1.**electricity bill**

//import java.util.\*;

public class Ebill

{

public static void main(string args[])

{

customerdata ob=new customerdata();

ob.getdata();

ob.calc();

ob.dispaly();

}

}

class Customerdata

{

scanner in=new scanner(system.in);

scanner ins=new scanner(system.in);

string cname,type;

int bn;

double current,previous,tbills,units;

void getdata()

{

system.out.print("\n\t Enter consumer number");

bn=in.nextlnt();

system.out.print("\n\t Enter type of connection(D for Domestic or C for Commercial)");

type=ins.nextLine();

system.out.print("\n\t Enter consumer name");

cname=ins.nextLine();

system.out.print("\n\t Enter previous month readings");

previous=in.nextDouble>>();

system.out.print("\n\t Enter current month reading");

current=in.nextDouble();

}

void calc()

{

units=current-previous;

if(type.equals("D"))

{

if(units<=100)

tbill=1\*units;

else if(units>100 && units<=200)

tbill=2.50\*units;

else if(units>200 && units<=500)

tbill=4\*units;

else

tbill=6\*units;

}

else

{

if(units<=100)

tbill=2\*units;

else if(units>100 && units<=200)

tbill=4.50\*units;

else if(units>200 && units<=5000

tbill=6\*units;

else

tbill=7\*units;

}

}

void display()

{

system.out.println("\n\t Consumer number="+bn);

system.out.println("\n\t Consumer name="+cname);

if(type.equals("D"))

system.out.println("\n\t type of connection=DOMESTIC");

else

system.out.println("\n\t type of connection=COMMERCIAL");

system.out.println("\n\t Current month readings="+current);

system.out.println("\n\t previous month readings="+previous);

system.out.println("\n\t Total units="+units);

system.out.println(\n\t Total bill=RS"+tbill);

}

}



2.converters

currency.java

package currencyconversion;

import java.util.\*;

public class currency

{

double inr,usd;

double euro,yen;

Scanner in=new Scanner(System.in);

public void dollartorupee()

{

System.out.println("Enter dollars to convert into Rupees:");

usd=in.nextInt();

inr=usd\*67;

System.out.println("Dollar ="+usd+"equal to INR="+inr);

}

public void rupeetodollar()

{

System.out.println("Enter Rupee to convert into Dollars:");

inr=in.nextInt();

usd=inr/67;

System.out.println("Rupee ="+inr+"equal to Dollars="+usd);

}

public void eurotorupee()

{

System.out.println("Enter euro to convert into Rupees:");

euro=in.nextInt();

inr=euro\*79.50;

System.out.println("Euro ="+euro +"equal to INR="+inr);

}

public void rupeetoeuro()

{

System.out.println("Enter Rupees to convert into Euro:");

inr=in.nextInt();

euro=(inr/79.50);

System.out.println("Rupee ="+inr +"equal to Euro="+euro);

}

public void yentorupee()

{

System.out.println("Enter yen to convert into Rupees:");

yen=in.nextInt();

inr=yen\*0.61;

System.out.println("YEN="+yen +"equal to INR="+inr);

}

public void rupeetoyen()

{

System.out.println("Enter Rupees to convert into Yen:");

inr=in.nextInt();

yen=(inr/0.61);

System.out.println("INR="+inr +"equal to YEN"+yen);

}

}

distance.java

package distanceconversion;

import java.util.\*;

public class distance

{

double km,m,miles;

Scanner sc = new Scanner(System.in);

public void kmtom()

{

System.out.print("Enter in km ");

km=sc.nextDouble();

m=(km\*1000);

System.out.println(km+"km" +"equal to"+m+"metres");

}

public void mtokm()

{

System.out.print("Enter in meter ");

m=sc.nextDouble();

km=(m/1000);

System.out.println(m+"m" +"equal to"+km+"kilometres");

}

public void milestokm()

{

System.out.print("Enter in miles");

miles=sc.nextDouble();

km=(miles\*1.60934);

System.out.println(miles+"miles" +"equal to"+km+"kilometres");

}

public void kmtomiles()

{

System.out.print("Enter in km");

km=sc.nextDouble();

miles=(km\*0.621371);

System.out.println(km+"km" +"equal to"+miles+"miles");

}

}

timer.java

package timeconversion;

import java.util.\*;

public class timer

{

int hours,seconds,minutes;

int input;

Scanner sc = new Scanner(System.in);

public void secondstohours()

