/\*\*

\* @(#)BhaktaBonnerScarsella003PA2.java

\* @author Jay Bhakta, Braden Bonner, and Avery Scarsella

\* version 1.00 2023/10/19 4:00 PM

\*

\* PROGRAM PURPOSE: Create a program for calculating the cost of

\* intended stock purchases for multiple people who trade stocks.

\*

\* Collaboration Tools: GroupMe chat, Discord call, and Zoom video call.

\*/

import java.util.Scanner; //By Braden: Class to access keyboard entries.

import java.util.Calendar; //By Avery: Class to access the system's date.

public class BhaktaBonnerScarsella003PA2

{

/\*\*

\* Investers can choose to proceed with the stock calculator

\* or not. If not, a thank you message is displayed; otherwise,

\* investors are asked to enter their name. Data pertaining to

\* the calculation is requested. The stock cost is calculated

\* and added to the respective totals. An online fee or commission

\* is calculated and added to their respective totals unless the

\* trade type is invalid. Investors can assess the costs for multiple

\* stocks. Multiple Investors can calculate stocks.

\* Once there are no more stock costs, the final output is

\* printed and a thank you message is displayed.

\*/

private static Scanner input = new Scanner(System.in); //By Jay: REF variable or object to read input from

//the keyboard

private static int shares = 0; //By Avery: Initialize shares to the default value. Stores the number of shares.

private static double sharePrice = 0.0; //By Avery: Initialize sharePrice to the default value. Stores sharePrice.

private static char anotherTrader = ' '; //By Braden: Initialize anotherTrader to the default value.

private static char anotherStock = ' '; //By Braden: Initialize anotherStock to the default value.

private static String stockCostRpt = String.format("%n%nSTOCK COST REPORT%n"); //By Jay: Initialize stockCostRpt

//with “%n%nSTOCK COST REPORT%n” using String.format()

/\*\*

\* Main method body containing logic for program and method calls

\* calculating the cost of intended stock purchases for multiple people who

\* trade stocks.

\*/

public static void main (String[] args)

{

String customerName = ""; //By Jay: Object for a customer's name.

int noStocks = 0; //By Avery: Variable to track the number of stocks in the calculation.

double stockCost = 0.0, //By Avery: Initialize stockCost to the default value. Stores stock cost.

commission = 0.0, //By Avery: Initialize commission to the default value. Stores commission.

totalCost = 0.0, //By Braden: Initialize totalCost to the default value. Stores totalCost.

onlineFee = 0.0, //By Braden:Initialize onlineFee to the default value. Stores onlineFee.

totalStockCost = 0.0, //By Jay: Initialize totalStockCost to the default value. Stores totalStockCost.

totalCommissions = 0.0, //By Jay: Initialize totalCommissions to the default value. Stores totalCommissions.

totalOnlineFees = 0.0; //By Jay: Initialize totalOnlineFees to the default value. Stores totalOnlineFees.

boolean alpha = false; //By Braden: Initialize alpha to false.

char onlineTrade = ' '; //By Avery: Initialize onlineTrade to the default value. Stores onlineTrade.

char brokerAssisted = ' '; //By Jay: Initialize brokerAssisted to the default value. Stores brokerAssisted.

promptAnotherTrader();

while (anotherTrader == 'Y')

{

noStocks = 0;

totalCommissions = 0.0;

totalOnlineFees = 0.0;

totalStockCost = 0.0;

totalCost = 0.0;

String name = setCustomerName();

do

{

alpha = isAlpha(name);

if(alpha)

{

customerName = capitalize(name);

alpha = true;

}//By Jay: END if(alpha)

else

{

System.out.printf("%n%s is not alphabetic.%n", name);

name = setCustomerName();

alpha = isAlpha(name);

customerName = (alpha) ? capitalize(name) : "";

}//By Jay: END else if(alpha)

} while (!alpha);

//By Jay: END do... while !alpha

promptAnotherStock();

while(Character.toUpperCase(anotherStock) == 'Y')

