A blue and white logo

Description automatically generated

|  |  |
| --- | --- |
| Submitted By | Habib ur Rehman (116) |
| Subject | OOP |
| Assignment | Lab Assignment Inheritance |
| Date | Nov 29th , 2024 |

**Submitted to:**

|  |  |
| --- | --- |
| Moderator | Ms, Sajida Kalsoom |

**Lab Task 01:**

// Public interface Shape

// {

// double getArea();

// }

// Create two classes Circle and Rectangle. Both must implement the interface Shape.

// Note: You can assume appropriate data members for circle and rectangle

interface Shape {

    double getArea();

}

class circle implements Shape {

    private double radius;

    public circle(double radius) {

        this.radius = radius;

    }

    public void setRadius(double radius) {

        this.radius = radius;

    }

    public double getRadius() {

        return radius;

    }

    public double getArea() {

        return radius \* radius \* 3.14;

    }

    @Override

    public String toString() {

        return "circle [radius=" + radius + "]";

    }

}

class Rectangle implements Shape {

    double length;

    double width;

    public Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    public Rectangle() {

    }

    public double getLength() {

        return length;

    }

    public void setLength(double length) {

        this.length = length;

    }

    public double getWidth() {

        return width;

    }

    public void setWidth(double width) {

        this.width = width;

    }

    @Override

    public String toString() {

        return "Rectangle [length=" + length + ", width=" + width + "]";

    }

    public double getArea() {

        return length \* width;

    }

}

public class labtask1 {

    public static void main(String[] args) {

        circle circle = new circle(10);

        Rectangle rectangle = new Rectangle(3, 2);

        System.out.println("Area of the circle: " + circle.getArea());

        System.out.println("Area of the rectangle: " + rectangle.getArea());

        System.out.println(circle);

        System.out.println(rectangle);

    }

}

**Lab Task 02:**

// Payable:

// Double getPaymenyAmount();

interface Payable {

    double getpaymentAmount();

}

class invoice implements Payable {

    private String partNumber;

    private String partDesc;

    private int quantity;

    private double priceperunit;

    public invoice() {

    }

    public invoice(String partNumber, String partDesc, int quantity, double priceperunit) {

        this.partNumber = partNumber;

        this.partDesc = partDesc;

        this.quantity = quantity;

        this.priceperunit = priceperunit;

    }

    public String getPartNumber() {

        return partNumber;

    }

    public void setPartNumber(String partNumber) {

        this.partNumber = partNumber;

    }

    public String getPartDesc() {

        return partDesc;

    }

    public void setPartDesc(String partDesc) {

        this.partDesc = partDesc;

    }

    public int getQuantity() {

        return quantity;

    }

    public void setQuantity(int quantity) {

        this.quantity = quantity;

    }

    public double getPriceperunit() {

        return priceperunit;

    }

    public void setPriceperunit(double priceperunit) {

        this.priceperunit = priceperunit;

    }

    public double getpaymentAmount() {

        return quantity \* priceperunit;

    }

    @Override

    public String toString() {

        return "invoice [partNumber=" + partNumber + ", partDesc=" + partDesc + ", quantity=" + quantity

                + ", priceperunit="

                + priceperunit + "]";

    }

}

abstract class employee implements Payable {

    private String Fname;

    private String Lname;

    private String SSN;

    public employee() {

    }

    public employee(String fname, String lname, String sSN) {

        Fname = fname;

        Lname = lname;

        SSN = sSN;

    }

    public String getFname() {

        return Fname;

    }

    public void setFname(String fname) {

        Fname = fname;

    }

    public String getLname() {

        return Lname;

    }

    public void setLname(String lname) {

        Lname = lname;

    }

    public String getSSN() {

        return SSN;

    }

    public void setSSN(String sSN) {

        SSN = sSN;

    }

    public abstract double getpaymentAmount();

    @Override

    public String toString() {

        return "employee [Fname=" + Fname + ", Lname=" + Lname + ", SSN=" + SSN + "]";

    }

}

class salriedEmploye extends employee {

    private double weeksalary;

    public salriedEmploye(String fname, String lname, String sSN, double weeksalary) {

        super(fname, lname, sSN);

        this.weeksalary = weeksalary;

    }

    public double getWeeksalary() {

        return weeksalary;

    }

    public void setWeeksalary(double weeksalary) {

        this.weeksalary = weeksalary;

    }

    public double getpaymentAmount() {

        return weeksalary;

    }

    @Override

    public String toString() {

        return super.toString() + "salriedEmploye [weeksalary=" + weeksalary + "]";

    }

}

public class labtask2 {

    public static void main(String[] args) {

        Payable p1 = new invoice("ddd", "2rd", 23, 12);

        System.out.println(p1.getpaymentAmount());

        System.out.println(p1.toString());

        p1 = new salriedEmploye("habib", "ur rehman", "2rhj", 232);

        System.out.println(p1.toString());