{

System.out.print("Enter the number of seconds: ");

input = sc.nextInt();

hours = input / 3600;

minutes = (input % 3600) / 60;

seconds = (input % 3600) % 60;

System.out.println("Hours: " + hours);

System.out.println("Minutes: " + minutes);

System.out.println("Seconds: " + seconds);

}

public void minutestohours()

{

System.out.print("Enter the number of minutes: ");

minutes=sc.nextInt();

hours=minutes/60;

minutes=minutes%60;

System.out.println("Hours: " + hours);

System.out.println("Minutes: " + minutes);

}

public void hourstominutes()

{

System.out.println("enter the no of hours");

hours=sc.nextInt();

minutes=(hours\*60);

System.out.println("Minutes: " + minutes);

}

public void hourstoseconds()

{

System.out.println("enter the no of hours");

hours=sc.nextInt();

seconds=(hours\*3600);

System.out.println("Minutes: " + seconds);

}

}

converter.java

import java.util.\*;

import java.io.\*;

import currencyconversion.\*;

import distanceconversion.\*;

import timeconversion.\*;

class converter

{

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

int choice,ch;

currency c=new currency();

distance d=new distance();

timer t=new timer();

do

{

System.out.println("1.dollar to rupee ");

System.out.println("2.rupee to dollar ");

System.out.println("3.Euro to rupee ");

System.out.println("4..rupee to Euro ");

System.out.println("5.Yen to rupee ");

System.out.println("6.Rupee to Yen ");

System.out.println("7.Meter to kilometer ");

System.out.println("8.kilometer to meter ");

System.out.println("9.Miles to kilometer ");

System.out.println("10.kilometer to miles");

System.out.println("11.Hours to Minutes");

System.out.println("12.Hours to Seconds");

System.out.println("13.Seconds to Hours");

System.out.println("14.Minutes to Hours");

System.out.println("Enter ur choice");

choice=s.nextInt();

switch(choice)

{

case 1:

{

c.dollartorupee();

break;

}

case 2:

{

c.rupeetodollar();

break;

}

case 3:

{

c.eurotorupee();

break;

}

case 4:

{

c.rupeetoeuro();

break;

}

case 5:

{

c.yentorupee();

break;

}

case 6:

{

c.rupeetoyen();

break;

}

case 7:

{

d.mtokm();

break;

}

case 8:

{

d.kmtom();

break;

}

case 9:

{

d.milestokm();

break;

}

case 10 :

{

d.kmtomiles();

break;

}

case 11:

{

t.hourstominutes();

break;

}

case 12:

{

t.hourstoseconds();

break;

}

case 13:

{

t.secondstohours();

break;

}

case 14:

{

t.minutestohours();

break;

}

}

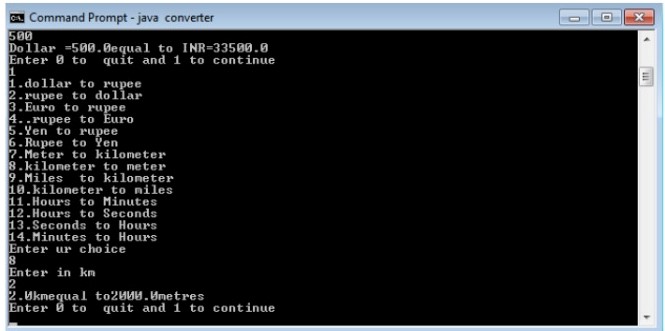
System.out.println("Enter 0 to quit and 1 to continue ");

ch=s.nextInt();

}

while(ch==1);

}

}



3.payslip generation

//Salary.java

import java.util.\*;

import java.util.Scanner;

class Employee

{

int empid;

long mobile;

String name, address, mailid;

Scanner get = new Scanner(System.in);

void getdata()

{

System.out.println("Enter Name of the Employee");

name = get.nextLine();

System.out.println("Enter Mail id");

mailid = get.nextLine();

System.out.println("Enter Address of the Employee:");

address = get.nextLine();

System.out.println("Enter employee id ");

empid = get.nextInt();

System.out.println("Enter Mobile Number");

mobile = get.nextLong();

}

void display()

{

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

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

System.out.println("Mail id : "+mailid);

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

System.out.println("Mobile Number: "+mobile);

}

}

class Programmer extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getprogrammer()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateprog()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR PROGRAMMER");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

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

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Asstprofessor extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getasst()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateasst()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

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

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Associateprofessor extends Employee

