

Programming Lab

[Home](#)[About Us](#)[Contact Us](#)[Privacy Policy](#)

Java Arrays Inheritance and Interface

1. Write a program to sort the elements of one dimensional array. Read value of

array elements through command line argument.

```
class u2_p1
{
    public static void main(String args[])
    {
        int a[]= new int[args.length];
        for(int i=0;i<args.length;i++)
        {
            a[i]=Integer.parseInt(args[i]);
        }

        int temp;
        for(int i=0;i<a.length;i++)
        {
            for(int j=i+1;j<a.length;j++)
            {
                if(a[i]<a[j])
                {
                    temp=a[i];
                    a[i]=a[j];
                    a[j]=temp;
                }
            }
        }

        for(int i=0;i<a.length;i++)
        {
            System.out.println(a[i]);
        }
    }
}
```

Search This Blog

Blog Archive

▼ 2020 (3)

▼ December (3)

[Welcome to Programming Lab](#)

[Java Programming](#)

[Advanced C Programming](#)

About Me

Dr. Darshana Patel , Dr. Bhavik Pandya

[View my complete profile](#)

2. Write a program to create an array to store 5 integer values. Also initialize the array with 5 numbers and display the array Elements in reverse order.

```
import java.util.*;

class u2_p2

{

    public static void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        int a[]= new int[5];

        for(int i=0;i<5;i++)

        {

            System.out.println("Enter value of an array");

            a[i]=sc.nextInt();

        }

        for(int i=4;i>=0;i--)

        {

            System.out.println(a[i]);

        }

    }

}
```

3. Write a program to find sum of two matrices of 3 x3.

```
import java.util.*;

class u2_p3

{

    public static void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        int a[][]= new int[3][3];

        int b[][]= new int[3][3];

        int c[][]= new int[3][3];

        for(int i=0;i<3;i++)

        {

            for(int j=0;j<3;j++)

            {

                System.out.println("Enter value of an array");

                a[i][j]=sc.nextInt();

            }

        }

        for(int i=0;i<3;i++)

        {


```

```

        for(int j=0;j<3;j++)
        {
            System.out.println("Enter value of an array");

            b[i][j]=sc.nextInt();
        } }
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        c[i][j]=a[i][j]+b[i][j];

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

    }

    System.out.println("\n");
} } }

```

4. Write program to create an array of company name and another array of price

quoted by the company. Fetch the company name who has quoted the lowest amount.

```

import java.util.*;

class u2_p4
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

        int p[]= new int[5];
        String n[]= new String[5];

        for(int i=0;i<5;i++)
        {
            System.out.println("Enter name of company");

            n[i]=sc.next();

            System.out.println("Enter price quoted");

            p[i]=sc.nextInt();

        }

        int min=p[0];

        for(int i=0;i<5;i++)
        {
            if(min>p[i])
            {
                min=p[i];
            }
        }
    }
}

```

```

    } }

for(int i=0;i<5;i++)

{
    if(min==p[i])
    {
        System.out.print("The company quated min amount is "+n[i]);
        break;
    }
} } }

```

5. Write an interface called numbers, with a method in Process(int x, int y). Write a class called Sum, in which the method Process finds the sum of two numbers and returns an int value. Write another class called Average, in which the Process method finds the average of the two numbers and returns an int.

```

interface Numbers

{

abstract int process(int x, int y);

}

class sum implements Numbers

{

    public int process( int x, int y)

    {

        int ans=x+y;

        return(ans); } }

class avg implements Numbers

{

    public int process( int x, int y)

    {

        int ans=(x+y)/2;

        return(ans); } }

class infdemo

{

    public static void main(String args[])

    {

        sum s1= new sum();

        System.out.println("Sum is"+s1.process(5,5));

        avg a1= new avg();

        System.out.println("Avg is"+a1.process (6,6)); } }

```

6. Create a class called NumberData that accept any array of the five numbers.

Create a sub class called Numplay which provides methods for followings:

1. Display numbers entered.

2. Sum of the number.

3. Average of the numbers.

4. Maximum of the numbers.

5. Minimum of the numbers.

Create a class that provides menu for above methods. Give choice from the

command-line argument.

```
import java.util.*;

class Numberdata
{
    int a[]= new int[5];

    Scanner sc= new Scanner(System.in);

    void createarray()
    {
        for(int i=0;i<5;i++)
        {
            System.out.println("\n Enter element");

            a[i]=sc.nextInt();
        }
    }
}

class Numplay extends Numberdata
{
    void display()
    {
        createarray();

        for(int i=0;i<5;i++)
        {
            System.out.println("\n element"+a[i]);
        }
    }

    void sum()
    {
        int s=0;

        createarray();

        for(int i=0;i<5;i++)
```

```
{  
    s=s+a[i];  
  
}  
  
System.out.println("\n Sum is "+s);  
}  
  
void avg()  
{  
    int s=0;  
    createarray();  
    for(int i=0;i<5;i++)  
    {  
        s=s+a[i];  
    }  
    float a=(float) s/5;  
    System.out.println("\n Avg is "+a);  
}  
  
void max()  
{  
    int m=0;  
    createarray();  
    for(int i=0;i<5;i++)  
    {  
        if(m<a[i])  
            m=a[i];  
    }  
    System.out.println("\n Maximum is "+m);  
}  
  
void min()  
{  
    createarray();  
    int m=a[0];  
    for(int i=0;i<5;i++)  
    {  
        if(m>a[i])  
            m=a[i];  
    }  
}
```

```

    }

    System.out.println("\n Minimum is "+m);

}

}

class u2_p6
{
    public static void main(String args[])
    {
        Numplay n1= new Numplay();

        int ch= Integer.parseInt(args[0]);

        switch(ch)
        {
            case 1:
                n1.display();
                break;

            case 2:
                n1.sum();
                break;

            case 3:
                n1.avg();
                break;

            case 4:
                n1.max();
                break;

            case 5:
                n1.min();
                break;

            default:
                System.out.println("\n Wrong choice");

        }

    }

}

