Review 2

import java.util.ArrayList;

public class Review1 {

public static void main(String[] args) {

int[] numbers = new int[3];

ArrayList<String> nameList = new ArrayList<String>();

BankAccount myAccount = CreateAccount(55);

System.out.println("CreateAccount");

System.out.println(myAccount.getBalance());

BankAccount[] myAcc = CreateArrayAcc(33, 44, 55);

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

for(int i = 0; i < myAcc.length; i++){

System.out.println(myAcc[i].getBalance());}

nameList.add("John");

nameList.add("Bill");

nameList.add("Carl");

System.out.println("\nArrayList Array");

for(int i = 0; i < nameList.size(); i++){

System.out.println(nameList.get(i));}

ArrayList<BankAccount> myAccountList = new ArrayList<BankAccount>();

myAccountList.add(new BankAccount(101));

myAccountList.add(new BankAccount("51"));

myAccountList.get(0).setBalance(77.32);

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

for(int i = 0; i <myAccountList.size(); i++){

System.out.println(myAccountList.get(i).getBalance());}

BankAccount[] myAccounts = new BankAccount[2];

//An old way to handle everything

//myAccounts[0] = new BankAccount(100);

//myAccounts[1] = new BankAccount("50");

//A cleaner way to handle the accounts

for(int i = 0; i < myAccounts.length; i++){

myAccounts[i] = new BankAccount();}

myAccounts[0].setBalance(100.0);

myAccounts[1].setBalance("50");

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

for(int i = 0; i < myAccounts.length; i++){

System.out.println(myAccounts[i].getBalance());} }

public static BankAccount CreateAccount(double startBalance){

return new BankAccount(startBalance); }

public static BankAccount[] CreateArrayAcc(int a, int b, int c){

BankAccount[] myBA = newBankAccount[3];

myBA[0] = new BankAccount(a);

myBA[1] = new BankAccount(b);

myBA[2] = new BankAccount(c);

return myBA; } }

public class BankAccount {

private double balance;

public BankAccount() {

balance = 0.0; }

public BankAccount(double startBalance) {

balance = startBalance; }

public BankAccount(String startBalance) {

balance = Double.parseDouble(startBalance); }

public void deposit(double amount) {

balance += amount; }

public void withdrawl(double amount) {

balance -= amount; }

public void withdrawl(String str) {

balance -= Double.parseDouble(str); }

public void setBalance(double b) {

balance = b; }

public void setBalance(String b) {

balance = Double.parseDouble(b);}

public double getBalance() {

return balance;} }

Project 1

import java.io.\*;

import java.util.Scanner;

public class TxstateSalary {

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

String firstName, lastName, fullName, answer, lineread;

int monthlySalary, monthsOfPayment;

char ans;

Scanner keyboard = new Scanner(System.in);

//Welcomes the user to the application

System.out.println("Welcome to the Texas State Yearly Salary Calculator");

//Asking for the users first and last name

System.out.println("First, we'll need your name.");

System.out.print("Please enter your first name: ");

firstName = keyboard.nextLine();

System.out.print("Please enter your last name: ");

lastName = keyboard.nextLine();

//Combines the first and last name to consolidate variables

fullName = firstName + " " + lastName;

//Asking for the users monthly information

System.out.println("Now, we'll need your monthly information.");

System.out.print("Please enter your monthly salary: ");

monthlySalary = keyboard.nextInt();

keyboard.nextLine();

System.out.print("Please enter the number of payment months: ");

monthsOfPayment = keyboard.nextInt();

keyboard.nextLine();

//Displays the Yearly Salary

System.out.println("");

System.out.println("Your yearly salary is: " + YearlySalary(fullName, monthlySalary, monthsOfPayment));

System.out.printf("");

//Asks the user whether they would like to read from the file

do{

System.out.printf("Would you like to read from file? (y/n) ");

answer = keyboard.nextLine();

answer = answer.toLowerCase();

ans = answer.charAt(0);

if(ans == 'n')

System.exit(0);

}while(ans != 'y');

//reading the file. This area only accessible if replied y or Y

File file = new File("./yearlySalary.txt");

Scanner inputFile = new Scanner(file);

while (inputFile.hasNext()){

lineread = inputFile.nextLine();

System.out.println(lineread); }

inputFile.close(); }

//Method to find the YearlySalary

public static double YearlySalary(String fullName, int monthlySal, int monthsOfPay) throws IOException{

double yearlySalary;

File filename = new File("./yearlySalary.txt");

FileWriter fwriter = new FileWriter(filename, true);

PrintWriter outputFile = new PrintWriter(fwriter);

yearlySalary = monthlySal \* monthsOfPay;

