```
//William Kelley
//Assignment 2 - Paranoid Airlines
//ParanoidAir.cs

using System;
using System.Collections.Generic;

namespace ParanoidAirlines
{
    class ParanoidAir
    {
        Queue<int> firstClassLine = new Queue<int>(); //first class line
        Queue<int> businessClassLine = new Queue<int>(); //business class line
        Queue<int> economyClassLine = new Queue<int>(); //economy class line
        int[] firstClassAgent = new int[1]; //first class agent
        int[] businessClassAgent = new int[1]; //business class agent
        int[] economyClassAgent = new int[1]; //economy class agent
        int clock = 0; //simulated clock
        int duration = 720; //duration of simulation
        int economyClassFrequency = 3; //one economy class customer in every 3 minutes
        int businessClassFrequency = 15; //one business class customer in every 15 minutes
        int firstClassFrequency = 30; //one first class customer in every 30 minutes

        static void Main(string[] args)
        {
            ParanoidAir passenger = new ParanoidAir();
            int firstClassCustomer = 0; //first class customer. 0 means no customer
            int businessClassCustomer = 0; //business class customer. 0 means no customer
            int economyClassCustomer = 0; //economy class customer. 0 means no customer

            passenger.firstClassAgent[0] = -1; //-1 means no customer
            passenger.businessClassAgent[0] = -1; //-1 means no customer
            passenger.economyClassAgent[0] = -1; //-1 means no customer

            int economyClassHelp = 0; //economy class completion time
            int businessClassHelp = 0; //business class completion time
            int firstClassHelp = 0; //first class completion time

            Random random = new Random(); //random number generator

            int economyClassTimer = 0; //economy class timer
            int businessClassTimer = 0; //business class timer
            int firstClassTimer = 0; //first class timer

            int sumEconomyClassTime = 0; //sum time taken for economy class customers
            int sumBusinessClassTime = 0; //sum time taken for business class customers
            int sumFirstClassTime = 0; //sum time taken for first class customers

            float averageEconomyClassTime = 0; //average time taken for economy class customer
            float averageBusinessClassTime = 0;//average time taken for business class customer
            float averageFirstClassTime = 0; //average time taken for first class customer

            for (int i = 0; i < passenger.duration; i++)
            {
                passenger.clock = i;
                //Console.WriteLine("Clock: " + passenger.clock);
                //Check if a new customer of any class has arrived.
                if (passenger.clock % passenger.economyClassFrequency == 0)
                {
                    economyClassCustomer++; //new economy class customer
                    passenger.economyClassLine.Enqueue(economyClassCustomer); //add economy class
customer to line
                    //Console.WriteLine("Economy: " + economyClassCustomer);
                }
                if (passenger.clock % passenger.businessClassFrequency == 0)
                {
                    businessClassCustomer++; //new business class customer
                    passenger.businessClassLine.Enqueue(businessClassCustomer); //add business
class customer to line
                    //Console.WriteLine("Business: " + businessClassCustomer);
                }
                if (passenger.clock % passenger.firstClassFrequency == 0)
```

```
                {
                    firstClassCustomer++; //new first class customer
                    passenger.firstClassLine.Enqueue(economyClassCustomer); //add first class
customer to line
                    //Console.WriteLine("First: " + firstClassCustomer);
                }

                //check if a customer is served
                //for economy class
                if (passenger.economyClassAgent[0] == economyClassTimer)
                    passenger.economyClassAgent[0] = -1;
                //for business class
                if (passenger.businessClassAgent[0] == businessClassTimer)
                    passenger.businessClassAgent[0] = -1;
                //for first class
                if (passenger.firstClassAgent[0] == firstClassTimer)
                    passenger.firstClassAgent[0] = -1;
                //Check if any of agent empty
                //If yes, then assign a customer
                if (passenger.economyClassAgent[0] == -1) //economy class agent empty
                {
                    if (passenger.economyClassLine.Count != 0) //economy class line not empty
                    {
                        passenger.economyClassLine.Dequeue(); //remove a customer from economy
class line and send to agent
                                                              //calculate expected time to serve
                        economyClassHelp = random.Next(5, 10);
                        passenger.economyClassAgent[0] = economyClassHelp;