{

double salary,bp,da,hra,pf,club,net,gross;

void getassociate()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateassociate()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

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

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

class Professor extends Employee{

double salary,bp,da,hra,pf,club,net,gross;

void getprofessor()

{

System.out.println("Enter basic pay");

bp = get.nextDouble();

}

void calculateprofessor()

{

da=(0.97\*bp);

hra=(0.10\*bp);

pf=(0.12\*bp);

club=(0.1\*bp);

gross=(bp+da+hra);

net=(gross-pf-club);

System.out.println("\*\*\*\*\*\*\*\*");

System.out.println("PAY SLIP FOR PROFESSOR");

System.out.println("\*\*\*\*\*\*\*\*");

System.out.println("Basic Pay: Rs. "+bp);

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

System.out.println("HRA: Rs. "+hra);

System.out.println("PF: Rs. "+pf);

System.out.println("CLUB: Rs. "+club);

System.out.println("GROSS PAY: Rs. "+gross);

System.out.println("NET PAY: Rs. "+net);

}

}

public class Salary

{

public static void main(String[] args){

int choice,cont;

do {

System.out.println("PAYROLL");

System.out.println(" 1.PROGRAMMER \t 2.ASSISTANT PROFESSOR \t 3.ASSOCIATE PROFESSOR \t 4.PROFESSOR ");

Scanner c = new Scanner(System.in);

System.out.print("Enter Your Choice:");

choice=c.nextInt();

switch(choice)

{

case 1:

{

Programmer p=new Programmer();

p.getdata();

p.getprogrammer();

p.display();

p.calculateprog();

break;

}

case 2:

{

Asstprofessor asst=new Asstprofessor();

asst.getdata();

asst.getasst();

asst.display();

asst.calculateasst();

break;

}

case 3:

{

Associateprofessor asso=new Associateprofessor();

asso.getdata();

asso.getassociate();

asso.display();

asso.calculateassociate();

break;

}

case 4:

{

Professor prof=new Professor();

prof.getdata();

prof.getprofessor();

prof.display();

prof.calculateprofessor();

break;

}

}

System.out.print("Please enter 0 to quit and 1 to continue: ");

cont= c.nextInt();

}

while(cont==1);

}

}

Graphical user interface, text

Description automatically generated

4. TEST STACK

import java.io.\*;

interface Stackoperation

{

public void push(int i);

public void pop();

}

class Astack implements Stackoperation

{

int stack[]=new int[5];

int top=-1;

int i;

public void push(int item)

{

if(top>=4)

{

System.out.println("Overflow");

}

else

{

top=top+1;

stack[top]=item;

System.out.print("Element pushed: "+stack[top]);

}

}

public void pop()

{

if(top<0)

System.out.println("Underflow");

else

{

System.out.print("Element popped: "+stack[top]);

top=top-1;

}

}

public void display()

{

if(top<0)

System.out.println("No Element in stack");

else

{

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

System.out.println("Element:"+stack[i]);

}

}

}

class Teststack

{

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

{

int ch,c;

int i;

Astack s=new Astack();

DataInputStream in=new DataInputStream(System.in);

do

{

try

{

System.out.println("ARRAY STACK");

System.out.println("1.Push 2.Pop 3.Display 4.Exit");

System.out.print("Enter your Choice:");

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

switch(ch)

{

case 1:

System.out.print("Enter the value to push:");

i=Integer.parseInt(in.readLine());

s.push(i);

break;

case 2:

s.pop();

break;

case 3:

System.out.println("The elements are: ");

s.display();

break;

default:

break;

}

}

catch(IOException e)

{

System.out.println("IO Error");

}

System.out.println("Please enter 0 to quit and 1 to continue ");

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

}while(c==1);

}

}

Graphical user interface, text

Description automatically generated

5. STRING OPERATION USING ARRAY

//Arraylistexample.java

import java.util.\*;

import java.io.\*;

public class Arraylistexample

{

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

{

ArrayList<String> obj = new ArrayList<String>();

DataInputStream in=new DataInputStream(System.in);

int c,ch;

int i,j;

String str,str1;

do

{

System.out.println("STRING MANIPULATION");

System.out.println("\*\*\*\*\*\*\*\*");

System.out.println("1. Append at end \t 2.Insert at particular index \t 3.Search \t");

System.out.println("4. List string that starting with letter \t");

System.out.println("5. Size \t 6.Remove \t 7.Sort \t 8.Display\t" );

