ADVANCED JAVA.

NT35 (New Topics).

represents 1 istanstS of a program, interacts b/w ~~us~~ user & JVM, whenever we have to fetch some info^n we use this class's method.

Runtime Class and Process Class.

↳studied ~~in~~ some in Garbage collec^n → There we have studies 2 methods

- totalmemory()
- freememory()

Runtime Class → exec() method

↓ starts any program via its exe file.

Program (NotePad.java)

↳invoking notepad via JavaProgra or runtime class.

```
Class NodePad
{
psvm()
{
Runtime rt = Runtime.getRuntime();
                        ↳sg.
try
{
rt.exec("notepad.exe");
 rt.exec("c:\\ProgramFile - - - - \\windword.exe");
}
catch(Excep^n e)
{
 Sop(e);
}
 Sop("hello");
}
```

• with help of process class we can get Error O/P at runti

Program (MyEditar.java)

```
import ~~java.io~~ javax.swing.*;
          java.awt.*;
          java.awt.event.*;
          java.io.*;
          java.util.*;

class MyEditar implement~~ation~~ Actionlistener
{
    JFrame jf;
    JLabel jl;
    J TesdField jtf;
    J TesdArea jta, jtal;
    JButton jbcompile, jbrun;
    JScrollPane jsp, jsp1;
    Runtime r;
    String str="";
    String fname="";
    String result="";
    String result1="";

 MyEditor()
 {
  jf=new JFrame("My Editar");
  jf.setLayout(null);
  jl=new JLabel("Enter java class Name");
  jl=setBounds(20,20,130,25);
  ~~jf~~ jtf = new J TesdField();
~~jtf~~
  jtf.setBounds(180,20,230,25);
  jta=new JTextArea(50,50);
jta.addFocusListener(new MyFocusListener(this));
    jta1=new JTextArea(50,50);
    jta.setFont(new Font("varinda",Font.PLAIN,15));
 jta.setFont(new Font("varinda", Font.PLAIN,15));
   jsp= new JScrollPane(jta);
  jsp1 = new JScrollPane(jta1);
  jsp.setBounds(50,60,320,150);
  jsp1.setBounds(50,270,320,150);
  jf.add(jsp);
  jf.add(jsp1);
  jbcompile=new JButton("compile");
  jbrun =new JButton ("Run");
  ~~jbcom~~ jbcompile.setBounds(100,230,80,25);
      jbrun.setBounds(280,230,80,25);
    jf.add(jl);
    jf.add(jtf);
   r=Runtime.getRuntime();
   jf.add(jbcompile);
  jf.add(jbrun);
  jbcompile.addActionListener(this);
  jbrun.addActionListener(this);
jf.setDefaultCloseOperation(JFrame.EXIT_ON_
                                     CLOSE);
  jf.setSize(550,550);
  jf.setVisible(true);
}
```

```
public void actionPerformed(ActionEvent e)
{
   if(e.getSource()==jbcombile)
   {
      str="";

      if(!jtf.getText().equals(""))
      {
         try
         {
fname=jtf.getText().trim()+".java";
FileWriter fw=new FileWriter(fname);
String s1=jta.getText();
PrintWriter pw=new PrintWriter(fw);
  pw.println(s1);
  pw.flush();

    Process error=r.exec("c:\\Program-
       -Files\\Java\\jdk-10\\bin\\javac.exe
                 -d.   "+fname);

BufferedReader err=new BufferedReader
          (new InputStreamReader(error.
            getErrorStream()));


While(true)
{
String temp=err.readLine();
if(temp!=null)
{
        result += temp;
         result += "\n";
     }
   else break;
   }
  if (result.equals(""))
     {
     jta1.setText("compilation successful!"+fname);
       err.close();
     }
    else
   jta1.setText(result);
  }
 catch(Exception e1)
   {
   System.out.println(e1);
  }
 }
 else
  jta1.setText("Please Enter the jave program name");
  }
  else if (e.getSource()==jbrun)
   {
   int start=0;
   try
  {
 String fn=jtf.getText().trim();
  Process  p=r.exec("C:\\Program Files\\Java\\
                    jdk-9\\bin\\java. "+fn);
```

```
BufferedReader output = new BufferedReader
        (new InputStreamReader(p.getInputStream()));

BufferedReader error = new BufferedReader
     (new InputStreamReader(p.getErrorStream()));

  While(true)
  {
  String temp = output.readLine();
   if (temp!=null)
  {
  result+=temp;
  result += "\n";
  }
  else
  {
  break;
  }
  }
 While(true)
  {
 String temp = error.readline();
 if (temp!=null)
   {
  result1 += temp;
  result1 += "\n";
  }
  else
  {
      break;
    }
  }
   Output.close();
   error.close();

  jta1.setText(result+"\n"+result1);
  }

  catch(Exception e2)
   {
   Sop(e2);
   }
 }
}

psvm
{
new MyEditor();
 }
}

Class MyFocuslistener extends FocusAdapter
 {
  MyEditor e;
  MyFocuslistener(MyEditor e)
   {
   this.e=e;
   }
  public void focusGained (FocusEvent ffe)
{
String str=e.jtf.getText().trim();
e.jta.setText("public class"+str+"\n"~~+~~
          + "{"+"\n"
          +"public static void main(String... s)"+"\n"
         + "{"+"\n"
         +"          "          "+"\n"
         +"}      "+"\n"
         +"}      ");
      }
   }
```