{

++noStocks;

setShares();

setSharePrice();

input.nextLine();

stockCost = calcStockCost();

totalStockCost += stockCost;

totalCost += stockCost;

onlineTrade = promptOnlineTrade();

if(Character.toUpperCase(onlineTrade) == 'Y')

{

onlineFee = 5.95;

totalOnlineFees += onlineFee;

totalCost += onlineFee;

}//By Avery: END if onlineTrade == 'Y'

else

{

brokerAssisted = promptBrokerAssisted();

if(Character.toUpperCase(brokerAssisted) == 'Y')

{

commission = calcCommission(stockCost);

input.nextLine();

totalCommissions += commission;

totalCost += commission;

}//By Jay: END else onlineTrade = 'Y'

else

{

System.out.printf("%nINVALID TRADE TYPE!%n"); //By Jay: Displaying INVALID TRADE TYPE

noStocks--;

totalStockCost -= stockCost;

totalCost -= stockCost;

}//By Avery: END else brokerAssisted = 'Y'

}//By Braden: END else onlineTrade = 'Y'

repromptAnotherStock();

}//By Jay: END while anotherStock == 'Y'

if(noStocks > 0)

{

stockCostRpt += formatFinalOutput(customerName, totalStockCost, totalOnlineFees,

totalCommissions, totalCost);

}//By Braden: END if noStocks > 0

repromptAnotherTrader();

}//By Avery: END while anotherTrader == 'Y'

if(noStocks > 0)

{

System.out.printf(stockCostRpt); //By Jay: Displaying stockCostRpt

}//By Avery: END noStocks > 0

printThankYouMessage();

System.exit(0); //By Jay: Exits the program

}//By Avery: END main(args: String[]): static void

/\*\*

\* Prints the company header and welcome message.

\* This primes the sentinel-loop control variable anotherTrader before entering the

\* outer while that controls each trader. Prompts and reads as uppercase.

\*/

public static void promptAnotherTrader()

{

System.out.printf("%nYEE-TRADE, INC. The Wild West of Electronic Trading%n"

+ "%nWelcome to Yee-Trade\'s stock cost calculator.%n");

System.out.printf("%nReady to generate a stock cost report? Enter \'Y\' or \'N\' to exit: ");

anotherTrader = input.nextLine().toUpperCase().charAt(0);

}//By Jay: END promptAnotherTrader(): static void

/\*\*

\* Prompts for the customer’s name and returns it from the keyboard.

\*/

public static String setCustomerName()

{

System.out.printf("%nWhat is your name? ");

return input.nextLine();

}//By Jay: END etCustomerName(): static String

/\*\*

\* Tests whether a value is an alpha.

\*/

public static final boolean isAlpha(String word)

{

/\* Strip non-alpha characters commonly found in names. \*/

word = new String(word).replace(".", "");

word = new String(word).replace(",", "");

word = new String(word).replaceAll("\\s+", "");

/\* Test to see if the word is not empty AND if each letter

\* in a word is an alphabetic character.

\*/

return word != null && word.chars().allMatch(Character ::

isLetter);

}//By Jay: END isAlpha(word: String): static final boolean

/\*\*

\* Capitalizes the first letter in a name.

\*/

public static final String capitalize(String str)

{

String words[] = str.split("\\s"); //Each word in str is an

//element in the array.

String capitalized = "", //Stores what came in the str

//with the correct capitalization.

firstWord = "", //Stores 1st letter of the str.

wordAfter = ""; //Stores the remaining letters in the str.

for(String aWord : words)

{

firstWord = aWord.substring(0, 1);

wordAfter = aWord.substring(1);

capitalized += firstWord.toUpperCase() + wordAfter.toLowerCase()

+ " ";

}//for each word from a String in the words array, capitalize the

//first letter

return capitalized.trim(); //Return the string with the first

//letters all capitalized.