        System.out.println(p1.getpaymentAmount());

    }

}

**Lab Task 03:**

class Computer{

//      contains data members of wordsize(in bits), memorysize (in megabytes),

// storagesize (in megabytes) and speed (in megahertz)

protected int wordSize;

protected int memorySize;

protected int storageSize;

protected int megahertz;

Computer(){

}

Computer(int w,int m,int s,int mega){

    this.wordSize=w;

    this.memorySize=m;

    this.storageSize=s;

    this.megahertz=mega;

}

public void setWordsize(int W){

    this.wordSize=W;

}

public void setmemorySize(int M){

    this.memorySize=M;

}

public void setStorageSize(int S){

    this.storageSize=S;

}

public void setmegaHertz(int M){

    this.megahertz=M;

}

public int getWordSize(){

    return wordSize;

}

public int getmemorySize(){

    return memorySize;

}

public int getStorageSize(){

    return storageSize;

}

public int getmegaHerz(){

    return megahertz;

}

public String toString(){

    return "WordSize: "+wordSize+ "bits" +", memory Size: "+memorySize+" in mbs"+" ,Storage Size:  "+storageSize+ " and mega hertz:  "+megahertz;

}

}

class Laptop  extends Computer{

    //  specifies the object’s length, width, height, and weight.

   protected double length;

   protected double height;

   protected double weight;

   Laptop(){}

   Laptop(int w,int m,int s,int mega,double length,double height,double weight){

    super(w,m,s,mega);

    this.length=length;

    this.height=height;

    this.weight=weight;

   }

   public void setLength(int l){

    this.length=l;

   }

   public void setheight(int h){

    this.height=h;

   }

   public void setweight(int w){

    this.weight=w;

   }

public double getHeight(){

    return height;

}

public double getWeight(){

    return weight;

}

public double getLength(){

    return length;

}

public String toString(){

    return super.toString()+" , length is: "+length+" , Height is "+height+" , weight is:"+weight;

}

}

public class LabTask3 {

    public static void main(String[] args) {

        Computer C1=new Computer(12,34,55,44);

        System.out.println(C1.toString());

        Laptop L1=new Laptop(12,34,56,77,88,44,55);

        System.out.println(L1.toString());

    }

}

// “InventoryItem”  . Each inventory item has a name and a

// unique ID number:

// class InventoryItem

// {

// private String name;

// private int uniqueItemID;

// }

interface compare {

    boolean compareObjects(Object o);

}

class InventoryItem implements compare {

    private String name;

    private int uniqueItemID;

    public InventoryItem() {

    }

    public InventoryItem(String name, int uniqueItemID) {

        this.name = name;

        this.uniqueItemID = uniqueItemID;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getUniqueItemID() {

        return uniqueItemID;

    }

    public void setUniqueItemID(int uniqueItemID) {

        this.uniqueItemID = uniqueItemID;

    }

    public boolean compareObjects(Object o) {

        if (o instanceof InventoryItem) {

            InventoryItem I1 = (InventoryItem) o;

            if (this.name.equals(I1.name) && this.uniqueItemID == I1.uniqueItemID) {

                return true;

            } else

                return false;

        }

        return false;

    }

    @Override

    public String toString() {

        return "InventoryItem [name=" + name + ", uniqueItemID=" + uniqueItemID + "]";

    }

}

public class labtask3 {

    public static void main(String[] args) {

        InventoryItem c1 = new InventoryItem("hb", 123);

        System.out.println(c1);

        InventoryItem c2 = new InventoryItem("hb", 123);

        System.out.println(" items equal? " + c1.compareObjects(c2));

    }

}

**Lab Task04**

// interface Enumeration

// {

// // return true if a value exists in the next index

// public boolean hasNext(int index);

// // returns the next element in the collection as an Object

// public Object getNext(int index);

// }

// //NameCollection implements a collection of names using  a simple array.

// class NameCollection

// {

// String[] names = new String[100];

// }

interface Enumeration {

    public boolean hasNext(int index);

    public Object getNext(int index);

}

class NameCollection implements Enumeration {

    String[] names = new String[100];

    public NameCollection() {

    }

    public NameCollection(String[] names) {

        this.names = names;

    }

    public boolean hasNext(int index) {

        if (index < names.length && names[index] != null) {

            return true;

        }

        return false;

    }

    public Object getNext(int index) {

        if (hasNext(index)) {

            return names[index];

        } else {

            return null;

        }

    }

}

public class labtask4 {

    public static void main(String[] args) {

        String[] names = { "jamil", "majid", "ali", "habib", "blocks" };

        NameCollection nameCollection = new NameCollection(names);

        System.out.println(nameCollection.hasNext(6));

        System.out.println(nameCollection.getNext(6));

        int in = 0;

        while (nameCollection.hasNext(in)) {

            System.out.println(nameCollection.getNext(in));

            in++;

        }

    }

}