DAA PROJECT CODE: FLIGHT AGENDA AYUSH HATE 1CR15CS036

Problem Statement: **Flight Agenda**

A travel agent requests software for making an agenda of flights for clients. The agent has access to a data base with all airports and flights. Besides the flight number, origin airport and destination, the flights have departure and arrival time. Specifically the agent wants to determine the **earliest arrival time** for the destination given an **origin airport** and **start time**.

```
package myPackage;

import java.util.*;

class Stack{
      int top=-1;
      int stackArray[]=new int[8];
      void push(int x)
      {
            stackArray[++top]=x;
      }
      int pop()
      {
            if(top==-1)
                  return 0;
            return stackArray[top--];
      }
}

class ArrDepData{
      String Airline[]=new String[8];
      int flightNumber[]=new int[8];
      long DepartureTime[]=new long[8];
      long ArrivalTime[]=new long[8];
      ArrDepData(String A[],int flno[],long DT[],long AT[])
      {
            Airline=A;
            flightNumber=flno;
            DepartureTime=DT;
            ArrivalTime=AT;
      }
}

class Time{
      long MinutetoMins(long x)
```

```
        {
                return(x%60);
        }
        long MinutetoHrs(long x)
        {
                return(x/60);
        }
        long HourstoMins(long x)
        {
                return(x*60);
        }
}

class VertexNames{
        String VertexNames[]=new String[8];
        VertexNames()
        {
                VertexNames[0]="DEL";
                VertexNames[1]="BOM";
                VertexNames[2]="MAA";
                VertexNames[3]="BLR";
                VertexNames[4]="HYD";
                VertexNames[5]="GOI";
                VertexNames[6]="CCU";
                VertexNames[7]="COK";
        }
        int getAirportasIndex(String DepAirpt)
        {
                int i=0;
                while(VertexNames[i].equalsIgnoreCase(DepAirpt)==false)
