**CODE**:

import java.util.\*;

class tree

{

public double hd, p;

char ch;

String parent;

tree(char c,int x)

{

ch=c;

if(x==1)

parent="outlook";

else if(x==2)

parent="temperature";

else if(x==3)

parent="humidity";

else if(x==4)

parent="windy";

}//tree function

}//class tree

class d\_tree

{

static char outlook[]={'S','S','O','R','R','R','O','S','S','R','S','O','O','R'};

static char temperature[]={'B','B','B','M','C','C','C','M','C','M','M','M','B','M'};

static char humidity[]={'H','H','H','H','N','N','N','H','N','N','N','H','N','H'};

static char windy[]={'F','T','F','F','F','T','T','F','F','F','T','T','F','T'};

static char class1[]={'N','N','P','P','P','N','P','N','P','P','P','P','P','N'};

static double G1,G2,G3,G4, HD, temp1,temp2;

static double play=9.0/14.0;

static double nplay=5.0/14.0;

static int row=0,column=0;

static char classify[][]=new char[10][10];

static tree cal\_hd(tree t,int choice)

{

double count1=0,count2=0;

if(choice==1)

{

for(int i=0;i<14;++i)

{

if(t.ch==outlook[i] && class1[i]=='P')

++count1;

else if(t.ch==outlook[i] && class1[i]=='N')

++count2;

}//for

}//if

if(choice==2)

{

for(int i=0;i<14;++i)

{

if(t.ch==temperature[i] && class1[i]=='P')

++count1;

else if(t.ch==temperature[i] && class1[i]=='N')

++count2;

}//for

}//if

if(choice==3)

{

for(int i=0;i<14;++i)

{

if(t.ch==humidity[i] && class1[i]=='P')

++count1;

else if(t.ch==humidity[i] && class1[i]=='N')

++count2;

}//for

}//if

if(choice==4)

{

for(int i=0;i<14;++i)

{

if(t.ch==windy[i] && class1[i]=='P')

++count1;

else if(t.ch==windy[i] && class1[i]=='N')

++count2;

}//for

}//if

temp1=count1/(count1+count2);

temp2=count2/(count1+count2);

t.p=(count1+count2)/14;

if(temp1==0 || temp2==0)

t.hd=0;

else

t.hd=temp1\*(Math.log(1/temp1)/Math.log(10))+temp2\*(Math.log(1/temp2)/Math.log(10));

return t;

}//cal\_hd function ends here

static int check\_if\_equal(char ca[][],int cnt)

{

int count1=0,count2=0,count3=0;

char c1=ca[0][0],c2=ca[0][1],c3=ca[0][2];

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

{

if(ca[i][0]==c1)

++count1;

if(ca[i][1]==c2)

++count2;

if(ca[i][2]==c3)

++count3;

}//for

if(count1==cnt)

return 0;

else if(count2==cnt)

return 1;

else if(count3==cnt)

return 2;

else

return -1;

}//check\_if\_equal function ends here

static void compute(char ch)

{

char ca1[][]=new char[10][10];

char ca2[][]=new char[10][10];

char ca3[][]=new char[10][10];

int cnt1=0;

int cnt2=0;

for(int i=0;i<14;++i)

{

if(outlook[i]==ch)

{

ca1[cnt1][0]=temperature[i];

ca1[cnt1][1]=humidity[i];

ca1[cnt1][2]=windy[i];

ca1[cnt1][3]=class1[i];

++cnt1;

}//if

}//for

int check1=0,check2=0;

for(int i=0;i<cnt1;++i)

{

if(ca1[i][3]=='P')

++check1;

else if(ca1[i][3]=='N')

++check2;

if(check1==cnt1)

{

classify[row][column++]=ch;

classify[row][column++]='P';

System.out.print("--------->PLAY");

return;

}//if

else if(check2==cnt1)

{

classify[row][column++]=ch;

classify[row][column++]='N';

System.out.print("--------->NO PLAY");

return;

}//else if

}//end of for loop

cnt1=0;

for(int i=0;i<10;++i)

{

if(ca1[i][3]=='P')