}//By Jay: END capitalize(str: String): static final String

/\*\*

\* Prompts anotherStock and reads as uppercase.

\*/

public static void promptAnotherStock()

{

System.out.printf("%nEnter \'Y\' to begin stock cost calculations or \'N\' to exit: ");

anotherStock = input.nextLine().toUpperCase().charAt(0);

}//By Jay: END promptAnotherStock(): static void

/\*\*

\* Prompts and reads input.hasNextInt() as the argument for call to

\* validateInteger() which is assigned to shares.

\*/

public static void setShares()

{

System.out.printf("%nHow many shares do you want to purchase? ");

shares = validateInteger(input.hasNextInt());

}//By Avery: END setShares(): static void

/\*\*

\* Prompts and reads input.hasNextDouble() as the argument

\* for call to validateDouble() which is assigned to sharePrice.

\*/

public static void setSharePrice()

{

System.out.printf("%nWhat is the price per share? ");

sharePrice = validateDouble(input.hasNextDouble());

}//By Avery: END setSharePrice(): static void

/\*\*

\* Calculates and returns the value of the stock cost.

\*/

public static double calcStockCost()

{

return shares \* sharePrice;

}//By Avery: END calcStockCost(): static double

/\*\*

\* Prompts for whether there is an online trade and

\* returns it uppercased from the keyboard.

\*/

public static char promptOnlineTrade()

{

System.out.printf("%nIs this an online trade? Enter \'Y\' or \'N\': ");

return input.nextLine().toUpperCase().charAt(0);

}//By Avery: END promptOnlineTrade(): static char

/\*\*

\* Prompts for whether there is a broker assisted trade

\* and returns it uppercased from the keyboard.

\*/

public static char promptBrokerAssisted()

{

System.out.printf("%nIs this a broker assisted trade? Enter \'Y\' or \'N\': ");

return input.nextLine().toUpperCase().charAt(0);

}//By Avery: END promptBrokerAssisted(): static char

/\*\*

\* Prompts and reads input.hasNextDouble() as the argument for call to

\* validateDouble() which is assigned to commissionRate.

\* Return the commission from the calculation of stockCost and commissionRate.

\*/

public static double calcCommission(double stockCost)

{

double commissionRate = 0.0;//NEW

System.out.printf("%nEnter the commission rate as a decimal: ");

commissionRate = validateDouble(input.hasNextDouble());

return stockCost \* commissionRate;

}//By Jay: END calcCommission(stockCost: double): static double

/\*\*

\* Prompts to calculate for another stock. Assign input uppercased to the correct field.

\*/

public static void repromptAnotherStock()

{

System.out.printf("%nEnter 'Y' to calculate the cost for another stock or 'N' to exit: ");

anotherStock = input.nextLine().toUpperCase().charAt(0);

}//By Braden: END repromptAnotherStock(): static void

/\*\*

\* Prompts to continue with another trader. Assign input uppercased to the correct field.

\*/

public static void repromptAnotherTrader()

{

System.out.printf("%nEnter \'Y\' to continue with another trader or \'N\' to exit: ");

anotherTrader = input.nextLine().toUpperCase().charAt(0);

}//By Braden: END repromptAnotherTrader(): static void

/\*\*

\* Formats the final output using String.format() and returns the output per the final output specifications.

\*/

public static String formatFinalOutput(String customerName, double totalStockCost, double totalOnlineFees,

double totalCommissions, double totalCost)

{

Calendar dateTime = Calendar.getInstance(); //Object to obtain the system's date.

String date = String.format("%1$TB %1$td, %1$tY", dateTime); //Object to format the system's date.

return String.format("%nYEE-TRADE, INC."

