Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

AIM: PROGRAM TO FIND THE PRODUCT HAVING THE LOWEST PRICE.

SOURCE CODE

import java.io.\*;

import java.lang.\*;

public class Product

{

int pcode,price;

String pname;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

public void getdata()

{

try

{

System.out.println("Enter the product code, price and name of the product: ");

pcode=Integer.parseInt(br.readLine());

price=Integer.parseInt(br.readLine());

pname=br.readLine();

}

catch(IOException e)

{

System.out.println(e);

}

}

void show()

{

System.out.println("Product code: "+pcode);

System.out.println("Price: "+price);

System.out.println("Name of the product: "+pname);

}

static void compare(Product p1,Product p2,Product p3)

{

System.out.println("The product with the lowest price is:");

if(p1.price<p2.price && p1.price<p3.price)

{

System.out.println("Product code: "+p1.pcode);

System.out.println("Price: "+p1.price);

System.out.println("Name of the product: "+p1.pname);

}

else if(p2.price<p1.price && p2.price<p3.price)

{

System.out.println("Product code: "+p2.pcode);

System.out.println("Price: "+p2.price);

System.out.println("Name of the product: "+p2.pname);

}

else

{

System.out.println("Product code: "+p3.pcode);

System.out.println("Price: "+p3.price);

System.out.println("Name of the product: "+p3.pname);

}

}

public static void main(String[]args)throws IOException

{

Product p1,p2,p3;

p1=new Product();

p2=new Product();

p3=new Product();

p1.getdata();

p2.getdata();

p3.getdata();

p1.show();

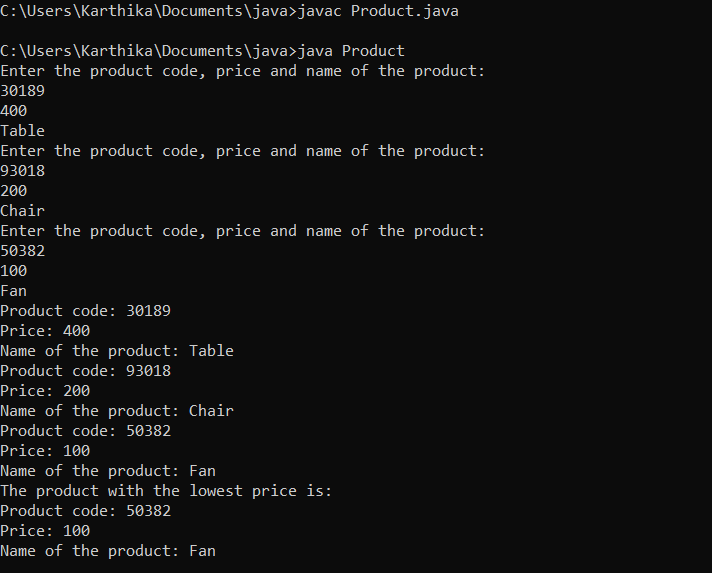
p2.show();

p3.show();

compare(p1,p2,p3);

}

}



2. Read 2 matrices from the console and perform matrix addition.

AIM: Program to read 2 matrices from the console and perform matrix addition.

SOURCE CODE

import java.io.\*;

public class Matrix

{

public static void main(String[] args)

{

int i,j;

int a[][]=new int[2][2];

int b[][]=new int[2][2];

int c[][]=new int[2][2];

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

try

{System.out.println("Enter the elements of the first matrix: ");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

a[i][j]=Integer.parseInt(br.readLine());

}

System.out.println();

}

System.out.println("Enter the elements of the second matrix: ");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

b[i][j]=Integer.parseInt(br.readLine());

}

System.out.println();

}

}

catch(IOException e)

{

System.out.println(e);

}

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

c[i][j]=a[i][j]+b[i][j];

}

}

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

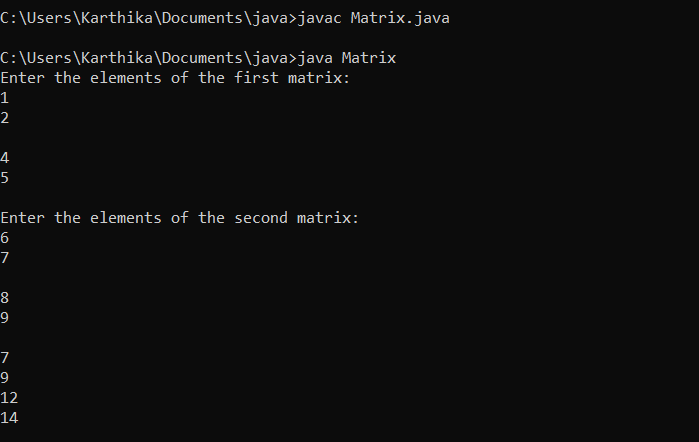
{

System.out.println(c[i][j]);

}

}

}}



3. Add complex numbers

Aim: Program to add complex numbers.

CODE:

import java.io.\*;

public class complex

{

int real;

int img;

BufferedReader br= new BufferedReader( new InputStreamReader(System.in));

public void getdata()

{

try

{

System.out.println("Enter the real part and imaginary part: ");

real=Integer.parseInt(br.readLine());

img=Integer.parseInt(br.readLine());

}

catch(IOException e)

{

System.out.println(e);

}

}

public void show()

{

System.out.println("complex no:"+ real +" +i"+ img);

}

public static void main(String[] args)

{

complex c1,c2;

c1 = new complex();

c2 = new complex();

c1.getdata();

c2.getdata();

c1.show();

c2.show();

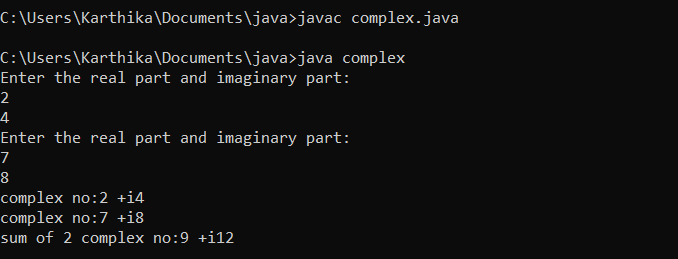
int x= c1.real + c2.real;

int y = c1.img + c2.img;

System.out.println("sum of 2 complex no:"+ x +" +i"+ y);

}

}



4. Read a matrix from the console and check whether it is symmetric or not

AIM : Program to check whether a matrix is symmetric or not.

SOURCE CODE

import java.io.\*;

public class matrix2

{

public static void main(String[] args)throws IOException

{

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

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

int r,c,i,j,flag=0;

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the number of rows and columns:");

r=Integer.parseInt(br.readLine());

c=Integer.parseInt(br.readLine());

System.out.println("Enter the elements of the matrix:");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

a[i][j]=Integer.parseInt(br.readLine());

}}

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

b[j][i]=a[i][j];

}

}

if(r==c)

{

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

if(b[i][j]!=a[i][j])

{

flag=1;

break;

}

}

}if(flag==1)

System.out.println("Is not symmetric.");

else

System.out.println("Is symmetric.");

