|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  **Assignment-2** | **Subject:**  **Object Oriented Programming**  **submitted by:**  **Daoud hussain**  (Sp21-bcs-102)  **submitted to:**  **Mam Saneeha AAmir**  **date of submission:**  **March 27 , 2022**  **Course Code:**  **March 27 , 2022** |

Rectangle Class:

public class Rectangle{

private int length, width;

public Rectangle(){

//Default Argument constructor

}

public Rectangle(int l){

if(l>0){

length = l;

}

}

public Rectangle(int l, int w){

if(l>0 && w>0){

length = l;

width = w;

}

}

public void setLength(int l){

if(l>0){

length = l;

}

}

public void setWidth(int w){

if(w>0){

width = w;

}

}

public int getLength(){

return length;

}

public int getWidth(){

return width;

}

public void display(){

System.out.print("Length: " + length + " Width: " + width);

}

//COPY constructor

public Rectangle(Rectangle obj){

this.length = obj.length;

this.width = obj.width;

}

//Equals Method

public Boolean equals(Rectangle obj){

if(obj.length == this.length && obj.width == this.width){

return true;

}

return false;

}

public int calculateArea(){

return length\*width;

}

public Boolean checkSquare(){

if(length == width){

return true;

}

return false;

}

public Rectangle compareArea(Rectangle obj){

if(this.calculateArea() > obj.calculateArea()){

return this;

}

else{

return obj;

}

}

}

Runner Rectangle:

public class Runner{

public static void main(String[] args) {

Rectangle a1 = new Rectangle(); //Setting default values

Rectangle a2 = new Rectangle(122,222); //2 Argument Values

a2.calculateArea();

a2.checkSquare();

a2.display();

a2.compareArea(a1);

a2.checkSquare();

a2.compareArea(a1);

}

}

----------------------------------------

Account Class:

public class Account{

int balance,yearOfOpening;;

String cnic;

public Account(){

//Default Constructor

}

public Account(int a){

if(a>=0){

balance = a;

}

else{

balance = 0;

}

}

public Account(int a, String b, int c){

if(a>=0 && c>=0){

balance = a;

cnic = b;

yearOfOpening = c;

}else{

balance = 0;

yearOfOpening = 0;

}

}

public int withDraw(int withdrawalValue){

if(withdrawalValue>=0){

balance = balance-withdrawalValue;

}

return balance;

}

public int deposit(int depositedValue){

if(depositedValue>=0){

balance = balance+depositedValue;

}

return balance;

}

public void display(){

System.out.printf("Value is: \nBalance is: %d \nCnic is: %s \nYear Of Opening is %d: \n", balance, cnic, yearOfOpening);

}

//copy Constructor

public Account(Account a){

this.balance = a.balance;

this.yearOfOpening = a.yearOfOpening;

this.cnic = a.cnic;

}

//Equals Method

public boolean equals(Account a){

if(this.balance == a.balance && this.yearOfOpening == a.yearOfOpening && this.cnic == a.cnic){

return true;

}

return false;

}

public boolean checkValidCnic(){

if(cnic.length() == 15){

return true;

}

return false;

}

}

Runner Account:

public class Runner{

public static void main(String[] args) {

Account a1 = new Account();

Account a2 = new Account(2000);

Account a3 = new Account(122, "37201-3209291-7", 2002);

a3.withDraw(700);

a3.deposit(100);

a3.display();

a3.checkValidCnic();

}

}

----------------------------------------

Point Class:

public class Point{

int x, y;

public Point(){

//Default Constructor

}

//One-Arg Constructor

public Point(int a){

x = a;

}

//2 argument Constructor

public Point(int a, int b){

x = a;

y = b;

}

public void display(){

System.out.printf("Value of A: %d and Value of B: %d", x, y);

}

//a and b are the points that how far you move in x and y coordinates

public void move(int a, int b){

x = a+x;

y = b+y;

}

public boolean checkOrigin(){

if(x==0 && y==0){

return true;

}

else{

return false;

}

}

//Copy Constructor

public Point(Point p){

this.x = p.x;

this.y = p.y;

}

//Equals Constructor

public boolean equals(Point p){

if(this.x == p.x && this.y == p.y){

return true;

}

return false;

}

public Point AddTwoPoints(Point pa){

int newX = this.x + pa.x;

int newY = this.y + pa.y;

Point newObj = new Point(newY, newY);

return newObj;

}

public Point AddThreePoints(Point pa, Point qa){

int finalX = qa.x + this.x + pa.x;

int finalY = qa.y + this.y + pa.y;

