**Assignment 6 - OOPS With JAVA**

**Name – Vaibhav Balasaheb Gargote**

**R\_no – 200940381125**

import java.util.\*;

import java.lang.Math;

//1. Create a class Student with 2 data members rollno and name. Create one method setData()

// that takes roll number and student name as parameter and stores them in data members

// rollno and name. Create one more method showData() to print the data member values.

// Create another class ( main class) StudentDemo that creates Student class object and

// calls setData() and showData() methods.

class Student

{

int rollno;

String name;

void setData(int rollno, String name)

{

this.rollno=rollno;

this.name=name;

}

void showData()

{

System.out.println("ID: "+rollno);

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

}

}

class Assignment6

{

public static void main(String [] args)

{

Student obj=new Student();

obj.setData(12345,"Sumit");

obj.showData();

}

}

//------------------------------------------------------------------------------------------------------

//2. Create a class Circle that has two data members, one to store the radius and

// another to store area and three methods first init() method to input radius

// from user, second calculateArea() method to calculate area of circle and third

// display() method to display values of radius and area. Create class CircleDemo

// (main class) that creates the Circle object and calls init(), calculateArea() and display() methods.

class Circle

{

int radius;

float area;

void init()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter Radius: ");

radius=sc.nextInt();

}

void calculateArea()

{

area=(float)(22/7.0f)\*(radius\*radius);

}

void display()

{

System.out.println("Area: "+area);

}

}

class Assignment6

{

public static void main(String [] args)

{

Circle obj=new Circle();

obj.init();

obj.calculateArea();

obj.display();

}

}

//------------------------------------------------------------------------------------------------------

//3. Create a class MathOperation that has four static methods. add() method that takes

// two integer numbers as parameter and returns the sum of the numbers. subtract() method

// that takes two integer numbers as parameter and returns the difference of the numbers.

// multiply() method that takes two integer numbers as parameter and returns the product.

// power() method that takes two integer numbers as parameter and returns the power of first

// number to second number. Create another class Demo (main class) that takes the two numbers

// from the user and calls all four methods of MathOperation class by providing entered numbers

// and prints the return values of every method.

class MathOperation

{

static int add(int a, int b)

{

return a+b;

}

static int subtract(int a, int b)

{

if(b>a)

return b-a;

else

return a-b;

}

static int multiply(int a, int b)

{

return a\*b;

}

static double power(int a, int b)

{

return Math.pow(a,b);

}

}

class Assignment6

{

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("a: ");

int a=sc.nextInt();

System.out.println("b: ");

int b=sc.nextInt();

System.out.println(MathOperation.add(a,b));

System.out.println(MathOperation.subtract(a,b));

System.out.println(MathOperation.multiply(a,b));

System.out.println(MathOperation.power(a,b));

}

}

//------------------------------------------------------------------------------------------------------

//4. Create a class MathOperation containing overloaded methods ‘multiply’ to calculate

// multiplication of following arguments.

// a. two integers

// b. three floats

// c. all elements of array

// d. one double and one integer

class MathOperation

{

int multiply(int a, int b)

{

return a\*b;

}

double multiply(float a, float b, float c)

{

return a\*b\*c;

}

double multiply(int arr[])

{

double pro=0;

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

{

pro=pro+arr[i];

}

return pro;

}

double multiply(double a, int b)

{

return a\*b;

}

}

class Assignment6

{

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

System.out.print("Int a: ");

int a=sc.nextInt();

System.out.print("Int b: ");

int b=sc.nextInt();

System.out.print("Float c: ");

float c=sc.nextFloat();

System.out.print("Float d: ");

float d=sc.nextFloat();

System.out.print("Float e: ");

float e=sc.nextFloat();

System.out.print("Enter Array Length: ");

int x=sc.nextInt();

int arr[]=new int[x];

for(int i=0;i<x;i++)

{

System.out.print("a[i+1]: ");

arr[i]=sc.nextInt();

}

System.out.print("Double f: ");

double f=sc.nextDouble();

System.out.print("Int g: ");

int g=sc.nextInt();

MathOperation obj=new MathOperation();

System.out.println("\nDouble a\*b: "+obj.multiply(a,b));

System.out.println("Double c\*d: "+obj.multiply(c,d,e));

System.out.println("Double e\*f: "+obj.multiply(arr));

System.out.println("Double g\*h: "+obj.multiply(a,b));

}

}

//------------------------------------------------------------------------------------------------------

//5. Create a class Employee with three data members (empNo, salary and totalSalary) and following features:

// a. Only parameterized constructor. [Do not overload the constructor]

// b. totalSalary always represents salary total of all the employees created.