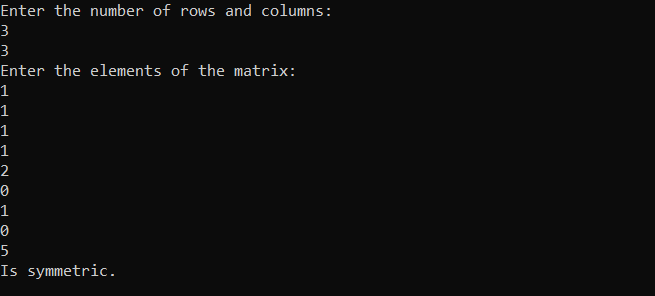
}

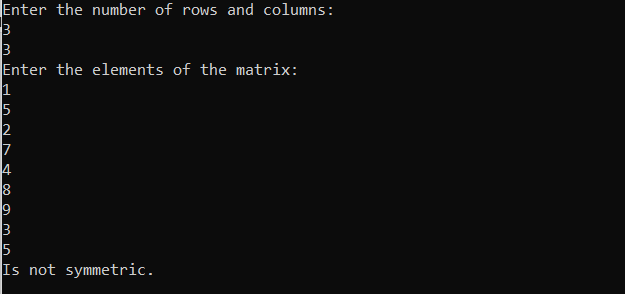
else

System.out.println("Since the given matrix is not a square, it isn't symmetric.");

}

}





5. Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

AIM: Program to create an object of CPU and print information of Processor and RAM.

import java.io.\*;

class CPU{

int price=50000;

class processor{

int noofcores=2;

String manuf="Intel";

}

static class RAM{

String mem="2GB";

String manuf="IBM";

}

public static void main(String []args) throws IOException

{

CPU c= new CPU();

CPU.processor p= c.new processor();

CPU.RAM r= new CPU.RAM();

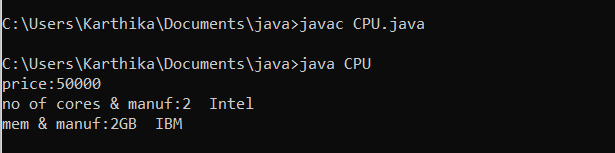
System.out.println("price:"+c.price);

System.out.println("no of cores & manuf:"+p.noofcores +" " +p.manuf);

System.out.println("mem & manuf:"+r.mem+" "+r.manuf);

}

}



6. Program to Sort strings

AIM: Program to Sort strings

/\*6. program to sort strings\*/

import java.io.\*;

public class arrstring

{

public static void main(String[] args) throws IOException

{

String[] arr= {"Java","CPP","Visual Basic", "Python"};

int size=arr.length;

for(int i=0;i<size-1;i++)

{

for(int j=i+1;j<size;j++)

{

if(arr[i].compareTo(arr[j])>0)

{

String temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

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

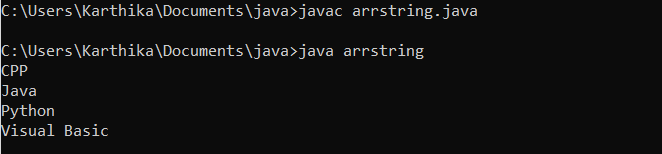
{

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

}

}

}



7. . Search an element in an array.

/\* search an element in an array\*/

import java.io.\*;

class array

{

public static void main(String []args)

{

int []arr= new int[5];

int item=0,flag=0,i=0;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

try

{

System.out.println("enter 5 numbers:");

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

{

arr[i]=Integer.parseInt(br.readLine());

}

System.out.println("enter the item to search:");

item=Integer.parseInt(br.readLine());

}

catch(IOException e)

{

System.out.println(e);

}

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

{

if(item==arr[i])

{

flag=1;

break;

}

}

if(flag==1)

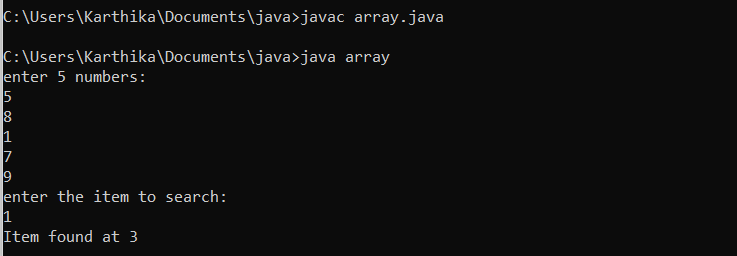
System.out.println("Item found at "+ (i+1));

else

System.out.println("Item not found ");

}

}



8. Perform string manipulations

import java.io.\*;

class strmanip

{

public static void main(String []args)

{

String s1,s2,s3;

int choice;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

try

{

System.out.println("enter 2 strings:");

s1=br.readLine();

s2=br.readLine();

do

{

System.out.println("enter ur(1-copy,2-append, 3-compare,4-reverse,5-exit) choice:");

choice=Integer.parseInt(br.readLine());

switch(choice)

{

case 1: s3=s1;

System.out.println("copied string:"+s3);

break;

case 2: s3=s1+ " "+s2;

System.out.println("appended strings:"+s3);

break;

case 3: int x=s1.compareTo(s2);

if(x==0)

System.out.println("strings are equal");

else if(x>0)

System.out.println("II string is larger");

else

System.out.println("I string is larger");

break;

case 4: s3=new StringBuffer(s1).reverse().toString();

System.out.println("reversed string:"+s3);

break;

}

}while(choice<5);

}

catch(IOException e)

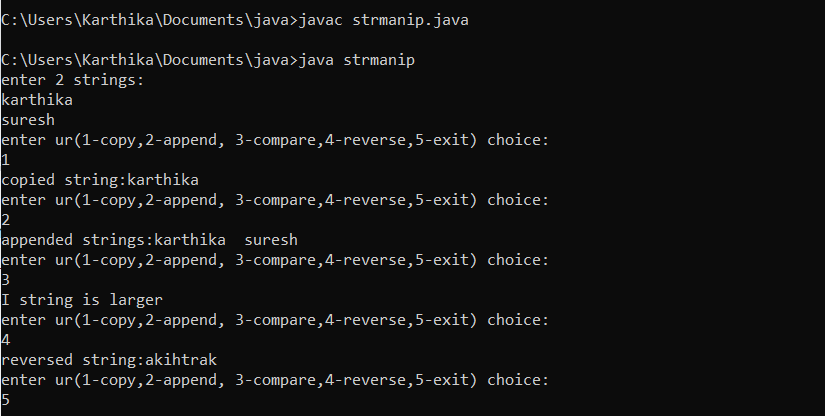
{

System.out.println(e);

}

}

}



9. Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

/\*9.array of objects\*/

import java.io.\*;

class emp

{

int empno,salary;

String ename;

void getdata(int empno, String ename, int salary)

{

this.empno=empno;

this.ename=ename;

this.salary=salary;

}

void display()

{

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

System.out.println("Emp no:"+ empno);

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

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

}

}

class Employee

{

public static void main(String[] args)

{

emp []e= new emp[5];

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int eno1=0,eno2=0,sal=0,i,n=0;

String name;

try

{

System.out.println("Enter no of employees:");

n=Integer.parseInt(br.readLine());

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

{

System.out.println("Enter the empno,ename,salary:");

eno1=Integer.parseInt(br.readLine());

name=br.readLine();

sal=Integer.parseInt(br.readLine());

e[i]=new emp();

e[i].getdata(eno1,name,sal);

}

System.out.println("Enter the emp no to search:");

eno2=Integer.parseInt(br.readLine());

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

{

if(e[i].empno==eno2)

{

e[i].display();

break;

