

Documentation

Target assessment level

The target assessment level of this work is 3.

Specification

What does the program do?

The program is a banking system which

1. Asks if a person is a part of the bank
2. If no directs them to create an account
3. If yes asks them to login using account number
4. After login the program asks if the person would like to deposit or withdraw money and acts accordingly
5. The accounts with name, accountNumber and balance are stored in a text file
6. The program is also able to read the text file and determine if the account number provided matches one in the file
7. The program updates the balance of a person in the text file if they deposit or withdraw money

The user supplies the program with simple yes/no, deposit/withdraw or entering their name answers. All of the inputs are taken from the keyboard. Input is also taken from the text file when it is being read.

Data format

The text file to which the user is writing consists of lines, each line containing:

name, accountNumber, balance

All of them are stored as single words. When numbers in file need to read by the program `parseInt()` function is called to read them.

Correctness and exception handling

Typical test case

The person does not have an account with the bank and would like to create one.

```
Run: BankSystem x
C:\Users\rintala.nero\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Users\rintala.nero\AppData\Local\JetBrains
Are you a member of our bank? (yes/no): no
Would you like to open an account? (yes/no): yes
What's your name? (first name only): Nero
Your account has been created successfully!
Your account number is: 4202
Please remember remember this number as you won't be able to have access to your account otherwise.

Process finished with exit code 0
```

This is if they don't want to open an account.

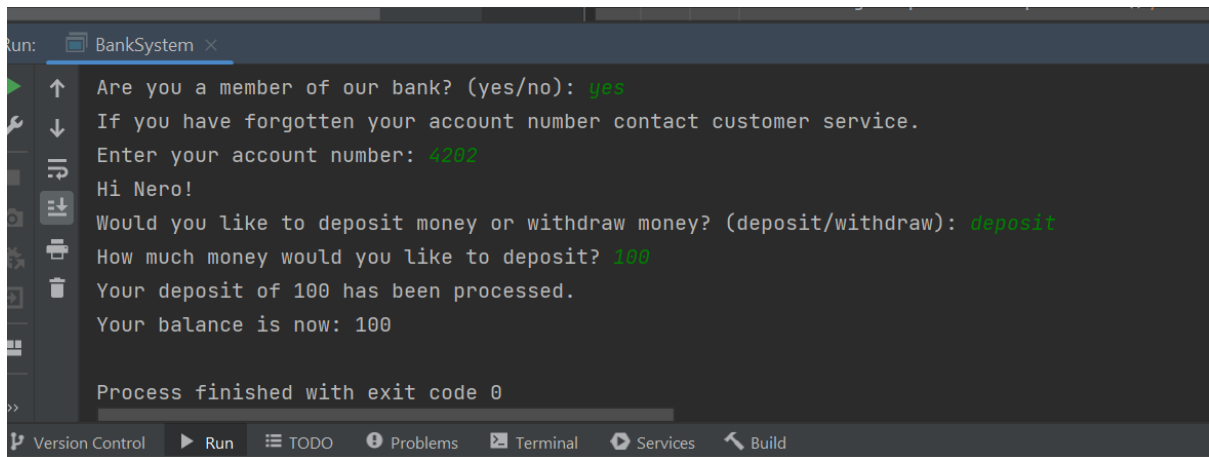
```
Run: BankSystem x
C:\Users\rintala.nero\.jdk\openjdk-19.0.1\bin\java.exe "-javaa
Are you a member of our bank? (yes/no): no
Would you like to open an account? (yes/no): no
Thank you for your visit! Have a nice day.

Process finished with exit code 0
```

This is if they already have an account with the bank.

```
Run: BankSystem x
C:\Users\rintala.nero\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Users\rintala.nero
Are you a member of our bank? (yes/no): yes
If you have forgotten your account number contact customer service.
Enter your account number: 4202
Hi Nero!
Would you like to deposit money or withdraw money? (deposit/withdraw): |
```