                {
                        i++;
                }
                return i;
        }
}

public class Flights {
        public static int tot_nodes=8;
        public static int tot_edges=20;
        public static int path[]=new int[10];//priority queue containing vertex index
        static Scanner s=new Scanner(System.in);
        static VertexNames AIRPORT=new VertexNames();
        static Time TimeConverter=new Time();
```

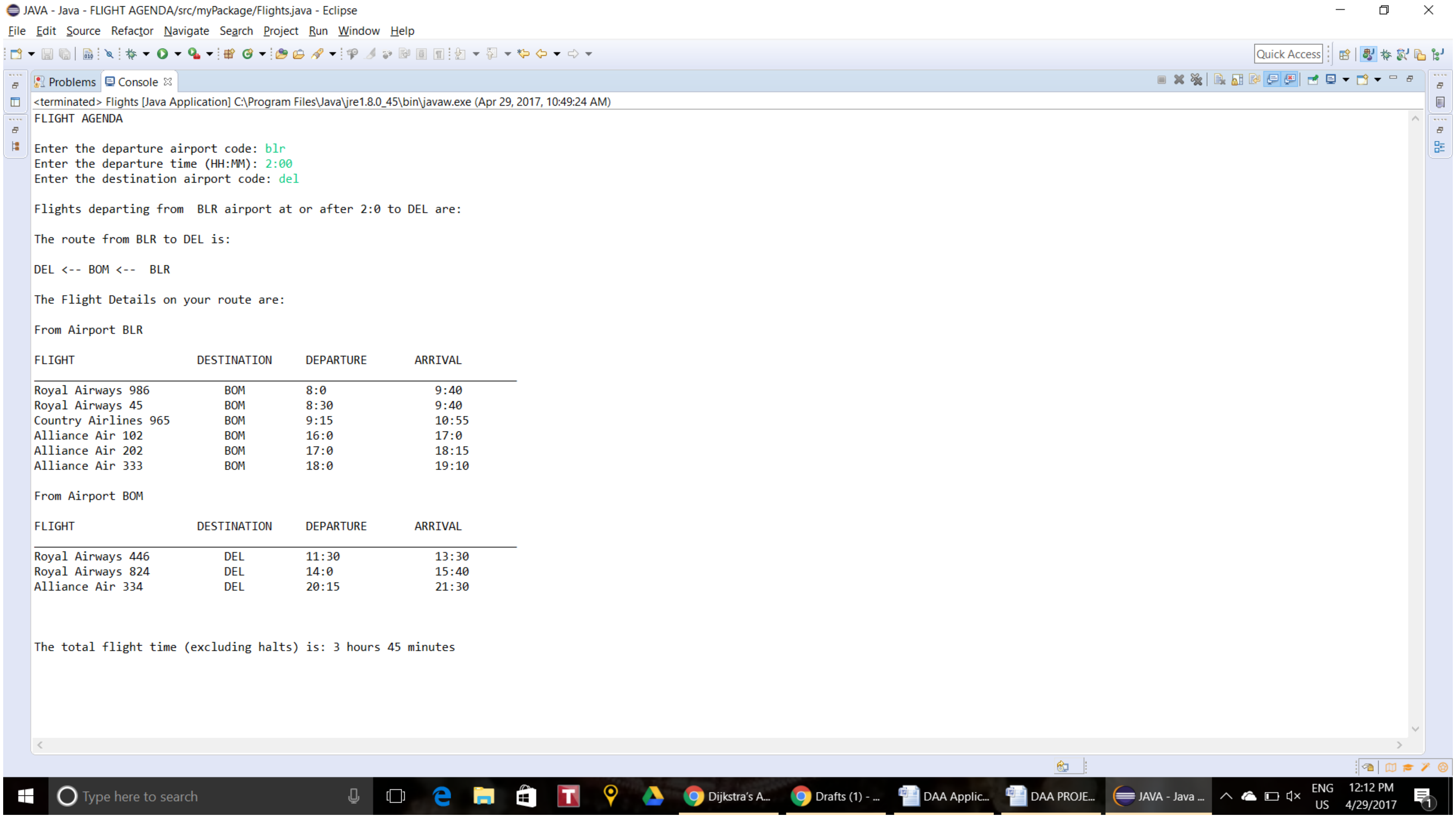
```
    static ArrDepData Schedule[]=new ArrDepData[8];
    static Stack Buffer=new Stack();
    static long MinimumTime;
    public static void main(String[] args){
        int i,j;
        long cost[][]=new long[8][8];//cost adjacency matrix
        long dist[]=new long[8];
        String StartTimeString;
        long StartTimeReader[]=new long[2];
        long startH,startM;
        long startT;
        String DepartureAirport;
        String ArrivalAirport;
        System.out.print("FLIGHT AGENDA\n\n");
        create(cost);
        System.out.print("Enter the departure airport code: ");
        DepartureAirport=s.next();
        i=AIRPORT.getAirportasIndex(DepartureAirport);
        System.out.print("Enter the departure time (HH:MM): ");
        StartTimeString=s.next();
        StringTokenizer SplitTime=new StringTokenizer(StartTimeString,":");
        int k=0;
        while(SplitTime.hasMoreTokens())
        {
            StartTimeReader[k]=Long.parseLong(SplitTime.nextToken());
            k++;
        }
        startH=StartTimeReader[0];
        startM=StartTimeReader[1];
        MinimumTime=startT=TimeConverter.HourstoMins(startH)+startM;
        System.out.print("Enter the destination airport code: ");
        ArrivalAirport=s.next();
        int A=AIRPORT.getAirportasIndex(ArrivalAirport);
        System.out.println("\nFlights departing from  "+(AIRPORT.VertexNames[i])+" airport at or after "+startH+":"+startM+" to 
"+(AIRPORT.VertexNames[A])+" are: \n");