}

}

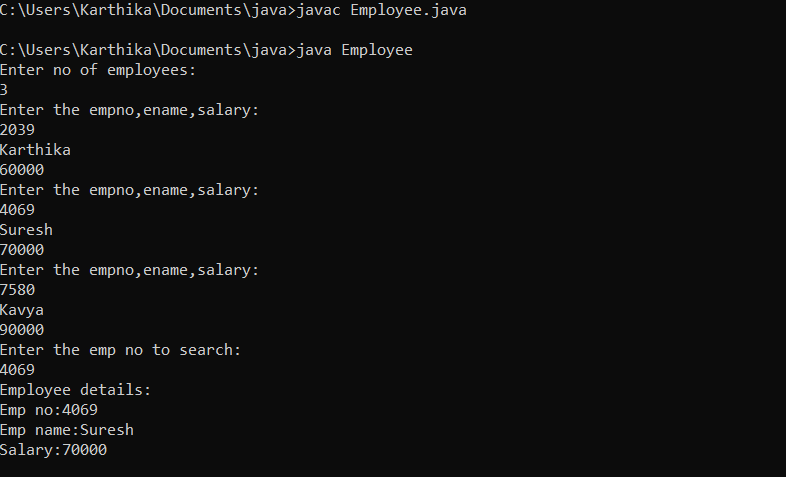
}

catch(IOException e1)

{System.out.println(e1);}

}

}



10. Area of different shapes using overloaded functions

/\*10. Area of figures using overloaded function\*/

import java.io.\*;

class over

{

void area(int l)

{

System.out.println("Area of square:"+l \* l);

}

void area(int l, int b)

{

System.out.println("Area of rectangle:"+l \* b);

}

void area(float l)

{

System.out.println("Area of circle:"+3.14 \* l \*l);

}

public static void main(String []args)

{

int l=0,b=0;

over o= new over();

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

try

{

System.out.println("Enter length & breadth:");

l= Integer.parseInt(br.readLine());

b= Integer.parseInt(br.readLine());

}

catch(IOException e)

{

System.out.println(e);

}

o.area(l);

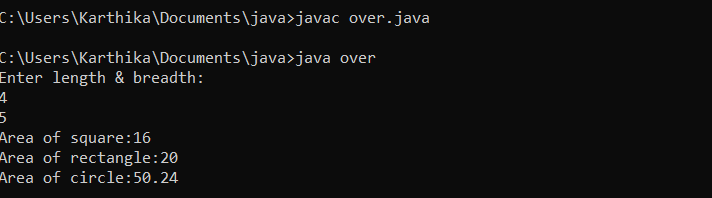
o.area(l,b);

float x= (float)l;

o.area(x);

}

}



11. Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

/\*11. single inheritance with constructor\*/

import java.io.\*;

class emp

{

int empid,salary;

String ename,address;

emp(int empid, String ename, int salary,String address)

{

this.empid=empid;

this.ename=ename;

this.salary=salary;

this.address=address;

}

}

class teacher extends emp

{

String dept,subj;

teacher(int empid, String ename, int salary,String address,String dept,String subj)

{

super(empid,ename,salary, address);

this.dept=dept;

this.subj=subj;

}

void display()

{

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

System.out.println("Emp no:"+ empid);

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

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

System.out.println("Address:"+ address);

System.out.println("Department:"+ dept);

System.out.println("Subject:"+ subj);

}

}

class inher

{

public static void main(String[] args)

{

teacher []t= new teacher[5];

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int eno1=0,sal=0,i,n=0;

String name,addr,dept,subj;

try

{

System.out.println("Enter no of employees:");

n=Integer.parseInt(br.readLine());

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

{

System.out.println("Enter the empno,ename,salary,address,dept,subj:");

eno1=Integer.parseInt(br.readLine());

name=br.readLine();

sal=Integer.parseInt(br.readLine());

addr=br.readLine();

dept=br.readLine();

subj=br.readLine();

t[i]=new teacher(eno1,name,sal,addr,dept,subj);

}

}

catch(IOException e1)

{System.out.println(e1);}

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

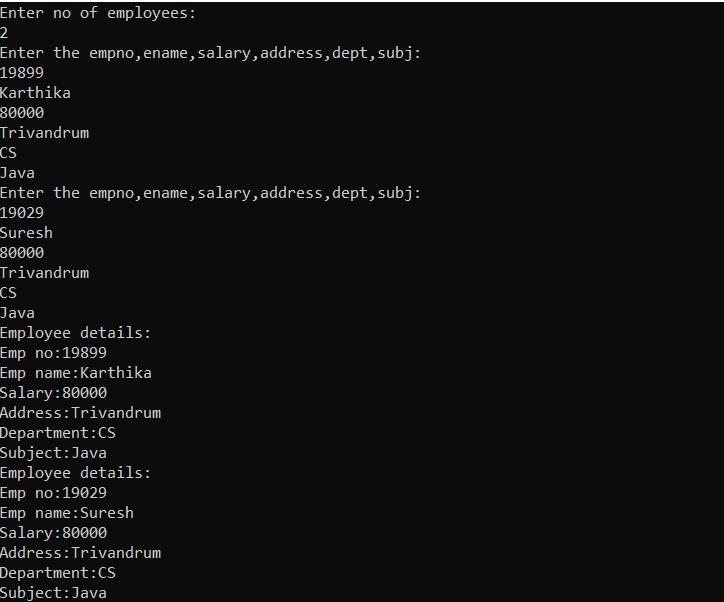
{

t[i].display();

}

}

}



12. Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

15. Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

import java.io.\*;

class students

{

int rollno;

String name;

float marks;

void getdata(int rollno,String name,float marks)

{

this.rollno=rollno;

this.name=name;

this.marks=marks;

}

}

interface sports

{

final int score=10;

void show();

}

class result extends students implements sports

{

public void show()

{

System.out.println("roll no:"+ rollno);

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

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

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

}

}

class inter1

{

public static void main(String []args)

{

int rollno=0;

String name="";

float marks=0;

result r = new result();

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

try

{

System.out.println("Enter rollno,name,marks:");

rollno=Integer.parseInt(br.readLine());

name=br.readLine();

marks=Float.parseFloat(br.readLine());

}

catch(IOException e)

{

System.out.println(e);

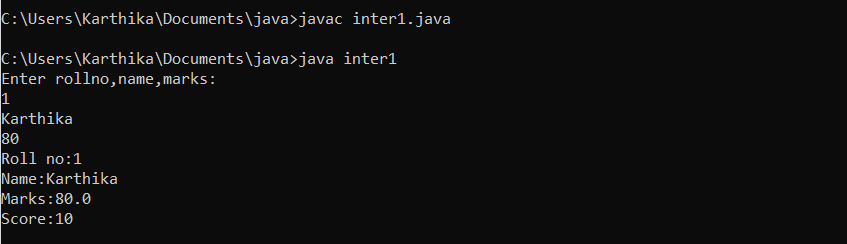
}

r.getdata(rollno,name,marks);

r.show();

}

}



16. Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

import java.io.\*;

interface areaperi

{

void area();

void perimeter();

}

class rectangle implements areaperi

{

int l=10,b=20;

public void area()

{

System.out.println("Area of rectangle:"+ (l\*b));

}

public void perimeter()

{

System.out.println("Perimeter of rectangle:"+(2\*(l+b)));

}

}

class circle implements areaperi

{

int r=10;

public void area()

{

System.out.println("Area of circle:"+ (3.14\*r\*r));

}

public void perimeter()

{

System.out.println("Perimeter of circle:"+ (2\*3.14\*r));

}

}

class inter2

{

public static void main(String[] args)

{

rectangle r = new rectangle();

circle c = new circle();

int ch=0;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

try

{

System.out.println("Enter 1 for circle, 2 for rectangle:");

ch=Integer.parseInt(br.readLine());

switch(ch)

{

case 1: c.area();

c.perimeter();

break;

case 2: r.area();

r.perimeter();

break;