Point Obj = new Point(finalX,finalY);

return Obj;

}

}

Point Runner:

public class Runner{

public static void main(String[] args) {

Point a1 = new Point(); //Setting default values

Point a2 = new Point(-1,3); //2 Argument Values

a2.move(3,4);

a2.checkOrigin();

a2.display();

a2.AddTwoPoints(a1);

a2.AddThreePoints(a1,a2);

}

}

----------------------------------------

Student Class:

public class Student{

int[] Result = new int[5];

String name;

public Student(){

//Default Constructor

}

//One-Arg Constructor

public Student(String n){

if(n!=""){

name = n;

}

}

//Two Arg constructor

public Student(String n, int[] s){

if(n!=""){

name = n;

}

if(s.length>=Result.length){

for (int i=0; i<Result.length-1 ;i++ ) {

Result[i] = s[i];

}

}

else{

for (int i=0; i<s.length-1 ;i++ ) {

Result[i] = s[i];

}

}

}

public double average(){

int sum = 0;

for(int i=0; i<Result.length-1;i++){

sum+=Result[i];

}

return sum/Result.length;

}

public Student(Student s){

this.name = s.name;

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

this.Result[i] = s.Result[i];

}

}

public boolean equals(Student s){

if(this.name == s.name){

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

if(this.Result[i]==s.Result[i]){

return true;

}

}

}

return false;

}

public boolean compareAverage(Student s){

if(this.average() == s.average()){

System.out.print("Haaan");

return true;

}

return false;

}

}

Student Runner:

public class Runner{

public static void main(String[] args) {

int[] arr= {10,23,34,44,53};

Student s1 = new Student();

Student s2 = new Student("Daoud", arr);

s2.average();

s1.compareAverage(s2);

}

}

----------------------------------------

Book Class:

public class Book{

String author = "";

String chapterNames[] = new String[5];

public Book(){

//Default Constructor

}

//2 argument Constructor

public Book(String a, String[] b ){

author = a;

for (int i=0; i<=4 ;i++ ) {

chapterNames[i] = b[i];

}

}

public boolean compareAuthors(Book b){

if(this.author == b.author){

return true;

}

return false;

}

public boolean compareChapters(Book b){

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

if(this.chapterNames[i] == b.chapterNames[i]){

return true;

}

}

return false;

}

}

Book Runner:

public class Runner{

public static void main(String[] args) {

String[] arr= {"Introduction","Sequence","Selection","Repetititon","Functions"};

Book a1 = new Book(); //Setting default values

Book a2 = new Book("Rizwan Rashid", arr); //2 Argument Values

a2.compareAuthors(a1);

a2.compareChapters(a1);

}

}

----------------------------------------

University Class:

public class University{

String[] departments = new String[20];

String uniName, location, rectorName;

public University(){

//Default Constructor

}

//Full-Argument Constructor

public University(String a,String b, String c, String[] s){

if(a!="" && b!="" && c!=""){

uniName = a;

location = b;

rectorName = c;

}

if(s.length < departments.length){

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

departments[i] = s[i];

}

}

else{

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

departments[i] = s[i];

}

}

}

public void setUniName(String un){

if(un!=""){

uniName = un;

}

}

public void setLocation(String l){

if(l!=""){

location = l;

}

}

public void setRectorName(String rn){

if(rn!=""){

rectorName = rn;

}

}

public void setDepartments(String[] s){

if(s.length < departments.length){

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

departments[i] = s[i];

}

}

else{

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

departments[i] = s[i];

}

}

}

public String getUniName(){

return uniName;

}

public String getRectorName(){

return rectorName;

}

public String getLocation(){

return location;

}

public void getDepartments(){

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

if(departments[i] != null){

System.out.print(departments[i] + " ");

}

}

}

//Method to Add a department

public void addADepartment(String abc){

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

if(departments[i] == null){

departments[i] = abc;

break;

}

}

}

//Method to check whether the location provided is in Islamabad or not

public boolean checkIfLocatedInCapital(){

if(location == "Islamabad"){

return true;

}

else{

return false;

}

}

}

University Runner:

public class Runner{

public static void main(String[] args) {

String[] arr= {"CE","CS" ,"Cyber Security","AI","DS"};

University a1 = new University(); //Setting default values

University a3 = new University("COMSATS", "Islamabad", " Fazal Mehmood", arr); //Full Argument Constructor

a3.checkIfLocatedInCapital();

a3.addADepartment("SE");

a3.getDepartments();

}

}

----------------------------------------