        j=A;
        Dijkstra(cost,i,dist);
        if(dist[i]==1441)//24 hrs and 1 minute is infinity
            System.out.println("\nNo Path from "+AIRPORT.VertexNames[i]+" to "+AIRPORT.VertexNames[j]);
        else
            display(i,j,dist);
    }
    public static void create(long cost[][])//initialize the adjacency matrix
    {
```

```
int i,j;
String Airline[];
int flightNumber[];
long DepartureTime[];
long ArrivalTime[];
for(i=0;i<tot_nodes;i++)
{
        for(j=0;j<tot_nodes;j++)
        {
                if(i==j)
                        cost[i][j]=0;
                else
                        cost[i][j]=1441;//infinity
        }
}
cost[0][1]=cost[1][0]=125;
cost[0][6]=cost[6][0]=135;
cost[1][2]=cost[2][1]=120;
cost[1][3]=cost[3][1]=100;
cost[1][5]=cost[5][1]=75;
cost[2][3]=cost[3][2]=60;
cost[2][4]=cost[4][2]=75;
cost[3][5]=cost[5][3]=75;
cost[3][7]=cost[7][3]=70;
cost[4][6]=cost[6][4]=130;
Airline=new String[] {"Alliance Air","Royal Airways","Alliance Air"};
flightNumber=new int[] {784,486,777,-1};
DepartureTime=new long[] {630,1050,1080};
ArrivalTime=new long[] {765,1180,1215};
Schedule[6]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
Airline=new String[] {"Royal Airways","Royal Airways","Royal Airways","Alliance Air"};
flightNumber=new int[] {433,223,213,197,-1};
DepartureTime=new long[] {420,720,1020,1320};
ArrivalTime=new long[] {490,790,1090,1390};
Schedule[7]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
Airline=new String[] {"Country Airlines", "Royal Airways","Alliance Air", "Royal Airways"};
flightNumber=new int[] {566,311,259,448,-1};
DepartureTime=new long[] {420,480,660,870};
ArrivalTime=new long[] {495,610,735,1000};
Schedule[4]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
Airline=new String[] {"Alliance Air","Alliance Air","Royal Airways","Royal Airways","Alliance Air"};
flightNumber=new int[] {648,448,742,445,287,-1};
DepartureTime=new long[] {240,270,720,810,1065};
ArrivalTime=new long[] {365,405,845,945,1190};
