Assignment 4

Name: Ramanpreet Kaur

Code:

* Seat.cs

public class Seat

{

public char Label { get; }

public bool Booked { get; private set; }

public Passenger Passenger { get; private set; }

public Seat(char label)

{

Label = label;

Booked = false;

Passenger = null;

}

public bool IsAvailable()

{

return !Booked;

}

public void BookSeat(Passenger passenger)

{

Passenger = passenger;

Booked = true;

}

}

* Passenger.cs

public enum SeatPreference

{

Window = 1,

Aisle

}

public class Passenger

{

public string FirstName { get; }

public string LastName { get; }

public SeatPreference Preference { get; }

public Seat Seat { get; set; }

public Passenger(string firstName, string lastName, SeatPreference preference)

{

FirstName = firstName;

LastName = lastName;

Preference = preference;

Seat = null;

}

}

* Row.cs

using System.Collections.Generic;

public class Row

{

public int RowNumber { get; }

public List<Seat> Seats { get; }

public Row(int rowNumber)

{

RowNumber = rowNumber;

Seats = new List<Seat>();

for (char label = 'A'; label <= 'D'; label++)

{

Seats.Add(new Seat(label));

}

}

public string DisplayRow()

{

string rowDisplay = "";

foreach (var seat in Seats)

{

if (seat.Booked)

{

rowDisplay += $"{seat.Passenger.FirstName[0]}{seat.Passenger.LastName[0]} ";

}

else

{

rowDisplay += $"{seat.Label} ";

}

}

return rowDisplay.Trim();

}

}

* Plane.cs

using System.Collections.Generic;

public class Plane

{

public List<Row> Rows { get; }

public Plane()

{

Rows = new List<Row>();

for (int i = 1; i <= 12; i++)

{

Rows.Add(new Row(i));

}

}

public void DisplaySeatingChart()

{

foreach (var row in Rows)

{

Console.WriteLine(row.DisplayRow());

}

}

}

* Program.cs

using System;

namespace TicketingApp

{

class Program

{

static void Main(string[] args)

{

Plane plane = new Plane();

while (true)

{

Console.WriteLine("\nPlease enter 1 to book a ticket.");

Console.WriteLine("Please enter 2 to see seating chart.");

Console.WriteLine("Please enter 3 to exit the application.");

string choice = Console.ReadLine().Trim();

switch (choice)

{

case "1":

BookTicket(plane);

break;

case "2":

plane.DisplaySeatingChart();

break;

case "3":

Console.WriteLine("Exiting the application. Goodbye!");

return;

default:

Console.WriteLine("Invalid choice. Please enter a valid option.");

break;

}

}

}

static void BookTicket(Plane plane)

{

Console.WriteLine("Please enter the passenger's first name:");

string firstName = Console.ReadLine().Trim();

Console.WriteLine("Please enter the passenger's last name:");

string lastName = Console.ReadLine().Trim();

Console.WriteLine("Please enter 1 for a Window seat preference, 2 for an Aisle seat preference, or hit enter to pick the first available seat:");

string preferenceInput = Console.ReadLine().Trim();

SeatPreference preference;

if (Enum.TryParse(preferenceInput, out preference))

{

if (preference != SeatPreference.Window && preference != SeatPreference.Aisle)

{

Console.WriteLine("Invalid preference. Defaulting to first available seat.");

preference = 0;

}

}

else

{

preference = 0;

}

foreach (var row in plane.Rows)

{

foreach (var seat in row.Seats)

{

if (preference == 0 || (preference == SeatPreference.Window && (seat.Label == 'A' || seat.Label == 'D')) || (preference == SeatPreference.Aisle && (seat.Label == 'B' || seat.Label == 'C')))

{

if (seat.IsAvailable())

{

Passenger passenger = new Passenger(firstName, lastName, preference);

seat.BookSeat(passenger);

passenger.Seat = seat;

Console.WriteLine($"The seat located in {row.RowNumber} {seat.Label} has been booked.");

return;

}

}

}

}

Console.WriteLine("Sorry, all seats are booked.");

}

}

}

* Output