// c. empNo should be auto incremented.

// d. display total employees and totalSalary using a method.

// Create another class EmployeeDemo (main class) that creates some Employee objects and

// calls Employee method to display no. of employees and total of their salaries.

class Employee

{

static int empNo=0;

int salery;

static int totalSalery=0;

Employee(int sal)

{

salery=sal;

empNo++;

totalSalery=totalSalery+salery;

}

static void display()

{

System.out.println("\nTotal Employees: "+empNo+"\nTotal Salery of all Employees: "+totalSalery);

}

}

class Assignment6

{

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter total Entries: ");

int n=sc.nextInt();

for(int i=0;i<n;i++)

{

System.out.print("Enter Salery of Employee "+(i+1)+": ");

int s=sc.nextInt();

Employee emp=new Employee(s);

}

Employee.display();

}

}

//------------------------------------------------------------------------------------------------------

//6. Create class Product with three data members (pid, price, quantity) and parameterized constructor that takes values for all three data members.

// Create a main method in different class (say ProductDemo) and perform following task:

// a. Accept information for five Product objects from user and store objects in an array

// b. Find pid of product with highest price.

// c. Create a static method (with array of product’s object as argument) in Product class to calculate and return total amount spent on all products.

// (amount spent on single product = price of product \* quantity of product )

class Product{

int pid;

double price;

int quantity;

public Product(int pid, double price, int quantity) {

this.pid = pid;

this.price = price;

this.quantity = quantity;

}

static void productWithHighestPrice(Product[] p) {

double max=p[0].price;

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

if(p[i].price>max)max=p[i].price;

}

System.out.println();

System.out.println("Maximum price is := "+max);

System.out.println();

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

if(p[i].price==max) {

System.out.println("The Pid of Highest price is := "+p[i].pid);

}

}

}

static double[] moneySpentOn(Product[] arr) {

double[] sum =new double[arr.length];

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

sum[i]=(arr[i].price \* arr[i].quantity);

}

return sum;

}

}

public class ProductOfStore {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

Product[] arr=new Product[5];

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

System.out.print("Enter Product Id := ");

int pid=sc.nextInt();

System.out.print("Enter the Price := ");

double price=sc.nextDouble();

System.out.print("Enter the Quantities :=");

int quantity=sc.nextInt();

arr[i]=new Product(pid,price,quantity);

}

Product.productWithHighestPrice(arr);

double[] spent=Product.moneySpentOn(arr);

System.out.println();

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

System.out.println(arr[i]+" spent on each item := "+spent[i]);

}

}

}

//------------------------------------------------------------------------------------------------------

//7. Create a class OneBHK with instance variables roomArea, hallArea and price. Then create default constructor that initializes

// instance variables with some values and a parameterized constructor that takes values for all instance variables and stores them in instance variables.

// Now create a method named show() to print OneBHK’s instance variable values.

// Create another class TwoBHK which has (inherits) all the properties and behaviors of OneBHK and a new instance variable room2Area.

// Then create default constructor to initialize all 4 instance variables and a parameterized constructor to take the values for initialization of all

// instance variables. Override show() method to print all data member information.

// Write main method in another class (Say Demo) and store three TwoBHK flat’s information and print information using show method. Also print total amount of all flats.

class OneBHK{

int roomArea;

int hallArea;

int price;

public OneBHK() {

this.roomArea=10;

this.hallArea=10;

this.price=1000;

}

public OneBHK(int roomArea, int hallArea, int price) {

this.roomArea = roomArea;

this.hallArea = hallArea;

this.price = price;

}

void show() {

System.out.println("Room Area := "+this.roomArea+" Hall Area := "+this.hallArea+" Price := "+this.price);

}

}

class TwoBHK extends OneBHK{

int room2Area;

public TwoBHK(){

super();

this.room2Area=10;

}

public TwoBHK(int roomArea, int room2Area,int hallArea,int price) {

super(roomArea,hallArea,price);

this.room2Area = room2Area;

}

void show() {

System.out.println("Room 1 Area := "+this.roomArea+" Room 2 Area := "+this.room2Area+" Hall Area := "+this.hallArea+" Price := "+this.price);

}

static void totalAmountOfFlates(TwoBHK ob1,TwoBHK ob2,TwoBHK ob3) {

int total=ob1.price+ob2.price+ob3.price ;