```

```
        Schedule[0]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
        Airline=new String[] {"Country Airlines","Alliance Air","Royal Airways","Royal Airways","Alliance Air"};
        flightNumber=new int[] {124,667,446,824,334,-1};
        DepartureTime=new long[] {300,360,690,840,1215};
        ArrivalTime=new long[] {425,485,810,940,1290};
        Schedule[1]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
        Airline=new String[] {"Country Airlines", "Alliance Air","Country Airlines","Country Airlines","Royal Airways"};
        flightNumber=new int[] {156,187,934,438,555,-1};
        DepartureTime=new long[] {480,600,1050,1125,1330};
        ArrivalTime=new long[] {555,675,1125,1200,1405};
        Schedule[5]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
        Airline=new String[] {"Alliance Air","Royal Airways", "Country Airlines", "Alliance Air","Royal Airways","Alliance Air"};
        flightNumber=new int[] {789,963,846,748,225,499,-1};
        DepartureTime=new long[] {470,480,660,840,1050,1290};
        ArrivalTime=new long[] {590,540,720,900,1125,1365};
        Schedule[2]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
        Airline=new String[] {"Royal Airways","Royal Airways","Country Airlines", "Alliance Air","Alliance Air","Alliance Air"};
        flightNumber=new int[] {986,45,965,102,202,333,-1};
        DepartureTime=new long[] {480,510,555,960,1020,1080};
        ArrivalTime=new long[] {580,580,655,1020,1095,1150};
        Schedule[3]=new ArrDepData(Airline,flightNumber,DepartureTime,ArrivalTime);
    }
    public static void Dijkstra(long[][] cost, int source, long[] dist)
    {
        int i,j,v1,v2;
        long minD;
        int src[]=new int[10];
        for(i=0;i<tot_nodes;i++)
        {
            dist[i]=cost[source][i];//initially put distance(time) from source to i
            src[i]=0;
            path[i]=source;
        }
        src[source]=1;//visited
