PPA Assignment 5

Wonjoon Seol, Computer Science with Intelligent Systems, K1631098

November 13, 2016

1 Introduction

In this assignment we create a flight simulation program. The plane flies from Beijing to Istanbul, Istanbul to Dhaka, Dhaka to Istanbul and back to Beijing each day in the course over 120 days. The plane also goes to repair at a given distance for 7 days. I need to demonstrate my ability to use conditional statements, loops, constructors and be able to interact with multiple classes (Topics from week 1,. . . ,5).

2 Pseudocode

2.1 Class Coordinates

Pseudocode 1: This class models Coordinate system.

```
Initialise private integer $x$
Initialise private integer $y$
Define Coordinates
    Set $x$
    Set $y$
Define getX
    Return $x$
Define getY
    Return $y$
Define setX
    Set $x$
Define sety
    Set $y$
```

2.2 Class Destination

Pseudocode 2: This class models destination.

```
Initialise private String name
Initialise private Coordinates coordinates
Define Destination
    Set name
    Set coordinates
Define getName
    Return name
Define getCoordinates
    Return coordinates
```

2.3 Class Aeroplane

Pseudocode 3: This class has a characteristics of a plane. (Continued on next page)

```
Define getName
    Return name
Define getCoordinates
    Return coordinates
Define getSpeed
    Return speed
Define getTotalDistance
    Return distance
Define setTotalDistance
    Set distance
Define getRepairDistance
    Return distance
```

2.4 Class Aeroplane (Continued)

Pseudocode 4: This class has a characteristics of a plane.

```
Initialise private String name
Initialise private Coordinates coordinates
Initialise private int speed
Initialise private int totalDistance
Initialise private int repairDistance
Define Aeroplane
    Set name
    Set coordinates
    Set speed
    Set totalDistance
    Set repairDistance
Define singleFlight
Initialise int distance = 0
while Coordinates of current location and destination are not equal do
    if Current x coordinate is greater than destination AND difference is less than or equal
    to speed then
        Add distance positive value of its difference
        Set Current Coordinate of X equal to its destination
    else if Current x coordinate is less than destination AND difference is less than or equal
    to speed then
        Add distance positive value of its difference
        Set Current Coordinate of X equal to its destination
    else if Current x coordinate is greater than destination then
        Add speed to distance speed
        Subtract amount of speed from Current Coordinate of X
    else
        Add speed to distance speed
        Add amount of speed from Current Coordinate of X
    end
    if Current y coordinate is greater than destination AND difference is less than or equal
    to speed then
        Add distance positive value of its difference
        Set Current Coordinate of y equal to its destination
    else if Current y coordinate is less than destination AND difference is less than or equal
    to speed then
        Add distance positive value of its difference
        Set Current Coordinate of y equal to its destination
    else if Current y coordinate is greater than destination then
        Add speed to distance speed
        Subtract amount of speed from Current Coordinate of y
    else
        Add speed to distance speed
        Add amount of speed from Current Coordinate of y
    end
end
Add distance to totalDistance
Return Distance
```

2.5 FlightSimulation

Pseudocode 5: This class is going to drive our program

```
Initialise Coordinates $beijing$ with $100, 45$
Initialise Destination $destination1$ with $"Beijing", beijing$
Initialise Coordinates $beijing$ with $145, 120$
Initialise Destination $destination1$ with $"Istanbul", istanbul$
Initialise Coordinates $beijing$ with $30, 90$
Initialise Destination $destination1$ with $"Dhaka", dhaka$
Initialise Coordinates $airbusCoordinates$ with $100, 45$
Initialise Aeroplane $airbus$ with $"Airbus", airbusCoordinates, 9, 0, 1600$
Print Airbus destination1 to destination2
Airbus flies to $destination2$
Print $distance$ and $totalDistance$
Print Airbus destination1 to destination3
Airbus flies to $destination3$
Print $distance$ and $totalDistance$
Print Airbus destination1 to destination2
Airbus flies to $destination2$
Print $distance$ and $totalDistance$
Print Airbus destination1 to destination1
Airbus flies to $destination1$
Print $distance$ and $totalDistance$
Initialise repairNum
for day 1 to day 120 do
    Print Current day number
    Print $totalDistance$
    if $totalDistance$ travelled is greater than $repairDistance$ then
        Add 6 to $day$
        Set $totalDistance$ 0
        Add 1 to $repairNum$
    else
        Airbus flies to $destination2$
        Airbus flies to $destination3$
        Airbus flies to $destination2$
        Airbus flies to $destination1$
    end
end
Print $repairNum$
```

