# AccountHolder.java

//AccountHolder.java

package accountHolder;

import java.text.DateFormat;

import java.text.DecimalFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

/\*\*

\* This program is for Lab 1 of ITMD 511

\* The program is designed to simulate a banking application where

\* A user can set their initial balance via the constructor

\* And make deposits and withdrawals and run through their interest

\* Gain at 4%. Additional features include the ability to change their

\* Interest rate to see if another bank's interest would be more beneficial

\* To the user. Accounts are charged $50 fee if withdrawal drops the balance

\* Below $500, and their ban\lance cannot go below $100.

\* @author Chris Doherty

\* Completed testing on 2/4/2018 @ 11:03am

\*

\*/

public class AccountHolder {

private static double annualInterestRate = 0.04;

private double balance;

/\*\*

\* \* Constructor.

\* Accept initial balance

\* Set balance equal to value passed

\* include error trapping for negative starting balance

\* @param initialbalance

\*/

public AccountHolder(double initialbalance) {

// TODO Auto-generated constructor stub

balance = initialbalance;

}

/\*\*

\* getter generated

\* @return

\*/

public static double getAnnualInterestRate() {

return annualInterestRate;

}

/\*\*

\* setter generated

\* @param annualInterestRate

\*/

public static void setAnnualInterestRate(double annualInterestRate) {

AccountHolder.annualInterestRate = annualInterestRate;

}

/\*\*

\* getter generated

\* @return

\*/

public double getBalance() {

return balance;

}

/\*\*

\* setter generated

\* @param balance

\*/

public void setBalance(double balance) {

this.balance = balance;

}

/\*\*

\* deposit instance method

\* need error trapping

\* @param balance

\*/

public void deposit(double balance){

double bal = getBalance();

bal += balance;

setBalance(bal);

}

/\*\*

\* withdrawal instance method

\* need error trapping

\* @param withdrawAmt

\*/

public void withdrawal(double withdrawAmt){

double currentbal = getBalance();

currentbal -= withdrawAmt;

setBalance(currentbal);

}

/\*\*

\* interest instance method

\*/

public void monthlyInterest(){

double bal = getBalance();

double interest = getAnnualInterestRate();

DecimalFormat df = new DecimalFormat("####0.00");

//loop to calculate and print out balance at each month

System.out.println("Monthly balance for one year at " + interest);

String outBal = df.format(bal);

System.out.format("Month: \t\t Balance\n", outBal);

System.out.format("Base: \t\t$%8s\n", outBal);

for(int i=1; i<13; i++){

//update the account holders’ balance

bal = (bal + (bal \* (interest / 12.0)));

outBal = df.format(bal);

System.out.format("Month " + i + ": \t$%8s\n", outBal);

/\*

\* trunk output to 2 numbers

\*/

}

//sets the balance after 12 months of interest

setBalance(bal);

setAnnualInterestRate(0.04);

//reset interest to 4% again

exitPrintout();

}

/\*\*

\* Static monthly interest

\* @param balance

\*/

public static void modifyMonthlyInterest(double rate){

//assign passed rate to variable

double annualir = (rate/100);

setAnnualInterestRate(annualir);

}//closes modifyMonthlyInterest

public void exitPrintout(){

DateFormat tf = new SimpleDateFormat("dd/MM/yy HH:mm:ss");

Date date = new Date();

System.out.println("Current Date " + tf.format(date));

System.out.println("Created By Chris Doherty \n");

}

/\*\*

\* toString instance method

\*/

public String toString(){

return String.format("$%.2f", balance);

}//closes toString

@SuppressWarnings("resource")

public void menu(){

//integer to capture what the user wants to do

int choice = 0;

//creates the scanner object

Scanner scanner = new Scanner(System.in);

//loop to keep menu displayed until the user puts in a correct input

//output the menu choices

System.out.println("What would you like to do?");

System.out.println("1. Make a deposit");

System.out.println("2. Make a withdrawal");

System.out.println("3. See Balance");

System.out.println("4. See what the balance would be with 4% rate");

System.out.println("5. See what the balance would be with any interest rate.");

System.out.println("6. Exit");

exitPrintout();

//while statement to check if the input is a int

while(!scanner.hasNextInt()){

String input = scanner.next();

//error message if the input is not an int

System.out.printf("\"%s\" is not a valid choice.\n Please choose 1 - 5 \n", input);

}

//assign variable to input

choice = scanner.nextInt();

//} while (choice>0 && choice<5); //keeps the loop up while input is not an int

//switch statement to do what the user wants to do

switch (choice) {

//switch goes to here if 1 is chosen

case 1:

//variable for user input

double deposit;