}

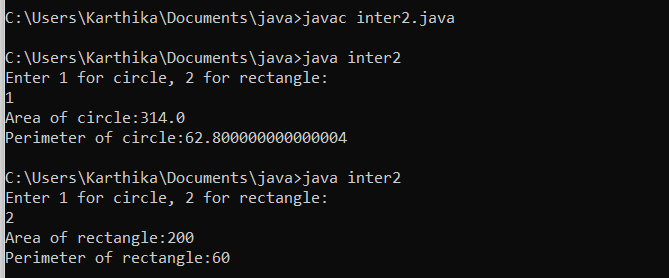
}

catch(IOException e1)

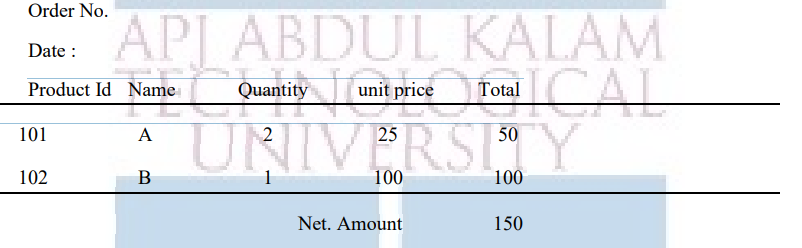
{System.out.println(e1);}

}

}



17. Prepare bill with the given format using calculate method from interface.



18. Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

package Graphics;

import java.io.\*;

public class rectangle1

{

int l,b;

BufferedReader br = new BufferedReader(new InputStreamReader (System.in));

public void getdata()

{

try

{

System.out.println("enter length & breadth:");

l=Integer.parseInt(br.readLine());

b=Integer.parseInt(br.readLine());

}

catch(IOException e)

{System.out.println(e);}

}

public void area()

{

System.out.println("Area of rectangle:"+(l\*b));

}

}

Find the average of N positive integers, raising a user defined exception for each negative input.

import java.io.\*;

class myexcep extends Exception

{

myexcep(String msg)

{

super(msg);

}

}

class positive

{

public static void main(String []args) throws IOException

{

int a[]= new int [5];

int l=0,i,s=0,av=0;

BufferedReader br = new BufferedReader(new InputStreamReader (System.in));

try

{

System.out.println("Enter 5 values:");

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

{

l=Integer.parseInt(br.readLine());

if(l<0)

throw new myexcep("Enter only positive value");

else

a[i]=l;

av=i+1;

}

}

catch(myexcep e)

{

System.out.println("Caught my exception");

System.out.println(e.getMessage());

}

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

{System.out.println("Array elements:"+a[j]);

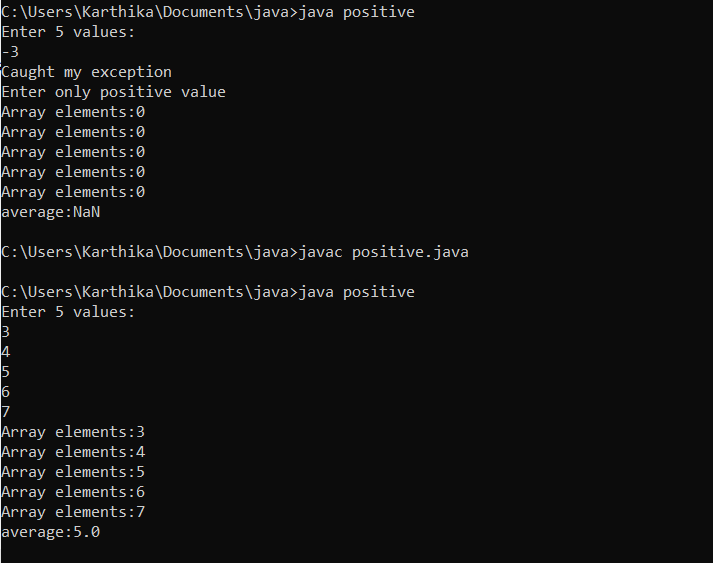
s+=a[j];

}

System.out.println("average:"+(float)((float)s/av));

}

}



Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

import java.io.\*;

class table extends Thread

{

public void run()

{

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

System.out.println("5 x "+i+" = "+(5\*i));

}

}

class prime extends Thread

{

public void run()

{

int flag=0;

for(int i=2;i<=10;i++)

{

flag=0;

for(int j=2;j<=i/2;j++)

{

if(i%j==0)

{

flag=1;

break;

}

}

if(flag==0)

System.out.println(i +" is prime");

}

}

}

class thread1

{

public static void main(String []args)

{

table t = new table();

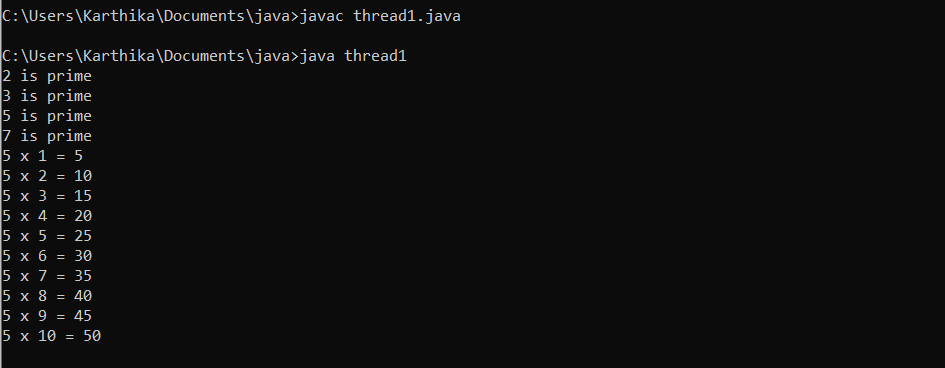
prime p = new prime();

t.start();

p.start();

}

}



Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

/\* runnable interface\*/

import java.io.\*;

class fibo implements Runnable

{

int a=0,b=1,c;

public void run()

{

System.out.println("Thread fibo:"+a);

System.out.println("Thread fibo:"+b);

for(int i=3;i<=10;i++)

{

c=a+b;

a=b;

b=c;

System.out.println("Thread fibo:"+c);

}

System.out.println("end of fibo");

}

}

class even implements Runnable

{

public void run()

{

for(int i=2;i<=10;i++)

System.out.println("Thread even:"+i);

System.out.println("end of even");

}

}

class thread2

{

public static void main(String []args)

{

fibo f= new fibo();

even e =new even();

Thread t1= new Thread(f);

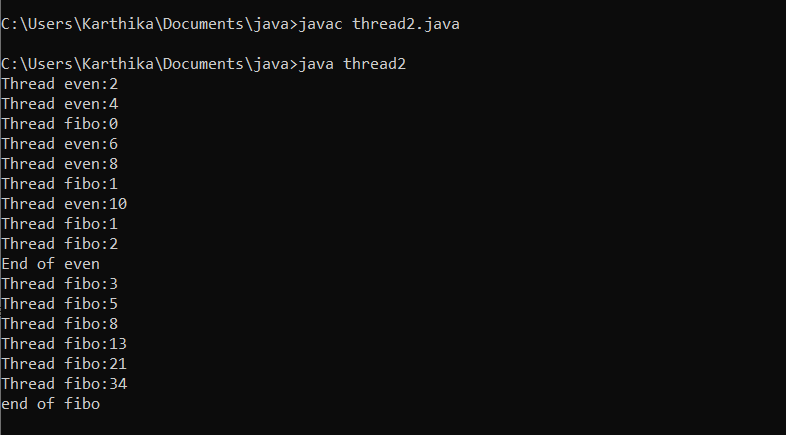
Thread t2= new Thread(e);

t1.start();

t2.start();

}

}



Producer/Consumer using ITC

import java.io.\*;

class Buffer

{

int a;

boolean produced=false;

public synchronized void produce(int x)

{

if(produced)

{

System.out.println("Producer is waiting");

try

{wait();}

catch(Exception e)