//Adding the header if the file doesn't exist

if(filename.length() == 0){

outputFile.println("Consumer Name\tMonthly Salary\tMonths of Pay\tYearly Salary"); }

outputFile.println(fullName + "\t" + monthlySal+ "\t" + monthsOfPay + "\t" + yearlySalary);

outputFile.close();

return yearlySalary; } }

public class TestRectangle {

public static void main(String[] args) {

Rectangle mLivingRoom = new Rectangle();

String name = new String("John");

mLivingRoom.setLength(5.6);

mLivingRoom.setWidth(4.2);

System.out.println("Has an area of: " + mLivingRoom.getArea() + ". " +

"The length is: " + mLivingRoom.getLength() + " and a width of " +

"" + mLivingRoom.getWidth());

Rectangle myRoom = new Rectangle(6.0, 5.0);

System.out.println("Has an area of: " + myRoom.getArea() + ". " +

"The length is: " + myRoom.getLength() + " and a width of " +

"" + myRoom.getWidth());} }

import javax.swing.\*;

public class RoomAreas {

public static void main(String[] args) {

double number;

double totalArea;

String input;

Rectangle kitchen = new Rectangle();

Rectangle bedroom = new Rectangle();

Rectangle den = new Rectangle();

input = JOptionPane.showInputDialog("What is the kitchen's length?");

number = Double.parseDouble(input);

kitchen.setLength(number);

input = JOptionPane.showInputDialog("What is the kitchen's width?");

number = Double.parseDouble(input);

kitchen.setWidth(number);

input = JOptionPane.showInputDialog("What is the bedroom's length?");

number = Double.parseDouble(input);

bedroom.setLength(number);

input = JOptionPane.showInputDialog("What is the bedroom's width?");

number = Double.parseDouble(input);

bedroom.setWidth(number);

input = JOptionPane.showInputDialog("What is the den's length?");

number = Double.parseDouble(input);

den.setLength(number);

input = JOptionPane.showInputDialog("What is the den's width?");

number = Double.parseDouble(input);

den.setWidth(number);

totalArea = kitchen.getArea() + bedroom.getArea() + den.getArea();

JOptionPane.showMessageDialog(null, "The total area of the apartment is " + totalArea);} }

public class Rectangle {

private double length, width;

public Rectangle() {

length = 1.0;

width = 1.0;}

public Rectangle(double len, double w) {

length = len;

width = w;}

public double getLength() {

return length;}

public void setLength(double len) {

length = len;}

public double getWidth() {

return width;}

public void setWidth(double w) {

width = w;}

public double getArea() {

return width \* length;} }

Project 2

public class TestStudent {

public static void main(String[] args) {

String name, gender, race, id, university, country;

Scanner keyboard = new Scanner(System.in);

//Student[] arrayStud = new Student[10];

// for(int i = 0; i < arrayStud.length; i++) {

System.out.println("Test 6 Student Full Constructor");

System.out.println("Please fill in the required information to test.");

System.out.print("Name: ");

name = keyboard.nextLine();

// arrayStud[i] = new Student(name, gender, race, id, university, country);

Student myStudent2 = new Student(name, gender, race, id, university, country);

// System.out.println(i);}

// for(int i = 0; i < arrayStud.length; i++){

// System.out.println(arrayStud[i].getName());

// System.out.println(arrayStud[i].getUniversity());}

System.out.println("Displaying Updated Student information");

System.out.println("Name: " + myStudent2.getName());

System.out.println("Gender: " + myStudent2.getGender());

System.out.println("Race: " + myStudent2.getRace());

System.out.println("ID: " + myStudent2.getId());

System.out.println("University: " + myStudent2.getUniversity());

System.out.println("Country: " + myStudent2.getCountry());

System.out.println("Test 8 Student Walk function");

myStudent2.walk();

System.out.println("");

System.out.println("Test 9 Student takeCourses function");

myStudent2.takeCourses();

System.out.println("");

System.out.println("Test 10 Student driveCar Interface function");

System.o.print("Please enter the brand of car you drive: "); Sstm.o.println(myStudent2.driveCar(keyboard.nextLine()));}}

public class Person {

private String gender, name, race;

public Person() {

System.out.println("I am a person");}

public Person(String pName, String pGender, String pRace) {

name = pName;

gender = pGender;

race = pRace;}

public String getGender() {

return gender; }

public void setGender(String pGender) {

gender = pGender; }

public String getRace() {

return race;}

public void setRace(String pRace) {

race = pRace; }

public String getName() {

return name; }

public void setName(String pName) {

name = pName; }

public void walk(){

System.out.println("I walk"); }}

public interface Driver {

String driveCar(String car);}

public class Student extends Person implements Driver {

private String university, id, country;

public Student() {

System.out.println("I am a student"); }

public Student(String pName, String pGender, String pRace, String pId, String pUniversity, String pCountry) {

super(pName, pGender, pRace);

id = pId;

university = pUniversity;

country = pCountry; }

public String getUniversity() {

return university; }

public void setUniversity(String pUniversity) {

university = pUniversity; }

public String getCountry() {

return country; }

public void setCountry(String pCountry) {

country = pCountry; }

public String getId() {

return id; }

public void setId(String pId) { id = pId; }

@Override

public void walk(){

System.out.println("I am a student who walks"); }

@Override

public String driveCar(String pCar){

return "I drive a " + pCar; }

public void takeCourses(){

System.out.println("I am taking 3 courses this semester");}}

Exam 1

public class ReadWrite {

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

String firstName, lastName, answer, fileWrite, fileRead, lineRead;

char ans;

int monthlySalary, monthsOfPayment;

double yearlySal;