                        economyClassTimer = 0; //reset economy class timer
                        sumEconomyClassTime += economyClassHelp;
                    }
                }
                else if (passenger.businessClassAgent[0] == -1) //business class agent empty
                {
                    if (passenger.businessClassLine.Count != 0) //business class line not empty
                    {
                        passenger.businessClassLine.Dequeue(); //remove a customer from business
class line and send to agent
                                                               //calculate expected time to serve
                        businessClassHelp = random.Next(6, 12);
                        passenger.businessClassAgent[0] = businessClassHelp;
                        businessClassTimer = 0; //reset business class timer
                        sumBusinessClassTime += businessClassHelp;
                    }
                    else if (passenger.economyClassLine.Count != 0) //in case if business class
and first class line both are empty, but economy class line not empty
                    {
                        passenger.economyClassLine.Dequeue(); //remove a customer from economy
class line and send to agent
                                                              //calculate expected time to serve
                        economyClassHelp = random.Next(5, 10);
                        passenger.economyClassAgent[0] = economyClassHelp;
                        economyClassTimer = 0; //reset economy class timer
                        sumEconomyClassTime += economyClassHelp;
                    }
                }
                else if (passenger.firstClassAgent[0] == -1) //first class agent empty
                {
                    if (passenger.firstClassLine.Count != 0) //first class line not empty
                    {
                        passenger.firstClassLine.Dequeue(); //remove a customer from first class
line and send to agent
                                                            //calculate expected time to serve
                        firstClassHelp = random.Next(5, 20);
                        passenger.firstClassAgent[0] = firstClassHelp;
                        firstClassTimer = 0; //reset first class timer
                        sumFirstClassTime += firstClassHelp;
                    }
                    else if (passenger.businessClassLine.Count != 0) //in case first class line
empty, but business class line not empty
                    {
                        passenger.businessClassLine.Dequeue(); //remove a customer from business
```

```
class line and send to agent
                                                        //calculate expected time to serve
                        businessClassHelp = random.Next(6, 12);
                        passenger.businessClassAgent[0] = businessClassHelp;
                        businessClassTimer = 0; //reset business class timer
                        sumBusinessClassTime += businessClassHelp;
                    }
                    else if (passenger.economyClassLine.Count != 0) //in case first class line
and business class line both are empty, but economy class line not empty
                    {
                        passenger.economyClassLine.Dequeue(); //remove a customer from economy
class line and send to agent
                                                       //calculate expected time to serve
                        economyClassHelp = random.Next(5, 10);
                        passenger.economyClassAgent[0] = economyClassHelp;
                        economyClassTimer = 0; //reset economy class timer
                        sumEconomyClassTime += economyClassHelp;
                    }
                }
                //increment timers
                economyClassTimer++;
                businessClassTimer++;
                firstClassTimer++;
            }
            averageEconomyClassTime = (float)sumEconomyClassTime / economyClassCustomer;
            averageBusinessClassTime = (float)sumBusinessClassTime / businessClassCustomer;
            averageFirstClassTime = (float)sumFirstClassTime / firstClassCustomer;
            Console.WriteLine("Total customers for First Class Line: {0}", firstClassCustomer);
            Console.WriteLine("Average time to process First Class customer: {0}",
averageFirstClassTime);
            Console.WriteLine("\nTotal customer for Business Class Line: " +
businessClassCustomer);
            Console.WriteLine("Average time to process Business Class customer: {0}",
averageBusinessClassTime);
            Console.WriteLine("\nTotal customers for Economy Class Line: " +
economyClassCustomer);
            Console.WriteLine("Average time to process Economy Class customer: {0}",
averageEconomyClassTime);
        }
    }
}
```

```
Total customers for First Class Line: 24
Average time to process First Class customer: 12.91667

Total customer for Business Class Line: 48
Average time to process Business Class customer: 8.895833

Total customers for Economy Class Line: 240
Average time to process Economy Class customer: 7.029167

Press any key to continue...
```