{System.out.println(e);}

}

a=x;

System.out.println("Product "+ a + " is produced");

produced=true;

notify();

}

public synchronized void consume()

{

if(!produced)

{

System.out.println("Consumer is waiting");

try

{wait();}

catch(Exception e)

{System.out.println(e);}

}

System.out.println("Product "+ a + " is consumed");

produced=false;

notify();

}

}

class producer extends Thread

{

Buffer b;

public producer(Buffer b)

{ this.b=b;}

public void run()

{

System.out.println("Producer start producing");

for(int i =1;i<=5;i++)

b.produce(i);

}

}

class consumer extends Thread

{

Buffer b;

public consumer(Buffer b)

{ this.b=b;}

public void run()

{

System.out.println("Consumer start consuming");

for(int i =1;i<=5;i++)

b.consume();

}

}

public class itc

{

public static void main(String []args)

{

Buffer b = new Buffer();

producer p = new producer(b);

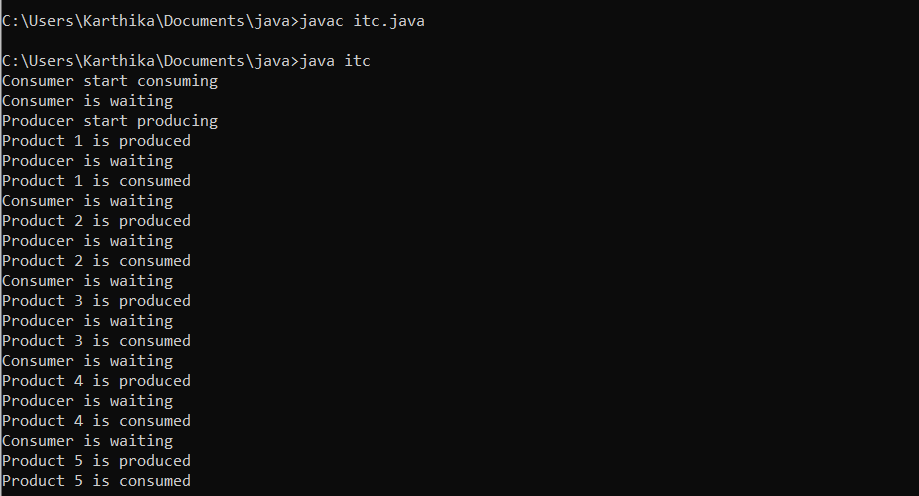
consumer c= new consumer(b);

p.start();

c.start();

}

}



Program to create a generic stack and do the Push and Pop operations.

import java.io.\*;

import java.util.\*;

class stack<T>

{

ArrayList<T> A;

int top = -1;

int size;

stack(int size)

{

this.size = size;

this.A = new ArrayList<T>(size);

}

void push(T X)

{

if (top + 1 == size)

{

System.out.println("Stack Overflow");

}

else

{

top = top + 1;

// Over-writing existing element

/\* if (A.size() > top)

A.set(top, X);

else

// Creating new element\*/

A.add(X);

}

}

T top()

{

if (top == -1)

{

System.out.println("Stack Underflow");

return null;

}

else

return A.get(top);

}

void pop()

{

if (top == -1)

System.out.println("Stack Underflow");

else

top--;

}

boolean empty() { return top == -1; }

public String toString()

{

String Ans = "";

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

Ans += String.valueOf(A.get(i)) + "->";

Ans += String.valueOf(A.get(top));

return Ans;

}

}

// Main Class

public class genstack

{

public static void main(String[] args)

{

stack<Integer> s1 = new stack<Integer>(3);

s1.push(10);

s1.push(20);

s1.push(30);

System.out.println("s1 after pushing 10, 20 and 30 :\n" + s1);

s1.pop();

System.out.println("s1 after pop :\n" + s1);

stack<String> s2 = new stack<String>(3);

s2.push("hello");

s2.push("world");

s2.push("java");

System.out.println("\ns2 after pushing 3 elements :\n" + s2);

stack<Float> s3 = new stack<Float>(2);

s3.push(100.0f);

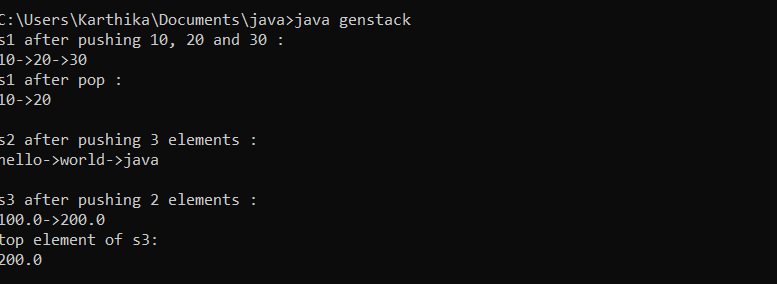
s3.push(200.0f);

System.out.println("\ns3 after pushing 2 elements :\n" + s3);

System.out.println("top element of s3:\n"+ s3.top());

}

}



Using generic method perform Bubble sort.

import java.util.Arrays;

public class BubSortGen<T extends Comparable<? super T>>

{

T[] array;

BubSortGen(T[] array)

{

this.array = array;

}

private T[] bubbleSort()

{

for(int i = array.length; i > 1; i--)

{

for(int j = 0; j < i - 1; j++)

{

if(array[j].compareTo(array[j+1]) > 0)

swapElements(j, array);

}

}

return array;

}

private void swapElements(int index, T[] arr)

{

T temp = arr[index];

arr[index] = arr[index+1];

arr[index+1] = temp;

}

public static void main(String[] args)

{

Integer[] intArr = {47, 62, 34, 7, 10, 2, 54};

BubSortGen<Integer> bsg1 = new BubSortGen<Integer>(intArr);

Integer[] sa1 = bsg1.bubbleSort();

System.out.println("Sorted Integer array- " + Arrays.toString(sa1));

String[] strArr = {"Earl", "Robert", "Asha", "Arthur"};

BubSortGen<String> bsg2 = new BubSortGen<String>(strArr);

String[] sa2 = bsg2.bubbleSort();

System.out.println("Sorted String array- " + Arrays.toString(sa2));

Float[] fArr = {100.0f, 90.0f,95.0f};

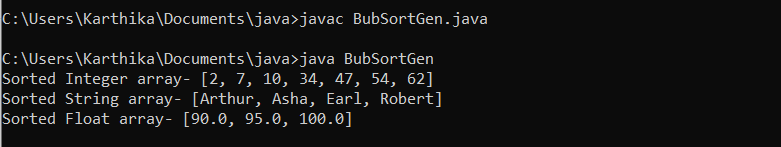
BubSortGen<Float> bsg3 = new BubSortGen<Float>(fArr);

Float[] sa3 = bsg3.bubbleSort();

System.out.println("Sorted Float array- " + Arrays.toString(sa3));

}

}



Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

import java.io.\*;

import java.util.\*;

class arrlist

{

public static void main(String[] args)

{

int n = 5;

ArrayList<Integer> arrli = new ArrayList<Integer>(n);

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

arrli.add(i);

System.out.println(arrli);

arrli.remove(3);

System.out.println(arrli);

arrli.add(2,33);

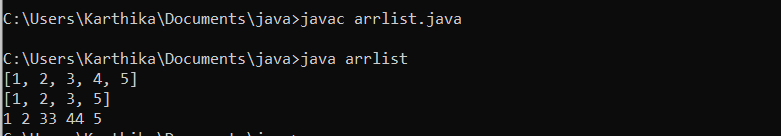
arrli.set(3,44);

for (int i = 0; i < arrli.size(); i++)