System.out.println("Enter the choice ");

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

switch(c)

{

case 1:

{

System.out.println("Enter the string ");

str=in.readLine();

obj.add(str);

break;

}

case 2:

{

System.out.println("Enter the string ");

str=in.readLine();

System.out.println("Specify the index/position to insert");

i=Integer.parseInt(in.readLine());

obj.add(i-1,str);

System.out.println("The array list has following elements:"+obj);

break;

}

case 3:

{

System.out.println("Enter the string to search ");

str=in.readLine();

j=obj.indexOf(str);

if(j==-1)

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

else

System.out.println("Index of:"+str+"is"+j);

break;

}

case 4:

{

System.out.println("Enter the character to List string that starts with specified character");

str=in.readLine();

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

{

str1=obj.get(i);

if(str1.startsWith(str))

{

System.out.println(str1);

}

}

break;

}

case 5:

{

System.out.println("Size of the list "+obj.size());

break;

}

case 6:

{

System.out.println("Enter the element to remove");

str=in.readLine();

if(obj.remove(str))

{

System.out.println("Element Removed"+str);

}

else

{

System.out.println("Element not present");

}

break;

}

case 7:

{

Collections.sort(obj);

System.out.println("The array list has following elements:"+obj);

break;

}

case 8:

{

System.out.println("The array list has following elements:"+obj);

break;

}

}

System.out.println("Please Enter 0 to break and 1 to continue");

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

}while(ch==1);

}

}

A screenshot of a computer

Description automatically generated with medium confidence

6.ABSTRACT CLASS DECLARATION

import java.util.\*;

abstract class shape

{

int a,b;

abstract public void printarea();

}

class rectangle extends shape

{

public int area\_rect;

public void printarea()

{

Scanner s=new Scanner(System.in);

System.out.println("Enter the length and breadth of rectangle");

a=s.nextInt();

b=s.nextInt();

area\_rect=a\*b;

System.out.println("Length of rectangle: "+a +"breadth of rectangle: "+b);

System.out.println("The area of rectangle is:"+area\_rect);

}

}

class triangle extends shape

{

double area\_tri;

public void printarea()

{

Scanner s=new Scanner(System.in);

System.out.println("Enter the base and height of triangle:");

a=s.nextInt();

b=s.nextInt();

System.out.println("Base of triangle: "+a +"height of triangle: "+b);

area\_tri=(0.5\*a\*b);

System.out.println("The area of triangle is:"+area\_tri);

}

}

class circle extends shape

{

double area\_circle;

public void printarea()

{

Scanner s=new Scanner(System.in);

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

a=s.nextInt();

area\_circle=(3.14\*a\*a);

System.out.println("Radius of circle:"+a);

System.out.println("The area of circle is:"+area\_circle);

}

}

class Shapeclass extends circle

{

public static void main(String[] args)

{

rectangle r=new rectangle();

r.printarea();

triangle t=new triangle();

t.printarea();

circle r1=new circle();

r1.printarea();

}

}

Graphical user interface, text, application

Description automatically generated

7.USER DEFINED EXCEPTION

//userdefined.java

import java.util.\*;

class NegativeAmtException extends Exception

{

String msg;

NegativeAmtException(String msg)

{

this.msg=msg;

}

public String toString()

{

return msg;

}

}

public class userdefined

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

System.out.print("Enter Amount:");

int a=s.nextInt();

try

{

if(a<0)

{

throw new NegativeAmtException("Invalid Amount");

}

System.out.println("Amount Deposited");

}

catch(NegativeAmtException e)

{

System.out.println(e);

}

}

}

Graphical user interface, text

Description automatically generated

8.FILE HANDLING IN JAVA

import java.io.\*;

import java.util.\*;

class filedemo

{

public static void main(String args[])

{

String filename;

Scanner s=new Scanner(System.in);

System.out.println("Enter the file name ");

filename=s.nextLine();

File f1=new File(filename);

System.out.println("\*\*\*\*\*\*");

System.out.println("FILE INFORMATION");

System.out.println("\*\*\*\*\*\*");

System.out.println("NAME OF THE FILE "+f1.getName());

System.out.println("PATH OF THE FILE "+f1.getPath());

System.out.println("PARENT"+f1.getParent());

if(f1.exists())

System.out.println("THE FILE EXISTS ");

else

System.out.println("THE FILE DOES NOT ExISTS ");

if(f1.canRead())