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

}

}

public class ConstructionPropertiesPVTLTD {

public static void main(String[] args) {

// TODO Auto-generated method stub

TwoBHK ob1=new TwoBHK(100,100,100,1000);

TwoBHK ob2=new TwoBHK(200,200,200,2000);

TwoBHK ob3=new TwoBHK(300,300,300,3000);

ob1.show();

ob2.show();

ob3.show();

TwoBHK.totalAmountOfFlates(ob1, ob2, ob3);

}

}

//------------------------------------------------------------------------------------------------------

//8. Create three classes

// Faculty with two data members facultyId and salary and two methods, one intput() method for accepting facultyId as input and another printSalary() to print salary.

// FullTimeFaculty that inherits class Faculty with two data members’ basicSalary and allowance. Override input() method in this class that calls super class inut() method and accepts basicSalary and allowance as input. Salary should not be accepted as input but should be calculated using formula (basicSalary + allowance)

// PartTimeFaculty that inherits class Faculty with two data members’ workingHours, ratePerHour. Override input() method in this class that calls super class inut() method and accepts workingHours and ratePerHour as input. Salary should not be accepted as input but should be calculated using formula ( workingHour \* ratePerHour )

class Faculty{

int facultyId;

double salary;

void input(int facultyId) {

this.facultyId=facultyId;

}

void printSalary() {

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

}

}

class FullTimeFaculty extends Faculty{

int basicSalary;

int allowance;

void input(int facultyId) {

this.facultyId=facultyId;

}

void input(int facultyId,int basicSalary,int allowance)

{

super.input(facultyId);

this.basicSalary=basicSalary;

this.allowance=allowance;

super.salary=basicSalary + allowance;

}

}

class PartTimeFaculty extends Faculty{

int workingHours;

int ratePerHour;

void input(int facultyId) {

this.facultyId=facultyId;

}

void input(int id,int workingHours,int ratePerHour)

{

super.input(facultyId);

this.workingHours=workingHours;

this.ratePerHour=ratePerHour;

super.salary=this.workingHours \* this.ratePerHour;

}

}

public class CollegeFacultyData {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("For Full Time Faculty");

System.out.print("Enter Faculty id := ");

int facultyId=sc.nextInt();

System.out.print("Enter Faculty basic Salary := ");

int basicSalary=sc.nextInt();

System.out.print("Enter Faculty Allowance := ");

int allowance=sc.nextInt();

FullTimeFaculty ob=new FullTimeFaculty();

ob.input(facultyId, basicSalary, allowance);

System.out.println("For Part Time Faculty");

System.out.print("Enter Faculty id := ");

int facultyId1=sc.nextInt();

System.out.print("Enter Faculty Working Hours := ");

int workingHours=sc.nextInt();

System.out.print("Enter Faculty ratePerHour := ");

int ratePerHour=sc.nextInt();

PartTimeFaculty ob2=new PartTimeFaculty();

ob2.input(facultyId1, workingHours, ratePerHour);

ob.printSalary();

ob2.printSalary();

}

}

//------------------------------------------------------------------------------------------------------

//9. Create an Abstract class Processor with int member variable data and method showData to display data value. Create abstract method process() to define processing of member data.

// Create a class Factorial using abstract class Processor to calculate and print factorial of a number by overriding the process method.

// b. Create a class Circle using abstract class Processor to calculate and print area of a circle by overriding the process method.

// Ask user to enter choice (factorial or circle area). Also ask data to work upon. Use Processor class reference to achieve this mechanism.

abstract class Processor{

double data;

void showData() {

System.out.println("Data := "+this.data);

}

abstract void process(int n);

}

class Factorial extends Processor{

void process(int n) {

int fact=1;

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

fact=fact\*i;

}

data=fact;

}

}

class Circle extends Processor{

void process(int n) {

final float PI=3.14f;

double area=PI\*n\*n;

data=area;

}

}

public class ComputerProcessor {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

System.out.println("1.Factroial of number \n 2.Area of Circle ");

int choice=sc.nextInt();

switch(choice) {

case 1:Factorial ob=new Factorial();

System.out.println("Enter the number whose Factorial is to be find := ");

int n=sc.nextInt();

ob.process(n);

ob.showData();

break;

case 2: Circle ob1=new Circle();

System.out.println("Enter the Circle whose Area is to be find := ");

int r=sc.nextInt();

ob1.process(r);

ob1.showData();

break;

default:System.out.println("Wrong Choice");

}

}

}