Scanner keyboard = new Scanner(System.in);

System.out.println("Do you want to create a file and add data? (y/n)");

answer = keyboard.nextLine();

answer = answer.toLowerCase();

ans = answer.charAt(0);

while(ans == 'y'){

System.out.println("What is the name of the file? (Please enter FileName.txt)");

fileWrite = keyboard.nextLine();

System.out.println("Please type the last name");

lastName = keyboard.nextLine();

System.out.println("Please type the first name");

firstName = keyboard.nextLine();

System.out.println("Please type the monthly salary");

monthlySalary = keyboard.nextInt();

keyboard.nextLine();

System.out.println("Please type the number of months of payments");

monthsOfPayment = keyboard.nextInt();

keyboard.nextLine();

yearlySal = YearlySalary(monthlySalary, monthsOfPayment);

File ofile = new File("./" + fileWrite);

FileWriter fWriter = new FileWriter(ofile, true);

PrintWriter outputFile = new PrintWriter(fWriter);

outputFile.println(lastName + "\t" + firstName + "\t" + monthlySalary + "\t" + monthsOfPayment + "\t" +

yearlySal);

outputFile.close();

System.out.println("Do you want to create a file or add data? (y/n)");

answer = keyboard.nextLine();

answer = answer.toLowerCase();

ans = answer.charAt(0); }

System.out.println("Do you want to read data from a file? (y/n)");

answer = keyboard.nextLine();

answer = answer.toLowerCase();

ans = answer.charAt(0);

System.out.println("What is the name of the file? (Please enter FileName.txt)");

fileRead = keyboard.nextLine();

while( ans == 'y'){

try{ File ifile = new File("./" + fileRead);

Scanner inputFile = new Scanner(ifile);

while (inputFile.hasNext()){

lineRead = inputFile.nextLine();

System.out.println(lineRead);}

inputFile.close();

}catch(IOException e) {

//System.err.println("Caught IOException: " + fileRead + " (The system cannot find the file specified)" + e);

System.out.println(e);

//"Caught IOException: " + fileRead + " (The system cannot find the file specified)"); }

System.out.println("Do you want to keep reading data from the file? (y/n)");

answer = keyboard.nextLine();

answer = answer.toLowerCase();

ans = answer.charAt(0);

if(ans == 'y'){

System.out.println("What is the name of the file? (Please enter FileName.txt)");

fileRead = keyboard.nextLine();} } } }

private static double YearlySalary(int monthlySalary, int monthsOfPayment){

double yearlySalary;

yearlySalary = monthlySalary \* monthsOfPayment;

return yearlySalary; }}

Enum Demo

public class EnumDemo {

enum Day {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

public static void main(String[] args) {

Day workDay = Day.Wednesday;

System.out.println(workDay);

System.out.println("The ordinal value for " + Day.Sunday + " is " + Day.Sunday.ordinal());

System.out.println("The ordinal value for " + Day.Saturday + " is " + Day.Saturday.ordinal());

if(Day.Friday.compareTo(Day.Monday) > 0 )

System.out.println(Day.Friday + " is greater than " + Day.Monday);

else

System.out.println(Day.Friday + " is NOT greater than " + Day.Monday);

CarType myCar = CarType.Ferrari; }}

enum CarType { Porsche, Ferrari, Jaguar};

Abstract Class

public abstract class Student {

private String name, idNumber;

private int yearAdmitted;

public Student(String name, String idNumber, int yearAdmitted) {

this.name = name;

this.idNumber = idNumber;

this.yearAdmitted = yearAdmitted; }

@Override

public String toString() {

return "Student{" + "name='" + name + '\'' +

", idNumber='" + idNumber + '\'' +

", yearAdmitted=" + yearAdmitted + '}';}

public abstract int getRemainingHours();}

public class StudentCIS extends Student implements Employee{

public StudentCIS(String name, String idNumber, int yearAdmitted) {

super(name, idNumber, yearAdmitted);}

@Override

public int getRemainingHours() {

return 0;}

@Override

public void displayJob() {

System.out.println("Student job = CIS"); }}

public interface Employee {

void displayJob();}