{

ca2[cnt1][0]=ca1[i][0];

ca2[cnt1][1]=ca1[i][1];

ca2[cnt1][2]=ca1[i][2];

ca2[cnt1][3]=ca1[i][3];

++cnt1;

}//if

}//for

classify[row][column++]=ch;

int z=check\_if\_equal(ca2,cnt1);

if(z==0)

{

System.out.print("\n--------->TEMPERATURE("+ca2[0][0]+")--------->PLAY\n");

classify[row][column++]=ca2[0][0];

classify[row][column++]='P';

}//if

else if(z==1)

{

System.out.print("\n--------->HUMIDITY("+ca2[0][1]+")--------->PLAY\n");

classify[row][column++]=ca2[0][1];

classify[row][column++]='P';

}//else if

else if(z==2)

{

System.out.print("\n--------->WINDY("+ca2[0][2]+")--------->PLAY\n");

classify[row][column++]=ca2[0][2];

classify[row][column++]='P';

}//else if

cnt1=0;

for(int i=0;i<10;++i)

{

if(ca1[i][3]=='N')

{

ca3[cnt1][0]=ca1[i][0];

ca3[cnt1][1]=ca1[i][1];

ca3[cnt1][2]=ca1[i][2];

ca3[cnt1][3]=ca1[i][3];

++cnt1;

}//if

}//for

++row;

column=0;

classify[row][column++]=ch;

z=check\_if\_equal(ca3,cnt1);

if(z==0)

{

System.out.print("--------->TEMPERATURE("+ca3[0][0]+")--------->NO PLAY");

classify[row][column++]=ca3[0][0];

classify[row][column++]='N';

}//if

else if(z==1)

{

System.out.print("--------->HUMIDITY("+ca3[0][1]+")--------->NO PLAY");

classify[row][column++]=ca3[0][1];

classify[row][column++]='N';

}//if

else if(z==2)

{

System.out.print("--------->WINDY("+ca3[0][2]+")--------->NO PLAY");

classify[row][column++]=ca3[0][2];

classify[row][column++]='N';

}//if

}//compute function

public static void main(String args[])

{

Scanner scr=new Scanner(System.in);

HD=play\*(Math.log(1/play)/Math.log(10))+nplay\*(Math.log(1/nplay)/Math.log(10));

System.out.println("\nThe value of H(D) is "+HD);

tree sunny=new tree('S',1);

tree overcast=new tree('O',1);

tree rain=new tree('R',1);

tree hot=new tree('B',2);

tree mild=new tree('M',2);

tree cool=new tree('C',2);

tree high=new tree('H',3);

tree normal=new tree('N',3);

tree tru=new tree('T',4);

tree fal=new tree('F',4);

sunny=cal\_hd(sunny,1);

overcast=cal\_hd(overcast,1);

rain=cal\_hd(rain,1);

hot=cal\_hd(hot,2);

mild=cal\_hd(mild,2);

cool=cal\_hd(cool,2);

high=cal\_hd(high,3);

normal=cal\_hd(normal,3);

tru=cal\_hd(tru,4);

fal=cal\_hd(fal,4);

G1=HD-(sunny.p\*sunny.hd+overcast.p\*overcast.hd+rain.p\*rain.hd);

G2=HD-(hot.p\*hot.hd+mild.p\*mild.hd+cool.p\*cool.hd);

G3=HD-(high.p\*high.hd+normal.p\*normal.hd);

G4=HD-(tru.p\*tru.hd+fal.p\*fal.hd);

int gflag=0;

System.out.println("\nGain(D,Outlook)= "+G1 +"\nGain(D,Temperature)= "+G2+"\nGain(D,Humidity)= "+G3+"\nGain(D,Windy)= "+G4);

System.out.print("\nThe Splitting factor is ");

if(G1>G2 && G1>G3 && G1>G4)

System.out.println("Outlook");

else if(G2>G1 && G2>G3 && G2>G4)

System.out.println("Temperature");

else if(G3>G2 && G3>G1 && G3>G4)

System.out.println("Humidity");

else if(G4>G2 && G4>G3 && G4>G1)

System.out.println("Windy");

System.out.println("\nThe Tree is as follows:-");

