***Java Programming***

***section-3.2 practice***

**Javabank and Bikeproject :**

import java.util.ArrayList;

import java.util.Random;

// Define abstract BankAccount class

abstract class AbstractBankAccount {

protected String name;

protected int accountNum;

protected double balance;

public AbstractBankAccount(String name, int accountNum, double balance) {

this.name = name;

this.accountNum = accountNum;

this.balance = balance;

}

public abstract void display();

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

}

// Define CreditAccount class

class CreditAccount extends AbstractBankAccount {

public CreditAccount(String name, int accountNum, double balance) {

super(name, accountNum, balance);

}

@Override

public void display() {

System.out.println("Credit Account - Name: " + name + ", Account Number: " + accountNum + ", Balance: $" + balance);

}

}

// Define BankAccountManager class

class BankAccountManager {

static ArrayList<AbstractBankAccount> myAccounts = new ArrayList<AbstractBankAccount>();

public static void addAccount(AbstractBankAccount account) {

myAccounts.add(account);

}

public static void displayAccounts() {

for (AbstractBankAccount account : myAccounts) {

account.display();

}

}

public static void updateBalance(int index, double deposit) {

if (index >= 0 && index < myAccounts.size()) {

AbstractBankAccount account = myAccounts.get(index);

account.setBalance(account.getBalance() + deposit);

}

}

}

// Define Bike superclass

abstract class Bike {

@Override

public abstract String toString();

}

// Define MountainBike class

class MountainBike extends Bike {

@Override

public String toString() {

return "Mountain Bike";

}

}

// Define RoadBike class

class RoadBike extends Bike {

@Override

public String toString() {

return "Road Bike";

}

}

// Define BikeList class with main method

public class BikeList {

public static void main(String[] args) {

// Bank account operations

BankAccountManager.addAccount(new CreditAccount("Alice", 1001, 500.00));

BankAccountManager.addAccount(new CreditAccount("Bob", 1002, 300.00));

BankAccountManager.updateBalance(0, 150.00);

System.out.println("Bank Accounts:");

BankAccountManager.displayAccounts();

// Bike operations

ArrayList<Bike> bikes = new ArrayList<Bike>();

fillArray(bikes);

System.out.println("\nBike Stock:");

displayStock(bikes);

System.out.println("\nBike Numbers:");

displayBikeNumbers(bikes);

}

public static void fillArray(ArrayList<Bike> bikes) {

Random rand = new Random();

for (int i = 0; i < 10; i++) {

if (rand.nextInt(2) == 0) {

bikes.add(new MountainBike());

} else {

bikes.add(new RoadBike());

}

}

}

public static void displayStock(ArrayList<Bike> bikes) {

for (Bike bike : bikes) {

System.out.println(bike);

}

}

public static int calculateStock(ArrayList<Bike> bikes) {

int bikesSold = 0;

for (Bike bike : bikes) {

if (bike instanceof MountainBike) {

bikesSold++;

}

}

return bikesSold;

}

public static void displayBikeNumbers(ArrayList<Bike> bikes) {

int mb = calculateStock(bikes);

int rb = bikes.size() - mb;

System.out.println("Stock Levels");

System.out.println("We have " + mb + " Mountain Bikes in stock");

System.out.println("We have " + rb + " Road Bikes in stock");

}

}

**Set and a List:**

import java.util.HashSet;

import java.util.Set;

public class CountryExample {

public static void main(String[] args) {

Set<String> countriesSet = new HashSet<>();

// Adding countries to the set

countriesSet.add("USA");

countriesSet.add("Canada");

countriesSet.add("Mexico");

countriesSet.add("Germany");

countriesSet.add("France");

countriesSet.add("Italy");

countriesSet.add("Canada"); // Duplicate entry

// Display the set

System.out.println("Countries set: " + countriesSet);

}

}