***1.mutliple inheritance***

interface AnimalEat {

void eat();

}

interface AnimalTravel {

void travel();

}

class Animal implements AnimalEat, AnimalTravel {

public void eat()

{

System.out.println("Animal is eating");

}

public void travel() {

System.out.println("Animal is travelling");

}

}

class Main

{

public static void main (String[] args) {

Animal a1=new Animal();

a1.eat();

a1.travel();

}

}

Output:

Animal is eating

Animal is travelling

2.multilevel inheritance

interface Car

{

int speed=60;

public void distanceTravelled();

}

interface Bus

{

int distance=100;

void speed();

}

class Vehicle implements Car,Bus

{

int distanceTravelled;

int averageSpeed;

public void distanceTravelled()

{

distanceTravelled=speed\*distance;

System.out.println("Total Distance Travelled is : "+distanceTravelled);

}

public void speed()

{

int averageSpeed=distanceTravelled/speed;

System.out.println("Average Speed maintained is : "+averageSpeed);

}

}

class Main

{

public static void main(String[] args)

{

Vehicle v1=new Vehicle();

v1.distanceTravelled();

v1.speed();

}

}

Output:

Total Distance Travelled is : 6000

Average Speed maintained is : 100

***3.Multilevel inheritance***

import java.util.\*;

interface sal

{

final double tax=0.05;

abstract void cal\_sal();

}

class Employee

{

int eid;

String name;

double bs;

void accept()

{

Scanner sc=new Scanner(System.in);

System.out.println("enter basic salary:");

bs=sc.nextInt();

System.out.println("enter name");

name=sc.next();

System.out.println("enter id");

eid=sc.nextInt();

}

void display()

{

System.out.println("eid:"+eid);

System.out.println("name:"+name);

System.out.println("basic salary:"+bs);

}

}

class emp\_sal extends Employee implements sal

{

double hra,ta,da,gs;

public void cal\_sal()

{

hra=bs\*0.40;

da=bs\*0.40;

ta=bs\*0.40;

gs=(hra+ta+da+bs)-(bs\*tax);

System.out.println("Hra"+hra);

System.out.println("Ta:"+ta);

System.out.println("DA:"+da);

System.out.println("Tax:"+tax);

System.out.println("GS:"+gs);

}

}

public class Main {

public static void main(String[] args)

{

emp\_sal e1=new emp\_sal();

e1.accept();

e1.display();

e1.cal\_sal();

}

}

***Output:***

enter basic salary:

2000

enter name

ritu

enter id

101

eid:101

name:ritu

basic salary:2000.0

Hra800.0

Ta:800.0

DA:800.0

Tax:0.05

GS:4300.0

***4.multilevel interface (student mark display ane calculate total ,percentage ,grade)***

import java.util.\*;

interface marks

{

abstract void cal\_grade();

abstract void cal\_total\_avg();

}

class student

{

int m1,m2,m3,total;

double per;

public void getmark()

{

Scanner sc=new Scanner (System.in);

System.out.println("enter the marks:");

System.out.println("subject 1:");

m1=sc.nextInt();

System.out.println("subject 2:");

m2=sc.nextInt();

System.out.println("subject 3:");

m3=sc.nextInt();

}

public void showmarks()

{

System.out.println("mark of sub 1:" +m1);

System.out.println("mark of sub 2:" +m2);

System.out.println("mark of sub 3:" +m3);

}

}

class result extends student implements marks{

public void cal\_total\_avg()

{

total=m1+m2+m3;

System.out.println("total is:"+total);

per=(total / 300.0) \* 100;

System.out.println("percentage is:"+per);

}

public void cal\_grade()

{

int avg;

System.out.print("The student Grade is: ");

if(per>=80)

{

System.out.print("A");

}

else if(per>=60 && per<80)

{

System.out.print("B");

}

else if(per>=40 && per<60)

{

System.out.print("C");

}

else

{

System.out.print("D");

}

}

}

public class Main

{

public static void main (String[] args) {

result r1=new result();

r1.getmark();

r1.showmarks();

r1.cal\_total\_avg();

r1.cal\_grade();

}

}

***Output:***

enter the marks:

subject 1:

89

subject 2:

67

subject 3:

98

mark of sub 1:89

mark of sub 2:67

mark of sub 3:98

total is:254

percentage is:84.66666666666667

The student Grade is: A

***5. write class number\_Genrator which has 1 int variable num write accept method which accept the numbers using scanner***

***\* write method print\_Table which will be override in another class;***

***\* write the class Math\_Operations which uses number\_genaerator and implement print table method in this class.***

import java.util.Scanner;

class Number\_Generator {

protected int num;

public void accept() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

num = scanner.nextInt();

}

public void print\_Table() {

System.out.println("Table for " + num + ":");

for (int i = 1; i <= 10; i++) {

System.out.println(num + " x " + i + " = " + (num \* i));

}

}

}

class Math\_Operations extends Number\_Generator {

@Override

public void print\_Table() {

super.print\_Table();

System.out.println("\nSquare: " + (num \* num));

System.out.println("Cube: " + (num \* num \* num));

}

}

public class Main {

public static void main(String[] args) {

Math\_Operations mathOperations = new Math\_Operations();

mathOperations.accept();

mathOperations.print\_Table();

}

}

Output:

Enter a number: 5

Table for 5:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

Square: 25

Cube: 125

***6. Write an interface Printable and implement it in below classes Employee(id,name,salary), Circle(radius), MyDate(dd,mm,yyyy) Your program should have Shape and rectangle classes which doesn't implements print. In main create objects for employee, circle and date and print the data using "print()"***

interface Printable {

void print();

}

class Employee implements Printable {

int id;

String name;

double salary;

Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

@Override

public void print() {

System.out.println("Employee ID: " + id);

System.out.println("Name: " + name);

System.out.println("Salary: " + salary);

}

}

class Circle implements Printable {

double radius;

Circle(double radius) {

this.radius = radius;

}

@Override

public void print() {

System.out.println("Circle Radius: " + radius);

System.out.println("area:"+(3.14\*radius\*radius));

}

}

class MyDate implements Printable {

int day;

int month;

int year;

MyDate(int day, int month, int year) {

this.day = day;

this.month = month;

this.year = year;

}

@Override

public void print() {

System.out.println("Date: " + day + "/" + month + "/" + year);

}

}

// Shape class (does not implement Printable)

class Shape {

// Shape-specific attributes and methods can be defined here

}

class Rectangle extends Shape {

double length;

double width;

Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

void display() {

System.out.println("Rectangle Length: " + length);

System.out.println("Rectangle Width: " + width);

}

}

public class Main {

public static void main(String[] args) {

Employee employee = new Employee(1, "Alice", 50000);

Circle circle = new Circle(7.5);

MyDate myDate = new MyDate(30, 7, 2024);

System.out.println("Employee Details:");

employee.print();

System.out.println();

System.out.println("Circle Details:");

circle.print();

System.out.println();

System.out.println("Date Details:");

myDate.print();

}

}

***Output:***

Employee Details:

Employee ID: 1

Name: Alice

Salary: 50000.0

Circle Details:

Circle Radius: 7.5

area:176.625

Date Details:

Date: 30/7/2024

***7. Write a class Shape(PI=3.142) (PI will be final double variable) declare double area() which will return zero method inherit class Circle(radius) from Shape implement area which will return area of a circle inherit class Rectangle(length, breadth) from Shape implement double area() which will return area of a Rectangle inherit class Cube(height) from Rectangle implement double area() which will return area of a Cube implement double volume() which will return volume of a Cube create objects of circle, rectangle and cube and test implemented methods***

import java.util.Scanner;

abstract class Shape {

final double PI = 3.142;

abstract double area();

}

class Circle extends Shape {

double radius;

Circle(double radius) {

this.radius = radius;

}

@Override

double area() {

return PI \* radius \* radius;

}

}

class Rectangle extends Shape {

double length;

double breadth;

Rectangle(double length, double breadth) {

this.length = length;

this.breadth = breadth;

}

@Override

double area() {

return length \* breadth;

}

}

class Cube extends Rectangle {

double height;

Cube(double length, double breadth, double height) {

super(length, breadth);

this.height = height;

}

@Override

double area() {

return 2 \* (length \* breadth + breadth \* height + height \* length);

}

double volume() {

return length \* breadth \* height;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter radius of the circle: ");

double radius = scanner.nextDouble();

Circle circle = new Circle(radius);

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

System.out.print("Enter length of the rectangle: ");

double length = scanner.nextDouble();

System.out.print("Enter breadth of the rectangle: ");

double breadth = scanner.nextDouble();

Rectangle rectangle = new Rectangle(length, breadth);

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

System.out.print("Enter length of the cube: ");

double cubeLength = scanner.nextDouble();

System.out.print("Enter breadth of the cube: ");

double cubeBreadth = scanner.nextDouble();

System.out.print("Enter height of the cube: ");

double height = scanner.nextDouble();

Cube cube = new Cube(cubeLength, cubeBreadth, height);

System.out.println("Surface Area of the Cube: " + cube.area());

System.out.println("Volume of the Cube: " + cube.volume());

}

}

***Output:***

Enter radius of the circle: 3

Area of the Circle: 28.278

Enter length of the rectangle: 4

Enter breadth of the rectangle: 5

Area of the Rectangle: 20.0

Enter length of the cube: 5

Enter breadth of the cube: 6

Enter height of the cube: 7

Surface Area of the Cube: 214.0

Volume of the Cube: 210.0

***6. write a java program of make abstract class ball method is getprice(), another class ironball,and another class cricket ball ,baouncable is interface in this bounce is method***

interface Bounceable {

void bounce();

}

abstract class Ball {

abstract double getPrice();

}

class IronBall extends Ball implements Bounceable {

private double price;

IronBall(double price) {

this.price = price;

}

@Override

double getPrice() {

return price;

}

@Override

public void bounce() {

System.out.println("IronBall bounces with a dull sound.");

}

}

class CricketBall extends Ball implements Bounceable {

private double price;

CricketBall(double price) {

this.price = price;

}

@Override

double getPrice() {

return price;

}

@Override

public void bounce() {

System.out.println("CricketBall bounces with a sharp sound.");

}

}

public class Main {

public static void main(String[] args) {

IronBall ironBall = new IronBall(15.99);

CricketBall cricketBall = new CricketBall(25.50);

System.out.println("IronBall Price: $" + ironBall.getPrice());

ironBall.bounce();

System.out.println("CricketBall Price: $" + cricketBall.getPrice());

cricketBall.bounce();

}

}

***Output:***

IronBall Price: $15.99

IronBall bounces with a dull sound.

CricketBall Price: $25.5

CricketBall bounces with a sharp sound.