        for(i=1;i<tot_nodes;i++)//i=1
        {
            minD=1441;//initialize minimum distance to max
            v1=-1;//reset previous value of source;
            for(j=0;j<tot_nodes;j++)
            {
                if(src[j]==0)//unvisited
                {
                    if(dist[j]<minD)
                    {
                        minD=dist[j];
                        v1=j;
                    }
                }
            }
            src[v1]=1;
            for(v2=0;v2<tot_nodes;v2++)
            {
                if(src[v2]==0)
                {
                    if((dist[v1]+cost[v1][v2])<dist[v2])
                    {
                        dist[v2]=dist[v1]+cost[v1][v2];//path is from source to v1 to v2
                        path[v2]=v1;//path is via v1
                    }
                }
            }
        }
    }
    public static void display(int Source,int Destination,long dist[])
    {
        int i;
        System.out.println("The route from "+AIRPORT.VertexNames[Source]+" to "+AIRPORT.VertexNames[Destination]+" is: \n");
        for(i=Destination;i!=Source;i=path[i])
        {
            System.out.print(AIRPORT.VertexNames[i]+" <-- ");
            Buffer.push(i);
        }
        System.out.println(" "+AIRPORT.VertexNames[i]);
        Buffer.push(i);
        System.out.println("\nThe Flight Details on your route are: \n");
        showData(Destination);
        System.out.println("\nThe total flight time (excluding halts) is: "+TimeConverter.MinutetoHrs(dist[Destination])+" hours 
"+TimeConverter.MinutetoMins(dist[Destination])+" minutes");
    }
    public static void showData(int dest)
    {
        int i=Buffer.pop();
        Stack StackToObtainArrivalTime=new Stack();
        while(i!=dest)
        {
            System.out.println("From Airport 
"+AIRPORT.VertexNames[i]+"\n\nFLIGHT\t\t\tDESTINATION\tDEPARTURE\tARRIVAL\n__________________________________________________________________
______");
```

