Nolan Anderson

CS 307

Jacob Hauenstein

14 February 2021

Programming Assignment 1: Preliminary class Diagram

Section 1 – Class Overview

In general, my main idea for this project is to use the Flight Number to do my calculations and receive data from the other classes. Essentially, Flight Sim is the main class that carries out the program. Flight sim will call helper functions. Those helper functions will create new objects (which then parse the xml files) and then use that object to return the data the parser found.

Section 2 – Class Outline

|  |
| --- |
| FlightSim::FlightSim(); // Constructor.  ~FlightSim::FlightSim();// Destructor.  int SetMultiplier(); // Set the time multiplier.  string SetInFile(); // Set the text file to read.  void Start(int Multiplier, string InFile) // Start the simulation.  // Calculate current location of the airline.  int CurrentLocation(int FlightNum, int CurrentHr, int CurrentMin);  int FlightTime(int FlightNum); // Calculate flight time  void OutNewFlight(int FlightNum); // On a new flight, output data.  void OutInterval(int FlightNum); // Output the data on 5 second intervals. |

Figure 1: Outline for FlightSim class.

|  |
| --- |
| FlightData::FlightData(string InFile); // Constructor calls SetData().  ~FlightData::FlightData();  void SetData(string InFile); // Calls parser to set Classes' data.  string ReturnAirline(int FlightNum); // Returns the airline.  string ReturnType(int FlightNum); // Returns the plane type.  string ReturnDepCity(int FlightNum); // Returns departure city.  string ReturnDestCity(int FlightNum); // Returns destination city.  int ReturnTime(int FlightNum); // Returns departure time.  int ReturnFlightNum(); // Returns the flight num. |

Figure 2: Outline for FlightData class.

|  |
| --- |
| CityData::CityData(string InFile); // Constructor calls SetData().  ~CityData::CityData();  void SetData(string InFile); // Calls parser to set Classes' data.  string ReturnName(); // Returns City name.  string ReturnState(); // Returns State of City.  int ReturnLatitude(string CityName); // Returns Lat of City.  int ReturnLongitude(string CityName); // Returns Lon of City.  int ReturnDistance(int FlightNum); // Returns distance to other city. |

Figure 3: Outline for CityData class.

|  |
| --- |
| // This data will help to calculate flight time.  AircraftData::AircraftData(string InFile); // Constructor calls SetData().  ~AircraftData::AircraftData();  void SetData(string InFile); // Calls Parser to set Classes' data.  // Returns data for class FlightSim. If we have flight num, we can find all of the data we need  // to return our information.  string ReturnMake(int FlightNum); // Returns plane make.  string ReturnModel(int FlightNum); // Returns plane model.  int ReturnSpeed(string model); // Returns plane speed.  int ReturnClimb(string model); // Returns climb speed.  int ReturnWing(string model); // Returns wing span.  int ReturnFuselage(string model); // Returns Fuselage length. |

Figure 4: Outline for AircraftData class.

|  |
| --- |
|  |

Figure 5: UML Class Outline