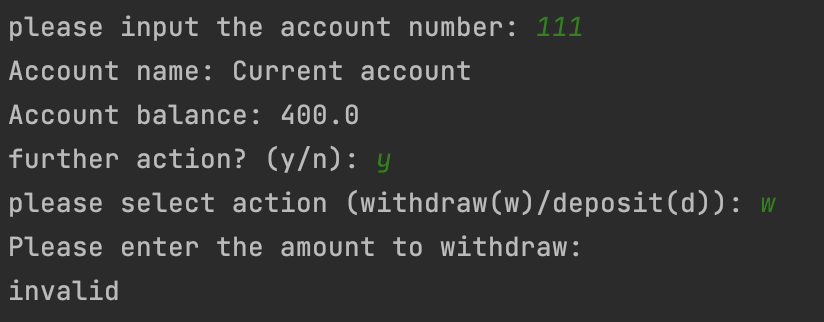
3.3



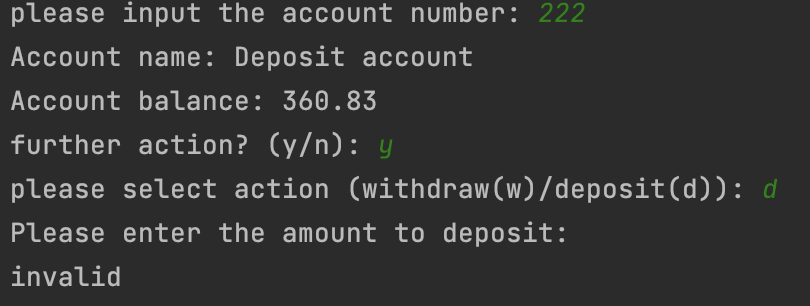
3.4



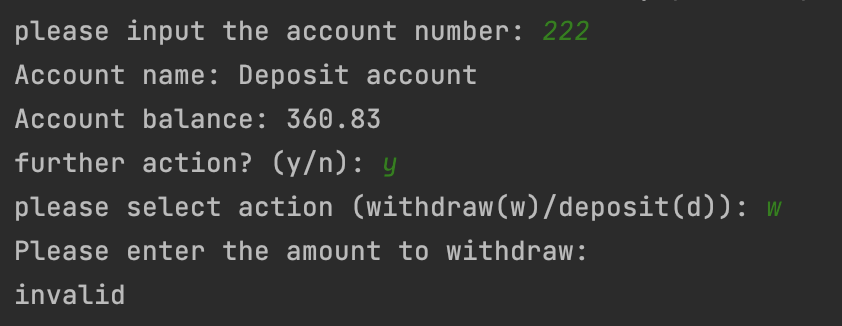
3.5



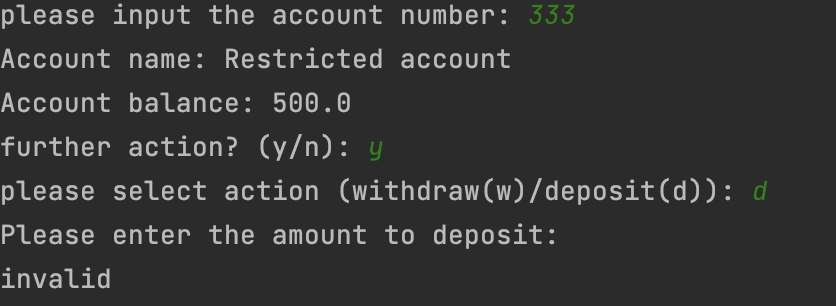
3.6



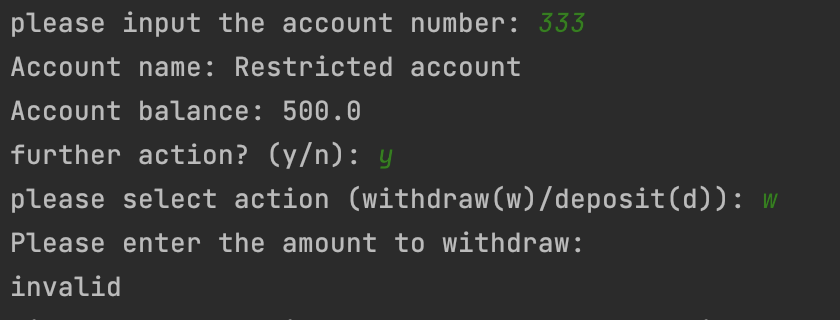
3.7



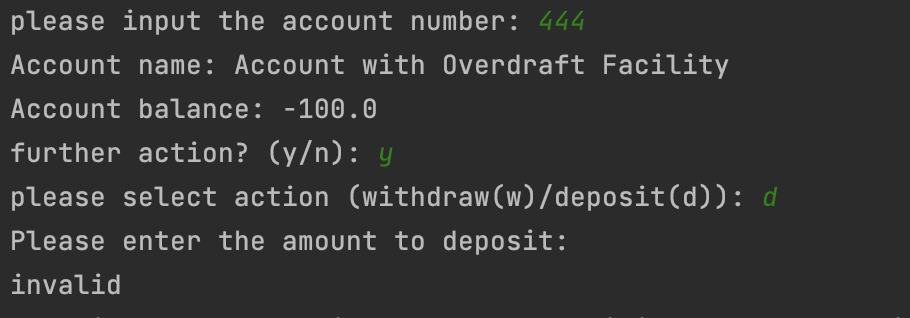
3.8



3.9



3.10



3.11