System.out.print("\nOUTLOOK(S)");

compute('S');

++row;

column=0;

System.out.print("\n\n\nOUTLOOK(O)");

compute('O');

++row;

column=0;

System.out.print("\n\n\nOUTLOOK(R)");

compute('R');

char input[]=new char[4];

String s;

System.out.println("\nMenu:\n\nOutlook: Sunny=S Overcast=O Rainy=R\n\nTemperature: Hot=B Medium=M Cool=C\n\nHumidity: High=H Normal=N\n\nWindy: True=T False=F");

System.out.println("\n\nEnter your new tuple to be classified ");

System.out.print("\nOutlook(S/O/R)= ");

s=scr.nextLine();

input[0]=s.charAt(0);

System.out.print("\nTemperature(B/M/C)= ");

s=scr.nextLine();

input[1]=s.charAt(0);

System.out.print("\nHumidity(H/N)= ");

s=scr.nextLine();

input[2]=s.charAt(0);

System.out.print("\nWindy(T/F)= ");

s=scr.nextLine();

input[3]=s.charAt(0);

System.out.print("Your input is ");

for(int i=0;i<4;++i)

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

int inc1=0;

do

{

if(input[0]==classify[inc1][0])

{

if(classify[inc1][1]=='P')

{

System.out.println("\n\nTuple classified as PLAY");

break;

}//if

else if(input[1]==classify[inc1][1] || input[2]==classify[inc1][1] || input[3]==classify[inc1][1])

{

if(classify[inc1][2]=='P')

{

System.out.println("\n\nTuple classified as PLAY");

break;

}//if

else if(classify[inc1][2]=='N')

{

System.out.println("\n\nTuple classified as NO PLAY");

break;

}//else if

}//outer else if

}//outer if

++inc1;

}//do loop

while(true);

}//main

};//class d\_tree ends here

**output:**

**d50112@d50112-ThinkCentre-M720t:~$** javac d\_tree.java

**d50112@d50112-ThinkCentre-M720t:~$** java d\_tree

The value of H(D) is 0.2830542780615224

Gain(D,Outlook)= 0.07427909717678749

Gain(D,Temperature)= 0.00879686881360553

Gain(D,Humidity)= 0.04570704031674405

Gain(D,Windy)= 0.014487679755121663

The Splitting factor is Outlook

The Tree is as follows:-

OUTLOOK(S)

--------->HUMIDITY(N)--------->PLAY

--------->HUMIDITY(H)--------->NO PLAY

OUTLOOK(O)--------->PLAY

OUTLOOK(R)

--------->WINDY(F)--------->PLAY

--------->WINDY(T)--------->NO PLAY

Menu:

Outlook: Sunny=S Overcast=O Rainy=R

Temperature: Hot=B Medium=M Cool=C

Humidity: High=H Normal=N

Windy: True=T False=F

Enter your new tuple to be classified

Outlook(S/O/R)= S

Temperature(B/M/C)= M

Humidity(H/N)= H

Windy(T/F)= T

Your input is S M H T

Tuple classified as NO PLAY

**d50112@d50112-ThinkCentre-M720t:~$** java d\_tree

The value of H(D) is 0.2830542780615224

Gain(D,Outlook)= 0.07427909717678749

Gain(D,Temperature)= 0.00879686881360553

Gain(D,Humidity)= 0.04570704031674405

Gain(D,Windy)= 0.014487679755121663

The Splitting factor is Outlook

The Tree is as follows:-

OUTLOOK(S)

--------->HUMIDITY(N)--------->PLAY

--------->HUMIDITY(H)--------->NO PLAY

OUTLOOK(O)--------->PLAY

OUTLOOK(R)

--------->WINDY(F)--------->PLAY

--------->WINDY(T)--------->NO PLAY

Menu:

Outlook: Sunny=S Overcast=O Rainy=R

Temperature: Hot=B Medium=M Cool=C

Humidity: High=H Normal=N

Windy: True=T False=F

Enter your new tuple to be classified

Outlook(S/O/R)= S

Temperature(B/M/C)= C

Humidity(H/N)= N

Windy(T/F)= F

Your input is S C N F

Tuple classified as PLAY