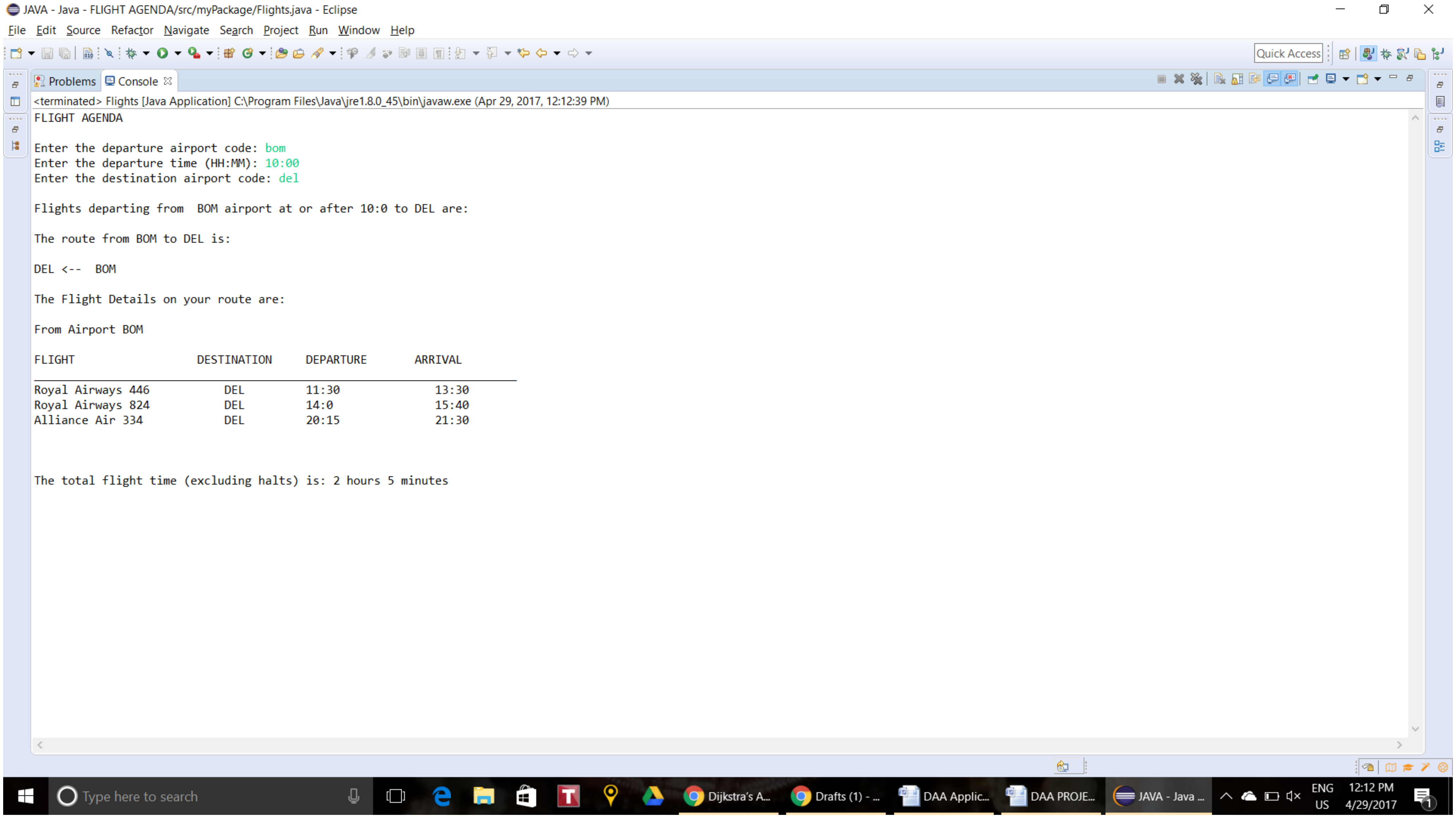
```
            for(int j=0;Schedule[i].flightNumber[j]!=-1;j++)
            {
                int k=Buffer.pop();
                Buffer.push(k);
                if(Schedule[i].DepartureTime[j]<MinimumTime)
                    continue;
                StackToObtainArrivalTime.push(j);
                System.out.println(Schedule[i].Airline[j]+" "+Schedule[i].flightNumber[j]+"\t
"+AIRPORT.VertexNames[k]+"\t\t"+TimeConverter.MinutetoHrs(Schedule[i].DepartureTime[j])+":"+TimeConverter.MinutetoMins(Schedule[i].Departure
Time[j])+"\t\t   "+TimeConverter.MinutetoHrs(Schedule[i].ArrivalTime[j])+":"+TimeConverter.MinutetoMins(Schedule[i].ArrivalTime[j]));
            }
            System.out.println();
            int LIMIT=0;
            while(StackToObtainArrivalTime.top!=-1)
            {
                LIMIT=StackToObtainArrivalTime.pop();
            }
            MinimumTime=Schedule[i].ArrivalTime[LIMIT];
            i=Buffer.pop();
        }
        System.out.println();
        Buffer.pop();
    }
}
```

SAMPLE OUTPUT RUNS:



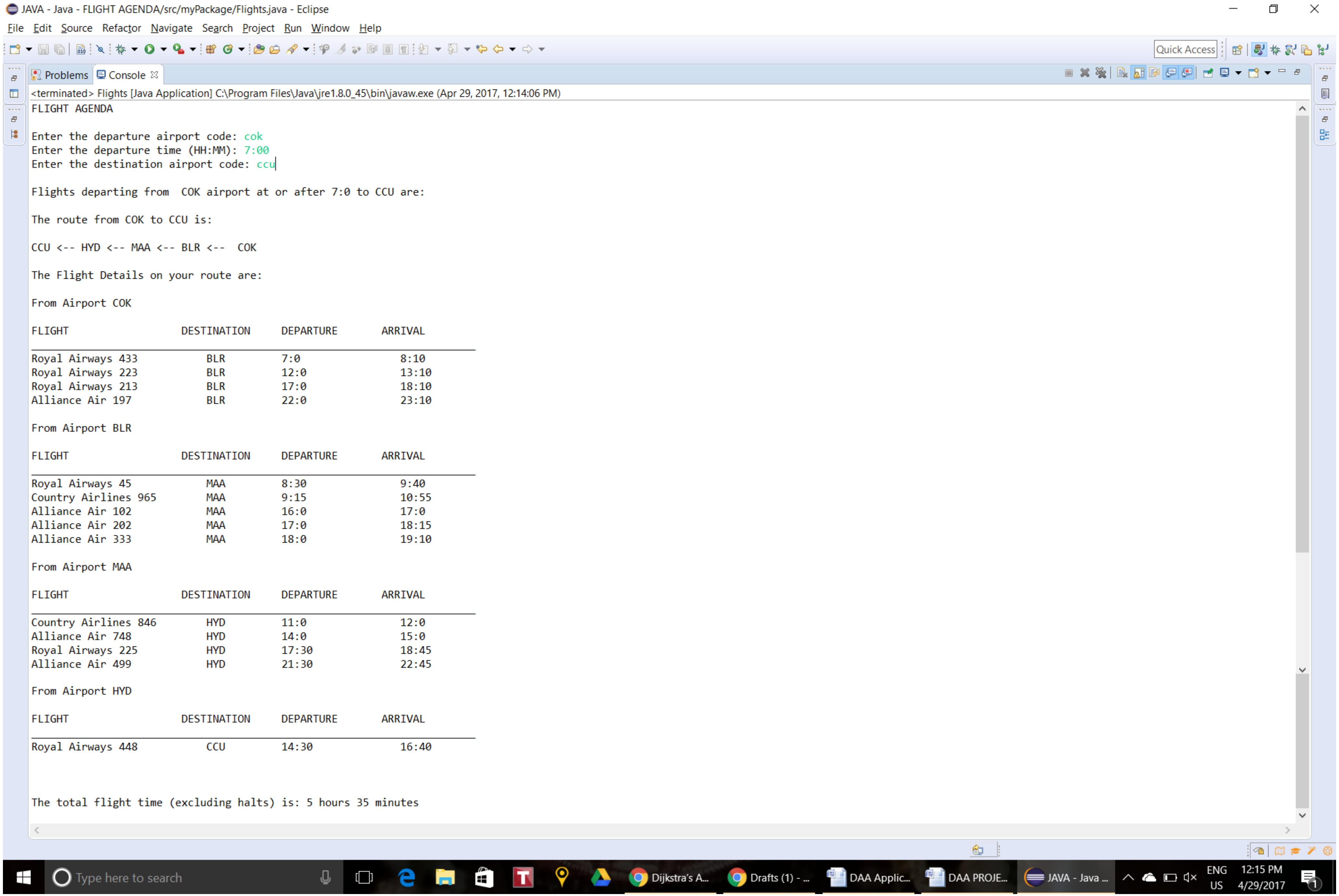
<terminated> Flights [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2017, 10:49:24 AM)

```
FLIGHT AGENDA

Enter the departure airport code: blr
Enter the departure time (HH:MM): 2:00
Enter the destination airport code: del

Flights departing from  BLR airport at or after 2:0 to DEL are:

The route from BLR to DEL is:

DEL <-- BOM <--  BLR

The Flight Details on your route are:

From Airport BLR

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 986           BOM         8:0                9:40
Royal Airways 45            BOM         8:30               9:40
Country Airlines 965        BOM         9:15               10:55
Alliance Air 102            BOM         16:0               17:0
Alliance Air 202            BOM         17:0               18:15
Alliance Air 333            BOM         18:0               19:10

From Airport BOM