System.out.println("THE FILE CAN BE READ ");

else

System.out.println("THE FILE CANNOT BE READ ");

if(f1.canWrite())

System.out.println("WRITE OPERATION IS PERMITTED");

else

System.out.println("WRITE OPERATION IS NOT PERMITTED");

if(f1.isDirectory())

System.out.println("IT IS A DIRECTORY ");

else

System.out.println("NOT A DIRECTORY");

if(f1.isFile())

System.out.println("IT IS A FILE ");

else

System.out.println("NOT A FILE");

System.out.println("File last modified "+ f1.lastModified());

System.out.println("LENGTH OF THE FILE "+f1.length());

System.out.println("FILE DELETED "+f1.delete());

}

}

Shape

Description automatically generated with low confidence

9.MULTI THREADING IMPLEMENTATION

import java.util.\*;

class even implements Runnable

{

public int x;

public even(int x)

{

this.x = x;

}

public void run()

{

System.out.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x \* x);

}

}

class odd implements Runnable

{

public int x;

public odd(int x)

{

this.x = x;

}

public void run()

{

System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x \* x \* x);

}

}

class A extends Thread

{

public void run()

{

int num = 0;

Random r = new Random();

try

{

for (int i = 1; i < 6; i++)

{

num = r.nextInt(100);

System.out.println("Main Thread and Generated Number is " + num);

if (num % 3 == 0)

{

Thread t1 = new Thread(new even(num));

t1.start();

}

else

{

Thread t2 = new Thread(new odd(num));

t2.start();

}

Thread.sleep(1000);

System.out.println("--------------------------------------");

}

}

catch (Exception ex)

{

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

}

}

}

public class multithreadprog

{

public static void main(String[] args)

{

A a = new A();

a.start();

}

}

Shape

Description automatically generated with medium confidence

10.GENERIC FUNCTION IMPLEMENTATION

//genericdemo.java

class MyClass<T extends Comparable<T>>

{

T[] vals;

MyClass(T[] o)

{

vals = o;

}

public T min()

{

T v = vals[0];

for(int i=1; i < vals.length; i++)

if(vals[i].compareTo(v) < 0)

v = vals[i];

return v;

}

public T max()

{

T v = vals[0];

for(int i=1; i < vals.length;i++)

if(vals[i].compareTo(v) > 0)

v = vals[i];

return v;

}

}

public class genericdemo

{

public static void main(String args[]) {

int i;

Integer inums[]={3,5,4,4,6,1};

Character chs[]={'v','s','t','a','n','h'};

Double d[]={20.2,45.4,71.6,78.3,54.6,8.4};

MyClass<Integer> iob = new MyClass<Integer>(inums);

MyClass<Character> cob = new MyClass<Character>(chs);

MyClass<Double>dob = new MyClass<Double>(d);

System.out.println("Max value in inums: " + iob.max());

System.out.println("Min value in inums: " + iob.min());

System.out.println("Max value in chs: " + cob.max());

System.out.println("Min value in chs: " + cob.min());

System.out.println("Max value in chs: " + dob.max());

System.out.println("Min value in chs: " + dob.min());

}

}

Shape

Description automatically generated with medium confidence

12.READ EXCEL FILE

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.FormulaEvaluator;

import org.apache.poi.ss.usermodel.Row;

public class ReadExcelFileDemo

{

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

{

//obtaining input bytes from a file

FileInputStream fis=new FileInputStream(new File("C:\\demo\\student.xls"));

//creating workbook instance that refers to .xls file

HSSFWorkbook wb=new HSSFWorkbook(fis);

//creating a Sheet object to retrieve the object

HSSFSheet sheet=wb.getSheetAt(0);

//evaluating cell type

FormulaEvaluator formulaEvaluator=wb.getCreationHelper().createFormulaEvaluator();

for(Row row: sheet) //iteration over row using for each loop

{

for(Cell cell: row) //iteration over cell using for each loop

{

switch(formulaEvaluator.evaluateInCell(cell).getCellType())

{

case Cell.CELL\_TYPE\_NUMERIC: //field that represents numeric cell type

//getting the value of the cell as a number

System.out.print(cell.getNumericCellValue()+ "\t\t");

break;

case Cell.CELL\_TYPE\_STRING: //field that represents string cell type

//getting the value of the cell as a string

System.out.print(cell.getStringCellValue()+ "\t\t");

break;

}

}

System.out.println();

}

}

}

Graphical user interface, text, application

Description automatically generated