System.out.print(arrli.get(i) + " ");

}

}



Program to remove all the elements from a linked list

import java.io.\*;

import java.util.LinkedList;

public class LLDemo {

public static void main(String args[])

{

// Creating an empty LinkedList

LinkedList<String> list = new LinkedList<String>();

// Using add() method to add elements in the list

list.add("Geeks");

list.add("for");

list.add("Geeks");

list.add("10");

list.add("20");

// Displaying the List

System.out.println("Original LinkedList:" + list);

// Clearing the list

list.clear();

// Accessing the List after clearing it

System.out.println("List after clearing all elements: " + list);

// Adding elements after clearing the list

list.add("Geeks");

list.add("for");

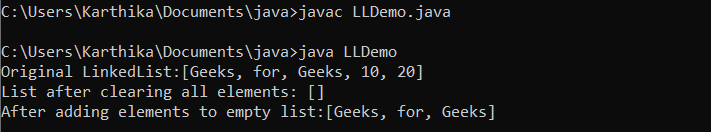
list.add("Geeks");

// Displaying the List

System.out.println("After adding elements to empty list:" + list);

}

}



Program to remove an object from the Stack when the position is passed as parameter

import java.io.\*;

import java.util.\*;

public class stack

{

public static void main(String []args)

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

Stack <String> s= new Stack<String>();

try

{

System.out.println("Enter 5 string values");

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

s.add(br.readLine());

System.out.println("Stack elements are:"+s);

int pos=0;

System.out.println("Enter the position of element to remove object:");

pos=Integer.parseInt(br.readLine());

String st= s.remove(pos);

System.out.println("Stack element removed:"+st);

System.out.println("Remaining Stack elements are:"+s);

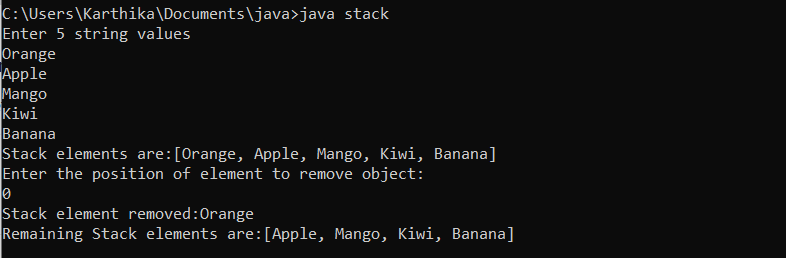
}

catch(IOException e)

{System.out.println(e);}

}

}



Program to demonstrate the creation of queue object using the PriorityQueue class

import java.util.\*;

class PQueue {

public static void main(String args[])

{

PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();

// Adding items to the pQueue using add()

pQueue.add(10);

pQueue.add(20);

pQueue.add(15);

pQueue.add(50);

System.out.println("Queue elements:"+pQueue);

System.out.println("First element:"+pQueue.peek());

System.out.println("Element removed:"+pQueue.poll());

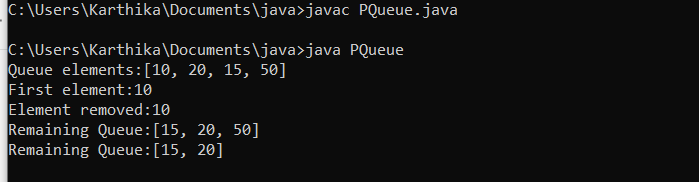
System.out.println("Remaining Queue:"+pQueue); //printing queue

pQueue.remove(50);

System.out.println("Remaining Queue:"+pQueue); //printing queue

}

}



Program to demonstrate the addition and deletion of elements in deque

import java.util.\*;

public class deque

{

public static void main(String[] args)

{

Deque<String> dq

= new LinkedList<String>();

// Add at the last

dq.add("Element 1 (Tail)");

// Add at the first

dq.addFirst("Element 2 (Head)");

// Add at the last

dq.addLast("Element 3 (Tail)");

// Add at the first

dq.push("Element 4 (Head)");

// Add at the last

dq.offer("Element 5 (Tail)");

// Add at the first

dq.offerFirst("Element 6 (Head)");

System.out.println(dq + "\n");

// We can remove the first element

// or the last element.

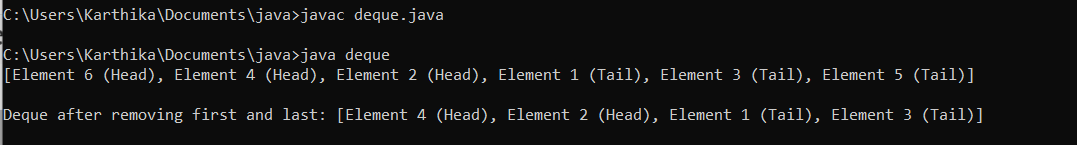
dq.removeFirst();

dq.removeLast();

System.out.println("Deque after removing " + "first and last: " + dq);

}

}



Program to demonstrate the creation of Set object using the LinkedHashset class

import java.util.LinkedHashSet;

public class LHSet

{

// Main Method

public static void main(String[] args)

{

LinkedHashSet<String> ls = new LinkedHashSet<String>();

// Adding element to LinkedHashSet

ls.add("A");

ls.add("B");

ls.add("C");

ls.add("D");

// This will not add new element as A already exists

ls.add("A");

ls.add("E");

System.out.println("Size of LinkedHashSet = " + ls.size());

System.out.println("Original LinkedHashSet:" + ls);

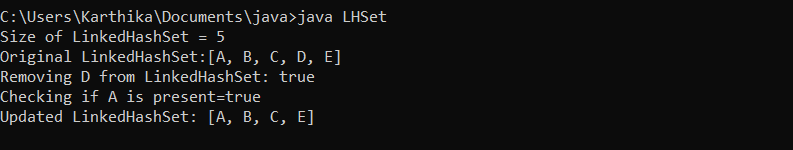
System.out.println("Removing D from LinkedHashSet: " +ls.remove("D"));

System.out.println("Checking if A is present=" + ls.contains("A"));

System.out.println("Updated LinkedHashSet: " + ls);

}

}



Write a Java program to compare two hash set

import java.util.\*;

public class hset

{

public static void main(String[] args)

{

// Create a empty hash set

HashSet<String> h\_set = new HashSet<String>();

// use add() method to add values in the hash set

h\_set.add("Red");

h\_set.add("Green");

h\_set.add("Black");

h\_set.add("White");

h\_set.add("Yellow");

for (String element : h\_set)

System.out.println("1st set:"+element);

HashSet<String> h\_set2 = new HashSet<String>();

h\_set2.add("Red");

h\_set2.add("Pink");

h\_set2.add("Black");

h\_set2.add("Orange");

h\_set2.add("Yellow");

for (String element : h\_set2)

System.out.println("2st set:"+element);

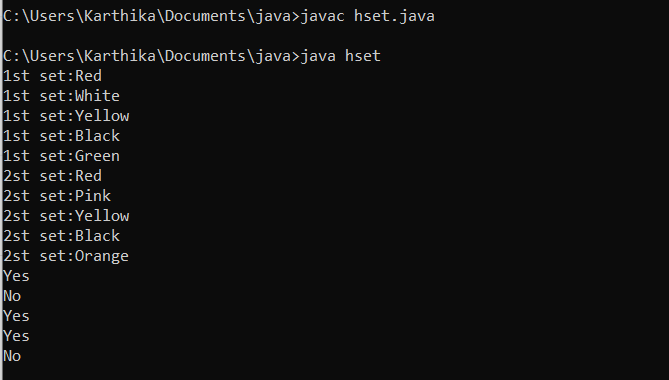
//comparison output in hash set

for (String element : h\_set)

System.out.println(h\_set2.contains(element) ? "Yes" : "No");

}

}



Program to demonstrate the working of Map interface by adding, changing and removing elements.