```

7. Declare an abstract class Vehicle with an abstract method named numWheels(). provide subclasses Car and Truck that each implements this method. Create instance of these subclasses and demonstrate the use of this method

```
abstract class vehicle
```

```
{
```

```
    abstract void numwheels();
```

```
}
```

```
class car extends vehicle
```

```
{
```

```
    void numwheels()
```

```
    {
```

```
        System.out.println("\n Car class with four wheels");
```

```
    }
```

```
}
```

```
class truck extends vehicle
```

```
{
```

```
    void numwheels()
```

```
    {
```

```
        System.out.println("\n Truck class with six wheels");
```

```
    }
```

```
}
```

```
class u2_p7
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        car c1= new car();
```

```
        c1.numwheels();
```

```
        truck t1= new truck();
```

```
        t1.numwheels();
```

```
    }
```

```
}
```

8. Write an interface called Exam with a method Pass(int mark) that returns a Boolean. Write another interface called Classify with a method Division(int average) which returns a string. Write a class called Result which implements both Exam and Classify. The pass method should return true if the marks is greater than or equal to 35 else false. The division method must return “First” when the parameter average is 60 or more, “second” when average is 50 or more but below 60, “no division” when average is less than 50.

```
interface exam
```

```
{
```



```

boolean pass(int marks);
}

interface classify
{
    String division(int avg);
}

```

class result implements exam, classify

```

{
    public boolean pass(int marks)
    {
        if(marks>35)
            return(true);
        else
            return(false);
    }

    public String division(int avg)
    {
        if(avg>=60)
            return("first");
        else if(avg<60 && avg>=50)
            return("second");
        else
            return("no division");
    }
}

```

```

class u2_p8
{
    public static void main(String args[])
    {
        result r1= new result();
        System.out.println("\n Your result is  " +r1.pass(50));
        System.out.println("\n Your division is  " +r1.division(50));
    }
}

```

9. Create class calculation with an abstract method area(). Create Rectangle and Triangle subclasses of calculation and find area of rectangle and triangle.

```

abstract class calculation
{
    int l,b;
}

```

```
abstract void area();  
}
```

```
class rect extends calculation
```

```
{  
    rect(int p, int q)  
    {  
        l=p;  
        b=q;  
    }  
    void area()  
    {  
        System.out.println("\n Area of  rectangle is" +(l*b));  
    }  
}
```

```
class tri extends calculation
```

```
{  
    tri(int p, int q)  
    {  
        l=p;  
        b=q;  
    }  
    void area()  
    {  
        System.out.println("\n Area of  triangle is" +((l*b)/2));  
    }  
  
}
```

```
class u2_p9
```

```
{  
    public static void main(String args[])  
    {  
        rect r1= new rect(5,6);  
        r1.area();  
        tri t1= new tri(6,6);  
        t1.area();  
    }  
  
}
```

10. The abstract Vegetable class has four subclasses named cabbage, carrot and potato. Write an application that demonstrates how to establish this class hierarchy. Declare one instance variable of type string that indicates the color of a vegetable. Create and display instances of these object. Override the toString() method of object to return a string with the name of the vegetable and its color.

```
abstract class vegetable
```

```
{ String s;
```

```
abstract String toString();
```

```
}
```

```
class cabbage extends vegetable
```

```
{
```

```
    cabbage(String s1)
```

```
    {
```

```
        s=s1;
```

```
    }
```

```
    String toString()
```

```
    {
```

```
        return("Cabbage color is "+s);
```

```
    }
```

```
}
```

```
class carrot extends vegetable
```

```
{
```

```
    carrot(String s1)
```

```
    {
```

```
        s=s1;
```

```
    }
```

```
    String toString()
```

```
    {
```

```
        return("Carrot color is "+s);
```

```
    }
```

```
}
```

```
class potato extends vegetable
```

```
{
```

```
    potato(String s1)
```

```
    {
```

```
        s=s1;
```

```
    }

    String toString()

    {

        return("Potato color is "+s);

    }

}

class u2_p10

{

    public static void main(String args[])

    {

        cabbage o1= new cabbage("Green");

        System.out.println(o1.toString());

        carrot o2= new carrot("Red");

        System.out.println(o2.toString());

        potato o3= new potato("Yellowish");

        System.out.println(o3.toString());

    }

}
```

[Home](#)

Subscribe to: [Posts \(Atom\)](#)