+ "%nTOTAL COST OF INTENDED STOCK PURCHASES "

+ "%nFOR %s"

+ "%nAS OF %s"

+ "%n%nTotal Stock Cost: $%,14.2f"

+ "%nTotal Online Fees: %14s"

+ "%nTotal Commissions: %14s"

+ "%n%nTOTAL COST: $%,14.2f%n", customerName,

date, totalStockCost, String.format("%,.2f", totalOnlineFees),

String.format("%,.2f", totalCommissions), totalCost);

}//By Braden: END formatFinalOutput(customerName: String, totalStockCost, totalOnlineFees,

//totalCommissions, totalCost: double): static String

/\*\*

\* Prints the thank you message.

\*/

public static void printThankYouMessage()

{

System.out.printf("%nThank you for using Yee-Trade\'s stock cost calculator!%n");

}//By Braden: END printThankYouMessage(): static void

/\*\*

\* While the parameter variable is not valid clear the buffer using next().

\* Read into the parameter variable using Scanner’s hasNextInt().

\* Return the integer from the keyboard.

\*/

public static final int validateInteger(boolean validInteger)

{

while(!validInteger)

{

input.next();

System.out.printf("%nNot an integer! Enter a valid integer: ");

validInteger = input.hasNextInt();

}//By Avery: END while !validInteger

return input.nextInt();

}//By Braden: END validateInteger(validInteger: boolean): static final int

/\*\*

\* While the parameter variable is not valid clear the buffer using next().

\* Read into the parameter variable using Scanner’s hasNextDouble().

\* Return the double from the keyboard.

\*/

public static final double validateDouble(boolean validDouble)

{

while(!validDouble)

{

input.next();

System.out.printf("%nNot a floating-point! Enter a valid float: ");

validDouble = input.hasNextDouble();

}//By Braden: END while !validDouble

return input.nextDouble();

}//By Braden: END validateDouble(validDouble: boolean): static final double

}//By Jay: END APPLICATION CLASS BhaktaBonnerScarsella003PA2

/\*

YEE-TRADE, INC. The Wild West of Electronic Trading

Welcome to Yee-Trade's stock cost calculator.

Ready to generate a stock cost report? Enter 'Y' or 'N' to exit: y

What is your name? haw#ye pierce

haw#ye pierce is not alphabetic.

What is your name? hawkeye pierce

Enter 'Y' to begin stock cost calculations or 'N' to exit: y

How many shares do you want to purchase? !000

Not an integer! Enter a valid integer: 1000

What is the price per share? 15

Is this an online trade? Enter 'Y' or 'N': y

Enter 'Y' to calculate the cost for another stock or 'N' to exit: y

How many shares do you want to purchase? 500

What is the price per share? 52

Is this an online trade? Enter 'Y' or 'N': n

Is this a broker assisted trade? Enter 'Y' or 'N': y

Enter the commission rate as a decimal: .02

Enter 'Y' to calculate the cost for another stock or 'N' to exit: n

Enter 'Y' to continue with another trader or 'N' to exit: y

What is your name? Mannie j. quinn

Enter 'Y' to begin stock cost calculations or 'N' to exit: y

How many shares do you want to purchase? 300

What is the price per share? l0.50

Not a floating-point! Enter a valid float: 10.50

Is this an online trade? Enter 'Y' or 'N': y

Enter 'Y' to calculate the cost for another stock or 'N' to exit: n

Enter 'Y' to continue with another trader or 'N' to exit: n

STOCK COST REPORT

YEE-TRADE, INC.

TOTAL COST OF INTENDED STOCK PURCHASES

FOR Hawkeye Pierce

AS OF OCTOBER 22, 2023

Total Stock Cost: $ 41,000.00

Total Online Fees: 5.95

Total Commissions: 520.00

TOTAL COST: $ 41,525.95

YEE-TRADE, INC.

TOTAL COST OF INTENDED STOCK PURCHASES

FOR Mannie J. Quinn

AS OF OCTOBER 22, 2023

Total Stock Cost: $ 3,150.00

Total Online Fees: 5.95

Total Commissions: 0.00

TOTAL COST: $ 3,155.95

Thank you for using Yee-Trade's stock cost calculator!

\*/