Program to Convert HashMap to TreeMap

import java.util.HashMap;

import java.util.TreeMap;

import java.util.Map;

public class hashtree

{

public static void main(String[] a)

{

Map<String, String> map = new HashMap<String, String>();

map.put("1", "A");

map.put("2", "B");

map.put("3", "C");

map.put("4", "D");

map.put("5", "E");

map.put("6", "F");

map.put("8", "H");

map.put("7", "G");

map.put("9", "I");

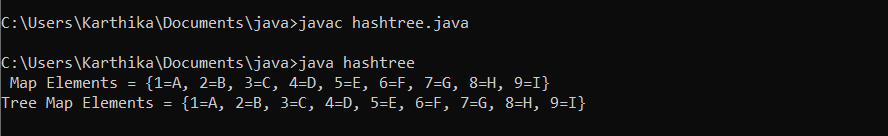
System.out.println(" Map Elements = "+map);

Map<String, String>sorted = new TreeMap<String, String>(map); //shows sorted form only

System.out.println("Tree Map Elements = "+sorted);

}

}



Program to draw Circle, Rectangle, Line in Applet.

CO.6

Program to list the sub directories and files in a given directory and also search for a file name.

AIM:

//writing and reading a file

import java.io.\*;

class file1

{

public static void main(String []a)

{

byte city[]={'D','E','L','H','I','\n','M','U','M','B','A','I','\n'};

FileOutputStream outfile=null; //writing to file

try

{

outfile= new FileOutputStream("city.txt");

outfile.write(city);

outfile.close();

}

catch(IOException e)

{

System.out.println(e);

}

FileInputStream infile=null; //reading from file

int b;

try

{

infile= new FileInputStream("city.txt");

while((b=infile.read()) != -1)

{

System.out.println((char)b);

}

infile.close();

}

catch(IOException e1)

{

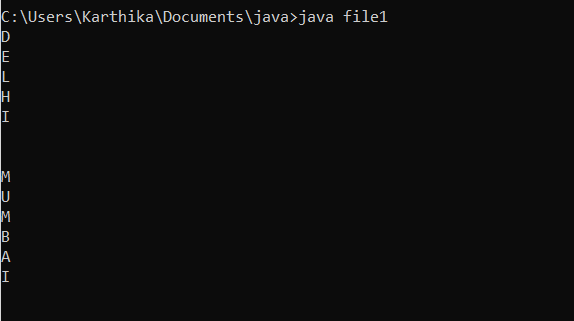
System.out.println(e1);

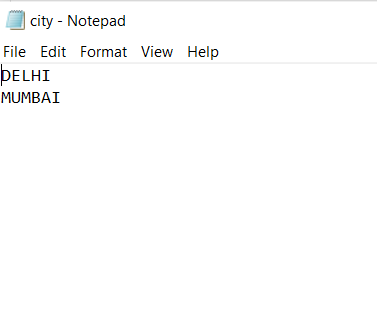
}

}

}

OUTPUT





Write a program to write to a file, then read from the file and display the contents on the console.

Write a program to copy one file to another.

//copying a file to another

import java.io.\*;

class file2

{

public static void main(String []a)

{

FileInputStream infile=null;

FileOutputStream outfile=null;

byte b;

try

{

infile = new FileInputStream("city.txt");

outfile= new FileOutputStream("citynew.txt");

do

{

b=(byte) infile.read();

outfile.write(b);

System.out.println((char)b);

}while(b != -1);

}

catch(IOException e)

{

System.out.println(e);

//System.exit(-1);

}

finally

{

try

{

outfile.close();

infile.close();

}

catch(IOException e1)

{

System.out.println(e1);

}

}

}

}

OUTPUT

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

/\* separating odd and even numbers\*/

import java.io.\*;

class file3

{

public static void main(String []a) throws IOException

{

int i;

File num= new File("integers.txt"); //creating main file

FileOutputStream fos=new FileOutputStream(num);

DataOutputStream dos= new DataOutputStream(fos); //for handling primitive data types

try

{

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

dos.writeInt(i);

}

catch(IOException e)

{System.out.println("from 1"+e);}

dos.close();

fos.close();

FileInputStream fis=new FileInputStream(num);

DataInputStream dis= new DataInputStream(fis);

File num1= new File("odd.txt"); //creating odd number file

FileOutputStream fos1=new FileOutputStream(num1);

DataOutputStream dos1= new DataOutputStream(fos1);

File num2= new File("even.txt"); //creating even number file

FileOutputStream fos2=new FileOutputStream(num2);

DataOutputStream dos2= new DataOutputStream(fos2);

try

{

System.out.println("file content:");

for(int j=1;j<=10;j++)

{

i=dis.readInt();

System.out.println("inside fn:"+i);

if(i%2==0)

dos2.writeInt(i);

else

dos1.writeInt(i);

}

}

catch(IOException e1)

{System.out.println("from 2"+e1);}

dos1.close();

fos1.close();

dos2.close();

fos2.close();

dis.close();

fis.close();

FileInputStream fis1=new FileInputStream(num1);

DataInputStream dis1= new DataInputStream(fis1);

System.out.println("\nOdd file: ");

try

{

for(int j=1;j<=5;j++)

{

i=dis1.readInt();

System.out.println(i + " ");

}

}

catch(IOException e2)

{System.out.println("from 3"+e2);}

fis1.close();

dis1.close();

FileInputStream fis3=new FileInputStream(num2);

DataInputStream dis3= new DataInputStream(fis3);

System.out.println("\nEven file: ");

try

{

for(int k=1;k<=5;k++)

{

i=dis3.readInt();

System.out.println(i + " ");

}

}

catch(IOException e2)

{System.out.println("from 4"+e2);}

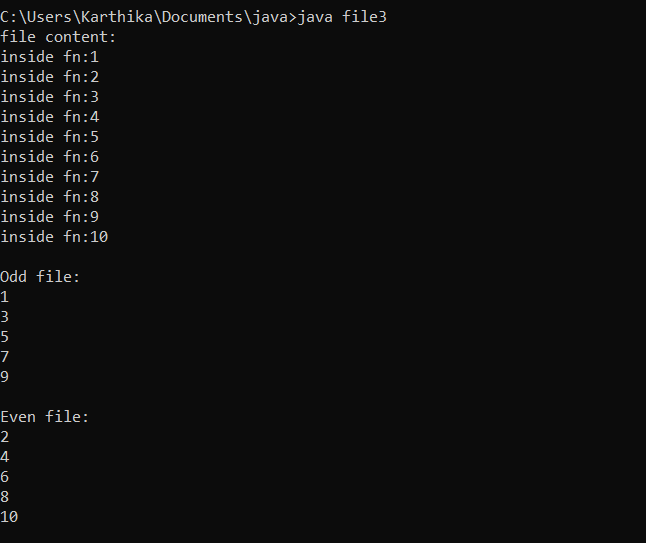
fis1.close();

dis1.close();

}

}

OUTPUT



Client server communication using Socket – TCP/IP

TCPCLIENT

import java.io.\*;

import java.net.\*;

class Tcpclient

{

public static void main(String []args) throws IOException

{

String s,ms;

BufferedReader infromuser = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Hello");

Socket clientsocket = new Socket("127.0.0.1",5000);

DataOutputStream outtoserver= new DataOutputStream(clientsocket.getOutputStream());

System.out.println("\n Enter a string:");

s=infromuser.readLine();

outtoserver.writeBytes(s+'\n');