3 Class Diagram

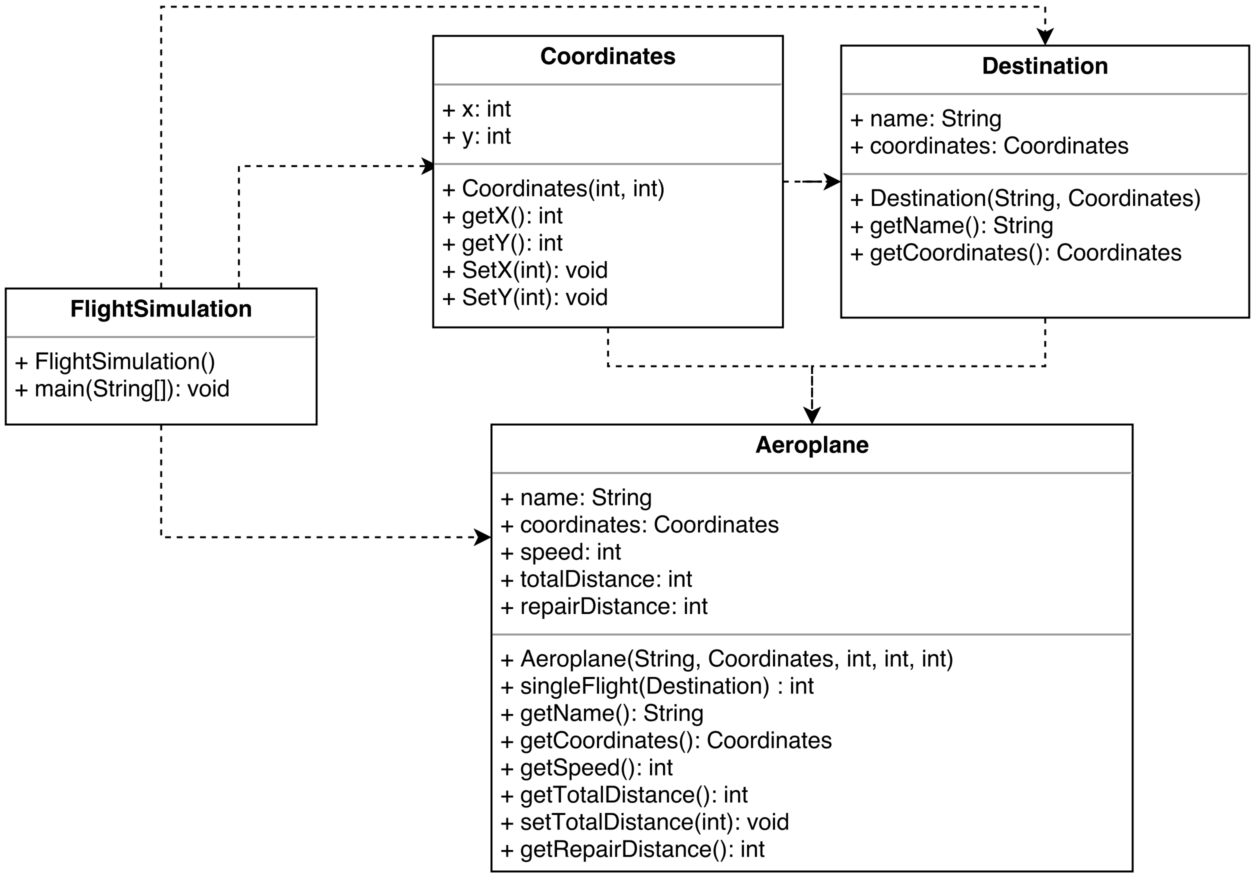

Figure 1: FlightSimulation Class Diagram.

4 Description

1. Class Coordinates and Destination
   Class coordinates have getters and setters for private integer fields x and y coordinates, and Class Destination holds this coordinates as an object with a string value representing its name.

2. Class Aeroplane
   The method SingleFlight would be the most difficult part in this assignment, the while loop checks whether current coordinates is not equal to the coordinates of the destination. Then depending on whether the current coordinates has larger value than the destination coordinates it will either subtract/add the value of *speed* to the current coordinates and increase *distance* travelled. If its difference is less than the speed the plane will only move its remaining distance. The distance travelled should be updated before changing the coordinates.

3. Class FlightSimulation
   This is our driver class, I tried to create a separate method called *journey* in the main method as the question number 5 has similar patterns for its sub questions. However, after discussing with Dr. Martin Chapman, this method was removed because the automatic marker may penalise printing starting destination name prior to the aeroplane name. I had a problem in this assignment. When I initialised aeroplane *Airbus* I saw its original location equals to Beijing and supplied Beijing instead. My belief was that When an object is passed as a

parameter it would be copied just like any other variable. Thus, the plane never returned back to Beijing as the coordinate of Beijing equaled to its departing location all the time.

Finally the solution to question number 6 involves loop. I used for-loop, so that it would iterate for 120 days. The plane undergoes repair for 7 days but as the loop checks for repair at the start of the day, the day should only be incremented by 6 instead. The necessary number of repairs can be verified mathematically: Each day the plane travels $120 \times 2 + 145 \times 2 = 530$. The $repairDistance = 1600$ so every 4 days the plane undergoes repair: $\frac{120}{4+7} \approx 10.91$. This means that the plane is still being repaired at day 120, making the total repair number 11. This equals to the console output.