// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 3 Problem 3.12

// Date: 01/14/2023

// Language: Java

// Description: Exercise 3.12

// - Invoice Class

// ------------------------------------------

public class Invoice {

private String itemNumber;

private String itemDesc;

private int itemQuantity;

private int itemPrice;

// Constructors

public Invoice() {}

public Invoice(String number, String desc,

int quantity, int price) {

itemNumber = number;

if (quantity >= 0) { itemQuantity = quantity; }

if (price >= 0) { itemPrice = price; }

itemDesc = desc;

}

// Get methods

public double getInvoiceAmount() { return itemQuantity \* itemPrice; }

public String getItemNumber() { return itemNumber; }

public String getItemDesc() { return itemDesc; }

public int getItemQuantity() { return itemQuantity; }

public int getItemPrice() { return itemPrice; }

// Set methods

public void setItemNumber(String number) { itemNumber = number; }

public void setItemDesc(String desc) { itemDesc = desc; }

public void setItemQuantity(int quantity) { itemQuantity = quantity; }

public void setItemPrice(int price) { itemPrice = price; }

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 3 Problem 3.12

// Date: 01/14/2023

// Language: Java

// Description: Exercise 3.12

// - Invoice Driver Class

// ------------------------------------------

public class InvoiceTest {

public static void main(String[] args) throws Exception {

// Adding newline for format in terminal

System.out.println("\n");

// Instantiate an invoice

Invoice invoice1 = new Invoice("5642A8E", "Power Washer",

7, 542);

// Print intital configuration

String total = String.format("%.2f", invoice1.getInvoiceAmount());

String output = "The invoice is for a " + invoice1.getItemDesc() +

" with part number " + invoice1.getItemNumber() +

" and price $" + invoice1.getItemPrice() +

" with a total of " + invoice1.getItemQuantity() + " in the order.\n" +

" The total price is $" + total + "\n\n";

System.out.println(output);

// Alter the instance variables

invoice1.setItemNumber("68464C");

invoice1.setItemDesc("Toothbrush");

invoice1.setItemPrice(5);

invoice1.setItemQuantity(2);

// Print updated configuration

total = String.format("%.2f", invoice1.getInvoiceAmount());

output = "The invoice is for a " + invoice1.getItemDesc() +

" with part number " + invoice1.getItemNumber() +

" and price $" + invoice1.getItemPrice() +

" with a total of " + invoice1.getItemQuantity() + " in the order.\n" +

" The total price is $" + total + "\n\n";

System.out.println(output);

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 3 Problem 3.15

// Date: 01/28/2023

// Language: Java

// Description: Exercise 3.15

// - Remove duplicate code (Driver class)

// ------------------------------------------

public class AccountDriver {

public static void main(String[] args) throws Exception {

// Instantiate both account objects

Account account1 = new Account("Jane Green", 500.00);

Account account2 = new Account("John Blue", 200.00);

// Display initial balances

account1.displayBalance();

account2.displayBalance();

// Make a deposit to object account1 + formatting

System.out.println();

account1.makeDeposit();

System.out.println();

// Display updated balances

account1.displayBalance();

account2.displayBalance();

// Make a deposit to object account2 + formatting

System.out.println();

account2.makeDeposit();

System.out.println();

// Display updated balances

account1.displayBalance();

account2.displayBalance();

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 3 Problem 3.15

// Date: 01/28/2023

// Language: Java

// Description: Exercise 3.15

// - Remove duplicate code

// ------------------------------------------

import java.util.Scanner;

public class Account {

private String name;

private double balance;

// Constructor

public Account(String name, double balance) {

this.name = name;

if (balance > 0) { this.balance = balance; }

}

// Set methods

public void setName(String name) { this.name = name; }

public void deposit(double amount) {

if (amount > 0.0) { this.balance = this.balance + amount; }

}

public void makeDeposit() {

Scanner input = new Scanner(System.in);

System.out.printf("Enter deposit amount for %s: ", name);

double amount = input.nextDouble();

System.out.printf("Adding %.2f to %s's account.\n", amount, name);

this.deposit(amount);

}

// Get methods

public String getName() { return this.name; }

public double getBalance() { return balance; }

public void displayBalance() {

String result = String.format("%s's balance: $%.2f",

name, balance);

System.out.println(result);

}

}

A picture containing graphical user interface

Description automatically generatedText

Description automatically generated

Figure 1: Exercise 3.12

Figure 2: Exercise 3.15