Final project

Object Oriented Programming

Bank stimulation

# Submitted by

[Mujtaba Ahmed]

SP20-BSE-079

[Umair Arshad]

SP20-BSE-097

[Zeeshan Fareed]

SP20-BSE-101

# Github link

<https://github.com/Mujtaba-cpu/OOP-Final-Project.git>

# Introduction

Our stimulates a bank's working on 2 entities i.e.

-ACCOUNTS-

-EMPLOYEES-

They both have following in common:

* Personal details

1. Name
2. Phone Number
3. Address

For which we used a base class "Person" and derived 2 classes from it.

An account has the following attributes:

-Account Number

-Balance

An Employee has the following attributes:

-Employee Id

-Designation

-Salary

Then we have 2 classes

-AccountList

-EmployeeList

Which are used to store data regarding both entities,

AccountList works out different tasks performed on an Account and interreacts with a file named “Accounts”

EmployeeList works out different tasks performed on an Employee and interreacts with a file named “Employee List”

It implements following OOP Concepts

-Classes

-Inheritance

-Encapsulation

-Polymorphism

-Interface

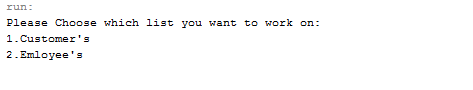
-Array List

-Exceptions

-File Handling

# STARTup interface

This basically asks you to choose on whom to want to work on



# customer interface-file before any chANGES

This is the interface when you choose option 1

Table

Description automatically generated

# eNTER NEW ACCOUNT

This adds a new account to the existing file

Text

Description automatically generated

Table

Description automatically generated

# Search account

It uses Account Number to find the Account from the file and then returns an object of type Account

Text

Description automatically generated

# Update accoUNT

This takes updated details as input and then updates the object and file

Text

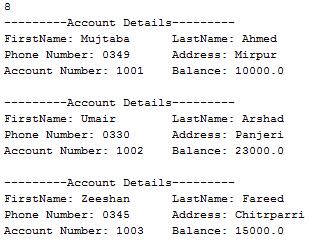
Description automatically generatedText

Description automatically generated

# delete account

Uses an account number to delete that specific account

A picture containing text

Description automatically generated 

# Withdraw money

Takes Account number and money as input, then deducts the specified amount from the relative account’s balance

Graphical user interface, text

Description automatically generatedText, letter

Description automatically generated

# deposit money

Takes Account number and money as input, then adds the specified amount to the relative account’s balance

Text, letter

Description automatically generatedText

Description automatically generated

# transfer money

Takes sender Account number, receiver Account number and money as input, then deducts the specified amount from the senders account’s balance and adds it to the receiver’s account balance.

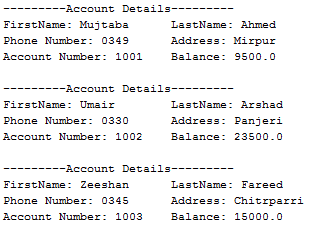
Text

Description automatically generatedText, letter

Description automatically generated

# display

Display all objects in file



# employee interface-file before any changes

Text

Description automatically generatedText, letter

Description automatically generated

# add, search, update and delete

These functions are same as above with minor changes i.e.

Instead of acc number employee id is used, and these functions interacts with employee class and file “Employee List”

# increment salary

This takes employee id and the increment percentage as input and the increments the salary of the relative employee by the given percentage.

Text

Description automatically generated Graphical user interface, text

Description automatically generated