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 446           DEL         11:30              13:30
Royal Airways 824           DEL         14:0               15:40
Alliance Air 334            DEL         20:15              21:30



The total flight time (excluding halts) is: 3 hours 45 minutes
```



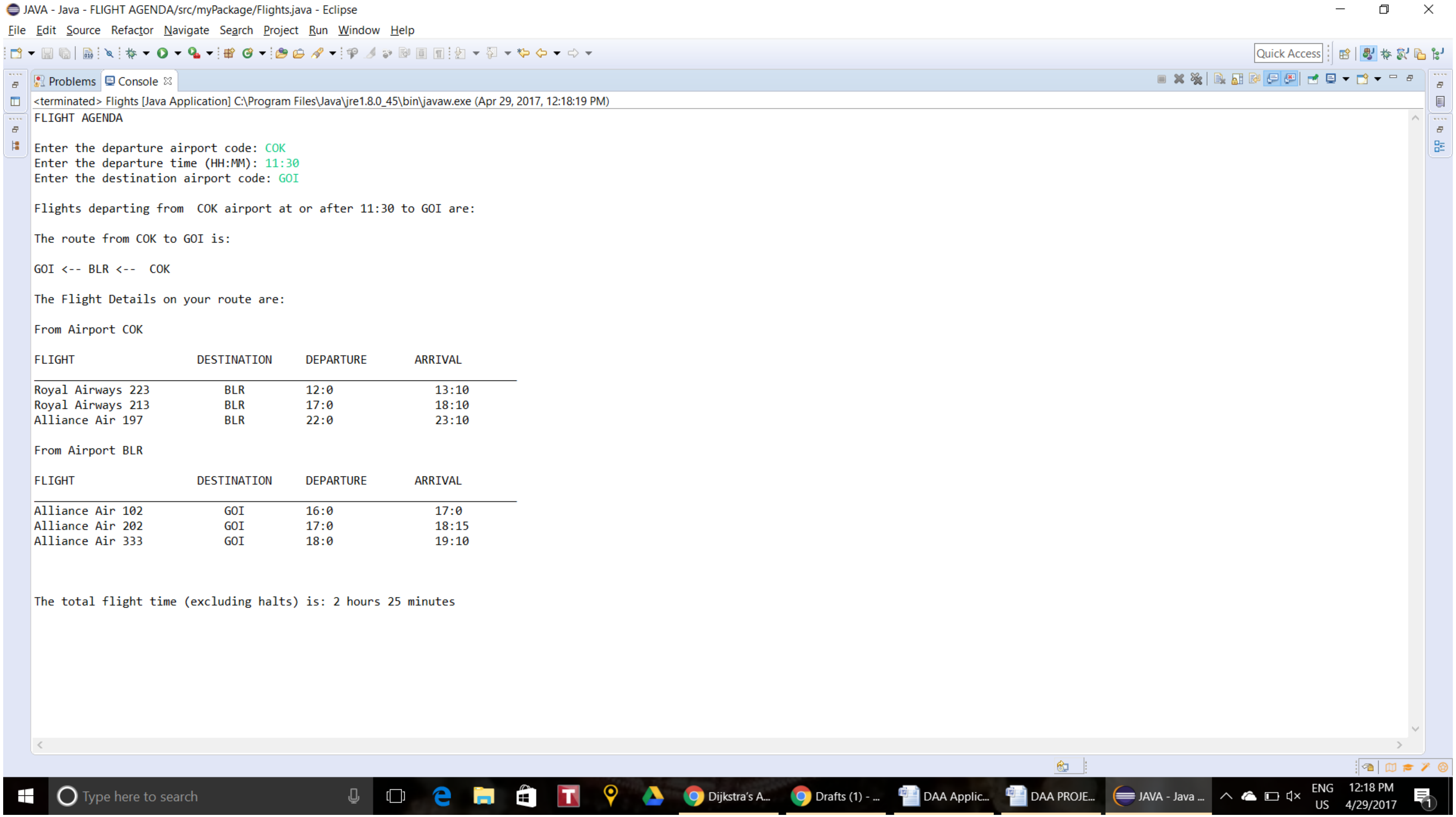
JAVA - Java - FLIGHT AGENDA/src/myPackage/Flights.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Console

```
<terminated> Flights [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2017, 12:12:39 PM)
FLIGHT AGENDA

Enter the departure airport code: bom
Enter the departure time (HH:MM): 10:00
Enter the destination airport code: del

Flights departing from  BOM airport at or after 10:0 to DEL are:

The route from BOM to DEL is:

DEL <--  BOM

The Flight Details on your route are:

From Airport BOM

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 446           DEL         11:30              13:30
Royal Airways 824           DEL         14:0               15:40
Alliance Air 334            DEL         20:15              21:30



The total flight time (excluding halts) is: 2 hours 5 minutes
```



<terminated> Flights [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2017, 12:14:06 PM)

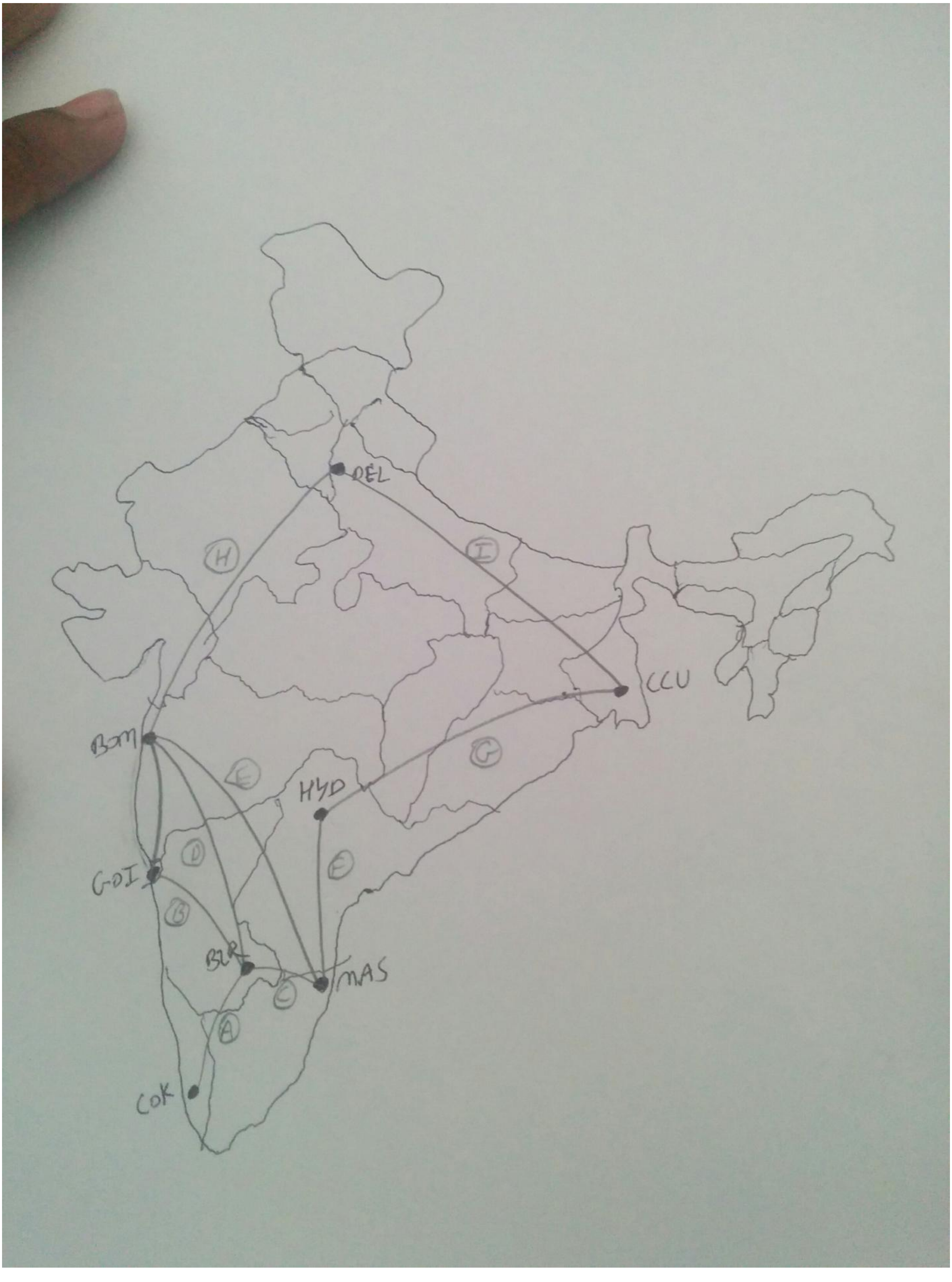
```
FLIGHT AGENDA

Enter the departure airport code: cok
Enter the departure time (HH:MM): 7:00
Enter the destination airport code: ccu

Flights departing from  COK airport at or after 7:0 to CCU are:

The route from COK to CCU is:

CCU <-- HYD <-- MAA <-- BLR <--  COK

The Flight Details on your route are:

From Airport COK

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 433           BLR         7:0                8:10
Royal Airways 223           BLR         12:0               13:10
Royal Airways 213           BLR         17:0               18:10
Alliance Air 197            BLR         22:0               23:10

From Airport BLR