//loop to keep menu displayed until the user puts in a correct input

do {

//asks user the deposit amount

System.out.println("How much would you like to deposit?");

//while statement to check if the input is a double

while (!scanner.hasNextDouble()){

//error message if the input is not a double

System.out.println("That is not a valid amount to deposit. Please try again");

scanner.next();

}

//assigns user input to variable

deposit = scanner.nextDouble();

} while (deposit < 0.0); //keeps the loop up while input is not an int

//passes the deposit amount to the deposit method

deposit(deposit);

break;

//switch goes to here if 2 is chosen

case 2:

//variable for user input

double withdrawal;

double bal = getBalance();

if (bal > 100){

//loop to keep menu displayed until the user puts in a correct input

do {

//asks user the withdrawal amount

System.out.println("How much would you like to withdrawal?");

//while statement to check if the input is a double

while (!scanner.hasNextDouble()){

//error message if the input is not a double

System.out.println("That is not a valid input. Please try again");

scanner.next();

}

//assigns user input to variable

withdrawal = scanner.nextDouble();

}

//keeps the loop up while input is not an int

while (withdrawal < 0.0);

//takes the balance to and sub withdraw desired amount stores it a test var

double endAmountAfterWithdrawal = (bal - withdrawal);

//checks to make sure the balance will not drop below $100

if(endAmountAfterWithdrawal <= 100){

System.out.println("Cannot have balance drop below 100");

}

//prints out the balance of the system

else if(endAmountAfterWithdrawal < 500) {

//statement to notify user they have incurred a fee

System.out.println("Your account balance is less than $500. You have incurred a $50 fee.");

double amtToWithdraw = (withdrawal + 50);

withdrawal(amtToWithdraw);

}else{

withdrawal(withdrawal);

}

}else{

System.out.println("Balance is already below 100. \n You cannot make a whithdrawal at this time.");

}

break;

//switch goes to here if 3 is chosen

case 3:

//program will print out the balance in the system

System.out.println(getBalance());

break;

//switch goes to here if 4 is chosen

case 4:

/\*\*

\* call annual interest rate at 4%

\*/

monthlyInterest();

break;

//switch goes to here if 5 is chosen

case 5:

//variable for user input

double rate;

//loop to keep menu displayed until the user puts in a correct input

do {

//asks user the withdrawal amount

System.out.println("What would you like your annual interest rate percentage to be?");

//while statement to check if the input is a double

while (!scanner.hasNextDouble()){

//error message if the input is not a double

System.out.println("That is not a valid rate. Please a number.");

scanner.next();

}

//assigns user input to variable

rate = scanner.nextDouble();

if(rate<0.0 || rate>100){

System.out.println("That is not a valid rate. Please input an interest rate from 0 to 100 percent.");

rate=-1;

}

}

//keeps the loop up while input is not valid

while (rate < 0.0);

//sets the annual interest rate

modifyMonthlyInterest(rate);

monthlyInterest();

break;

//switch goes to here if 6 is chosen

case 6:

exitPrintout();

//program will exit

System.exit(0);

break;

//switch goes to here if 1-4 are not selected

default:

//print this out if all else fails

System.out.println("Invalid Choice");

break;

}//closes switch

}

}

# AccountHolderTest.java

//AccountHolderTest.java

/\*\*

\* This program is for Lab 1 of ITMD 511

\* The program is designed to simulate a banking application where

\* A user can set their initial balance via the constructor

\* And make deposits and withdrawals and run through their interest

\* Gain at 4%. Additional features include the ability to change their

\* Interest rate to see if another bank's interest would be more beneficial

\* To the user. Accounts are charged $50 fee if withdrawal drops the balance

\* Below $500, and their ban\lance cannot go below $100.

\* @author Chris Doherty

\* Completed testing on 2/4/2018 @ 11:03am

\*

\*/

package accountHolder;

import java.text.DecimalFormat;

import java.util.Scanner;

public class AccountHolderTest {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

AccountHolder aObj = null;

@SuppressWarnings("resource")

Scanner scanner = new Scanner(System.in);

DecimalFormat df = new DecimalFormat("####0.00");

// Ask user for beginning balance

double initialbalance = 0.0;

// loop for error trapping

do {

System.out.println("What is your initial Balance?");

while (!scanner.hasNextDouble()) {

System.out.println("Input is not valid. Balance must be a number. Please try again.");

scanner.next();

}

initialbalance = scanner.nextDouble();

if (initialbalance < 0.0) {

System.out.println("Input is not valid. Balance cannot be negative.");

}

} while (initialbalance < 0.0);

// instantiate class

aObj = new AccountHolder(initialbalance);

int again = 0;

// calls the menu method

do {

if (again == 0 || again == 1){

aObj.menu();