BufferedReader infromserver = new BufferedReader(new InputStreamReader(clientsocket.getInputStream()));

ms=infromserver.readLine();

System.out.println("From server:"+ms);

clientsocket.close();

}

}

TCPSERVER

import java.io.\*;

import java.net.\*;

class Tcpserver

{

public static void main(String []args) throws Exception

{

String s1,ms1;

ServerSocket serversocket = new ServerSocket(5000);

while(true)

{

Socket clientsocket = serversocket.accept();

BufferedReader infromclient = new BufferedReader(new InputStreamReader(clientsocket.getInputStream()));

s1=infromclient.readLine();

ms1=s1.toUpperCase()+'\n';

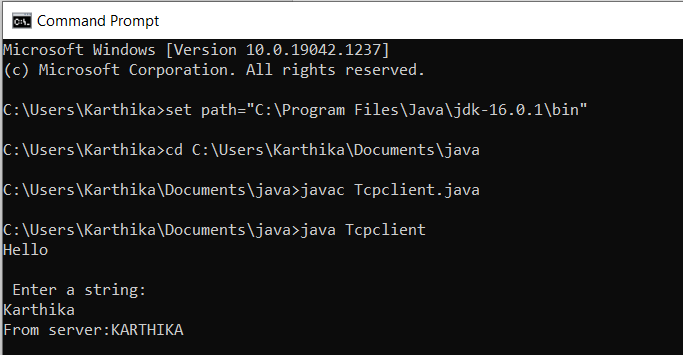
DataOutputStream outtoclient= new DataOutputStream(clientsocket.getOutputStream());

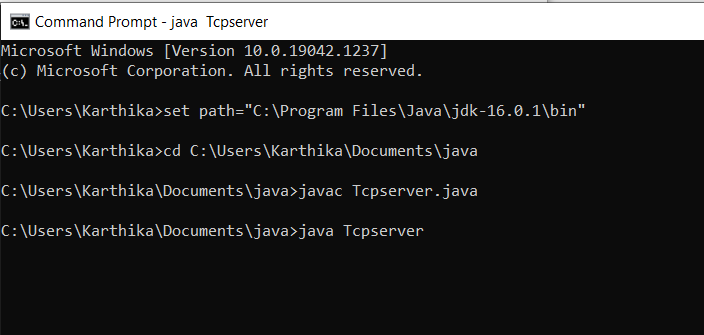
outtoclient.writeBytes(ms1);

}

}

}





Client Server communication using DatagramSocket – UDP

Udpclient

import java.io.\*;

import java.net.\*;

class udpclient

{

public static void main(String a[]) throws IOException

{

BufferedReader infromuser = new BufferedReader (new InputStreamReader(System.in));

DatagramSocket clientsocket = new DatagramSocket();

InetAddress ipaddress = InetAddress.getByName("127.0.0.1");

byte[] receivedata =new byte[1024];

byte[] senddata =new byte[1024];

System.out.println("\n Enter a string:");

String str= infromuser.readLine();

senddata= str.getBytes();

DatagramPacket sendpacket = new DatagramPacket (senddata,senddata.length,ipaddress,5000);

clientsocket.send(sendpacket);

DatagramPacket receivepacket = new DatagramPacket (receivedata,receivedata.length);

clientsocket.receive(receivepacket);

String modified = new String(receivepacket.getData());

System.out.println("From Server: "+modified);

clientsocket.close();

}

}

Udpserver

import java.io.\*;

import java.net.\*;

class udpserver

{

public static void main(String a[]) throws IOException

{

DatagramSocket serversocket=new DatagramSocket(5000);

byte[] receivedata =new byte[1024];

byte[] senddata =new byte[1024];

while(true)

{

DatagramPacket receivepacket=new DatagramPacket(receivedata,receivedata.length);

serversocket.receive(receivepacket);

String sentence=new String(receivepacket.getData());

InetAddress ipaddress =receivepacket.getAddress();

int port=receivepacket.getPort();

String capital=sentence.toUpperCase();

System.out.println(capital);

senddata=capital.getBytes();

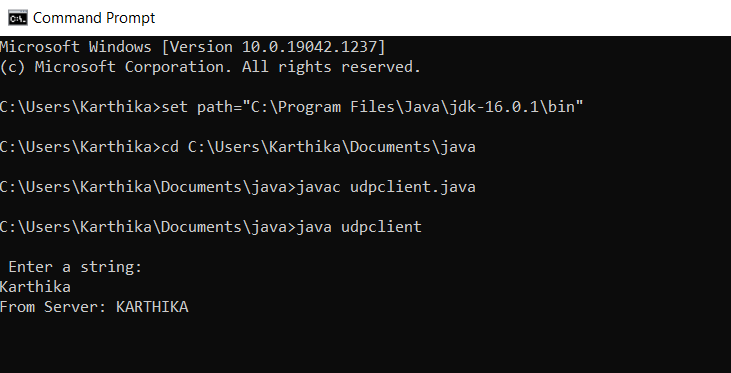
DatagramPacket sendpacket=new DatagramPacket(senddata,senddata.length,ipaddress,port);

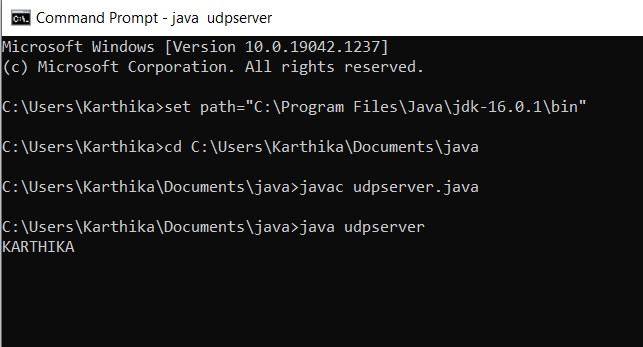
serversocket.send(sendpacket);

}

}

}





PACKAGE

package operations;

import java.util.Scanner;

interface calculation{

public void addition();

public void multiplication();

public void subtraction();

public void division();

}

public class arithop implements calculation {

double sum;

Scanner obj1=new Scanner(System.in);

public void addition() {

int l,b;

System.out.println("Enter first number");

l=obj1.nextInt();

System.out.println("Enter second number");

b=obj1.nextInt();

sum=l+b;

System.out.println("sum:"+sum+"\n");

}

public void multiplication() {

int h,b;

int mul;

System.out.println("Enter first number:");

b=obj1.nextInt();

System.out.println("Enter second number");

h=obj1.nextInt();

mul=h\*b;

System.out.println("result=:"+mul+"\n");

}

public void subtraction() {

float r,z,sub;

System.out.println("first number:");

r=obj1.nextInt();

System.out.println("second number:");

z=obj1.nextInt();

sub=r\*z;

System.out.println("Result:"+sub+"\n");

}

public void division() {

float r,z,div;

System.out.println("first number:");

r=obj1.nextInt();

System.out.println("second number:");

z=obj1.nextInt();

div=r/z;

System.out.println("Result:"+div+"\n");

}

}

TEST1

import java.io.\*;

import java.util.Scanner;

import operations.arithop;

public class Test1

{

public static void main(String[] args) {

Scanner obk=new Scanner(System.in);

arithop obj=new arithop();

int ch=0;

//l=obj1.nextInt();

while(ch<5) {

System.out.println("SELECT OPERATION:"+" 1.addition 2.multiplication3.subtraction 4.division 5.Exit");

ch=obk.nextInt();

switch(ch) {

case 1:

obj.addition();

break;

case 2:

obj.multiplication();

break;

case 3:

obj.subtraction();

break;

case 4:

obj.division();

break;

case 5: break;

default:

System.out.println("Invalid");

break;

}

}

}

}