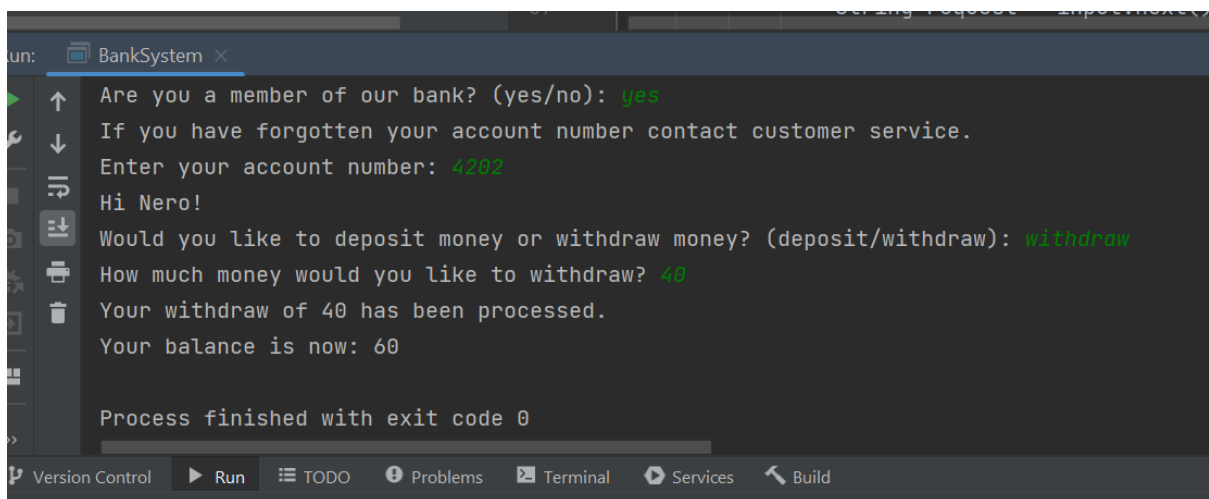
Then if you want to deposit money.



```
Run: BankSystem x
↑ Are you a member of our bank? (yes/no): yes
↓ If you have forgotten your account number contact customer service.
Enter your account number: 4202
Hi Nero!
Would you like to deposit money or withdraw money? (deposit/withdraw): deposit
How much money would you like to deposit? 100
Your deposit of 100 has been processed.
Your balance is now: 100

Process finished with exit code 0
Version Control Run TODO Problems Terminal Services Build
```

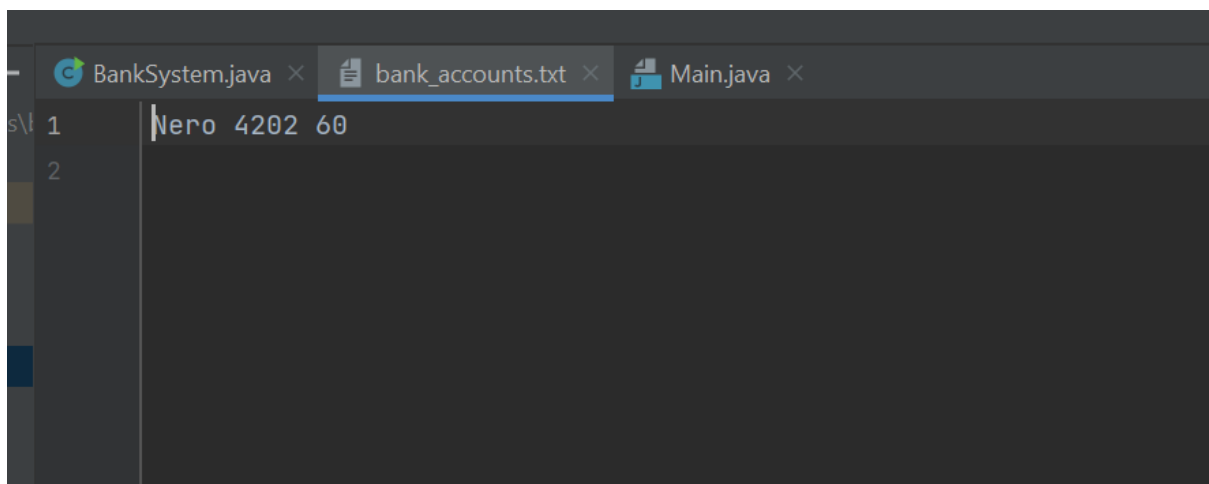
And lastly if you withdraw money.



```
Run: BankSystem x
↑ Are you a member of our bank? (yes/no): yes
↓ If you have forgotten your account number contact customer service.
Enter your account number: 4202
Hi Nero!
Would you like to deposit money or withdraw money? (deposit/withdraw): withdraw
How much money would you like to withdraw? 40
Your withdraw of 40 has been processed.
Your balance is now: 60

Process finished with exit code 0
Version Control Run TODO Problems Terminal Services Build
```

All of the data is also stored in the bank_accounts.txt file.



```
BankSystem.java x bank_accounts.txt x Main.java x
1 Nero 4202 60
2
```

Exception handling

The following are all the possible exceptions / special cases and the way they are handled.

1. If input is not yes/no the program asks the user to answer with yes/no
2. If input is not deposit/withdraw similar prompt is made
3. If the user inputs a wrong account number the program says: Invalid account number. Please enter a valid number.
4. If the withdraw amount exceeds the users balance the program says: Incufficient funds.
5. If you write a name with two words the program breaks. I was not able to fix that case.
6. The readLines function will catch an exception if the file is not able to be read.
7. Similarly the updateLines function will catch an exception if it is not able to update the balances in the file.

Wow-factor

I think that the way the code keeps track of a person's balance in the text file and updates it according to a member's deposits or withdrawals is a kind of wow-factor.

Resource management

The buffered reader resource is opened with the try-with-resources statement and is therefore closed automatically when the program no longer needs them, even in the case of an exception.

Reflection

Reflecting on the work it would have been useful to create a class to store the details of each bank account.

Program code

Attached in the GitHub repository.