}

//asks the user if they want to make another transaction

System.out.println("Your current balance is " + df.format(aObj.getBalance()));

System.out.println("Did you want to make another transaction?");

System.out.println("1. Yes");

System.out.println("2. Exit");

aObj.exitPrintout();

// while statement to check if the input is a double

while (!scanner.hasNextInt()) {

// error message if the input is not a double

System.out.println("That is not a valid choice. Please input either 2 or 1");

scanner.next();

}

// assigns input to variable

again = scanner.nextInt();

//prints out my name and date

if (again == 2) {

aObj.exitPrintout();

}else {again = 0;}

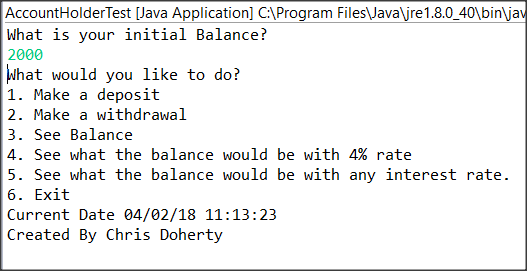
} while (again != 1 || again != 2); // keeps the loop up while input is not 2

}//closes main

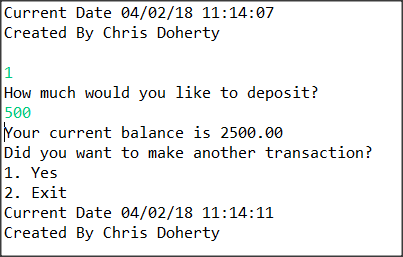
}//closes class

# Screen Shots

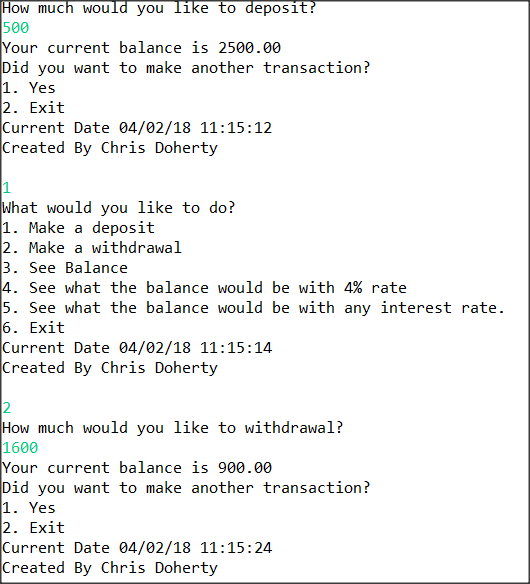
## Settinging Initial Balance



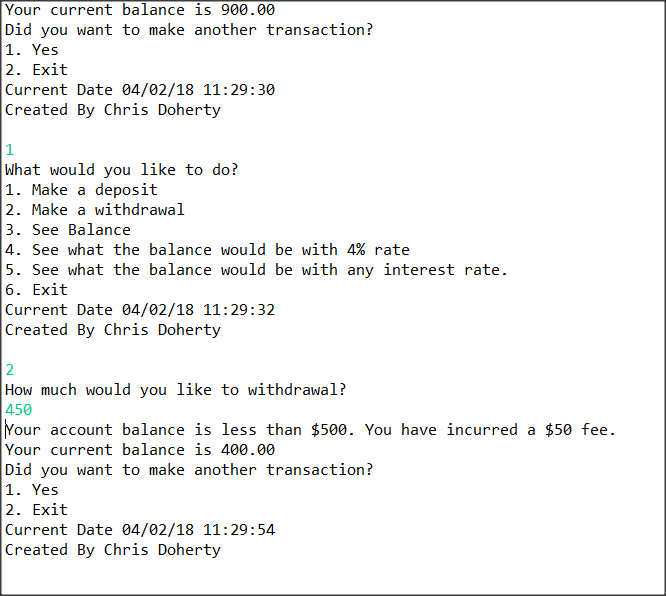
## Making A Deposit



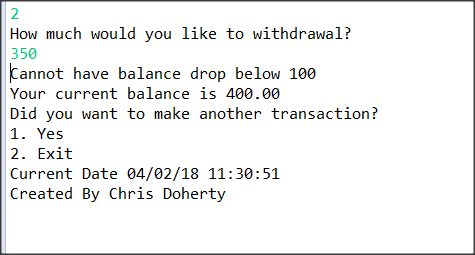
## Making A Withdrawal



## Displaying Fee Assessed When Balance Drops Below $500



## Displaying Balance’s Inability To Drop Below $100



## Calculating Monthly Interest At 4%

## 

## Calculating Interest at 6%