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 45            MAA         8:30               9:40
Country Airlines 965        MAA         9:15               10:55
Alliance Air 102            MAA         16:0               17:0
Alliance Air 202            MAA         17:0               18:15
Alliance Air 333            MAA         18:0               19:10

From Airport MAA

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Country Airlines 846        HYD         11:0               12:0
Alliance Air 748            HYD         14:0               15:0
Royal Airways 225           HYD         17:30              18:45
Alliance Air 499            HYD         21:30              22:45

From Airport HYD

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 448           CCU         14:30              16:40



The total flight time (excluding halts) is: 5 hours 35 minutes
```



<terminated> Flights [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (Apr 29, 2017, 12:18:19 PM)

```
FLIGHT AGENDA

Enter the departure airport code: COK
Enter the departure time (HH:MM): 11:30
Enter the destination airport code: GOI

Flights departing from  COK airport at or after 11:30 to GOI are:

The route from COK to GOI is:

GOI <-- BLR <--  COK

The Flight Details on your route are:

From Airport COK

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Royal Airways 223           BLR         12:0               13:10
Royal Airways 213           BLR         17:0               18:10
Alliance Air 197            BLR         22:0               23:10

From Airport BLR

FLIGHT                  DESTINATION     DEPARTURE       ARRIVAL
_______________________________________________________________________
Alliance Air 102            GOI         16:0               17:0
Alliance Air 202            GOI         17:0               18:15
Alliance Air 333            GOI         18:0               19:10



The total flight time (excluding halts) is: 2 hours 25 minutes
```

GRAPH:


DEL
H
I
CCU
Bom
E
G
HYD
D
F
GOI
B
BLR
MAS
C
A
COK