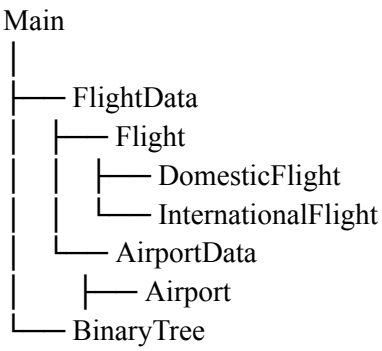
Flight Management System Documentation:

1. Overview

This program allows users to search for domestic or international flights based on various criteria such as origin, destination, date, and available airports. The user can interact with the program through a simple console interface, making it easy to search for flights from a list of flight data. The program uses a **binary tree** to store and search for flights efficiently.

2. Class Hierarchy

Class hierarchy used in the program:



1. **Main**
    - This is the entry point of the program that contains the main method. It displays the flight search menu on the terminal, processes user input, and interacts with other classes to perform flight search tasks.
2. **FlightData**
    - Responsible for loading and creating flight objects from CSV files based on the selected flight type (domestic or international).
3. **Flight**
    - The parent class that represents a generic flight. It contains attributes common to both domestic and international flights, such as flight number, origin, destination, airline, etc.
4. **DomesticFlight** and **InternationalFlight**
    - These are subclasses of Flight. They represent specific types of flights (domestic or international) and override methods like isBaggageAllowed() to provide specific behavior.

5. **AirportData**
    - Provides lists of available domestic and international airports.
6. **Airport**
    - Represents an airport, with the capability to get a list of destinations served from the origin airport.
7. **BinaryTree**
    - A generic class used to create a binary tree structure for storing and searching flights efficiently based on date and other criteria.

3. Class Documentation

1. **Main**
    - **Purpose**: Provides the main user interface for the program. It prompts the user for flight type and desired actions (like checking available airports or searching for flights by criteria).
    - **Key Methods**:
        - main(String[] args): The entry point of the program that initializes the program, takes input from the user, and manages the flow of execution.
    - **Workflow**:
        - The user is prompted to choose between domestic or international flights.
        - Based on the user's choice, the corresponding CSV file is loaded, and the flight list is created.
        - The program offers a menu of actions (check airports, search flights by date, etc.), and the user can interact with the program until they choose to exit.
2. **FlightData**
    - **Purpose**: Responsible for creating and managing flight data from CSV files.
    - **Key Methods**:
        - createFlight(String fileName, boolean isDomestic): Reads the corresponding CSV file and creates a list of flights.
    - **Workflow**:
        - The class loads a CSV file (either for domestic or international flights) and returns a list of flight objects. The class uses file I/O operations and parses the data into Flight objects.
3. **Flight (abstract)**
    - **Purpose**: Represents a generic flight. This class is extended by specific flight types like DomesticFlight and InternationalFlight.
    - **Attributes**:
        - flightNumber: Flight's unique identifier.

        - date: Date of the flight.
        - origin: Starting airport.
        - originState: State of the origin.
        - destination: Destination airport.
        - destinationState: State of the destination.
        - airline: Airline name.
        - departureTime: Time of departure.
        - arrivalTime: Time of arrival.
    - **Methods**:
        - getDetails(): Returns a string with detailed information about the flight.
        - isBaggageAllowed(): Abstract method to be implemented by subclasses.
4. **DomesticFlight and InternationalFlight**
    - **Purpose**: These are subclasses of Flight, each representing a type of flight. DomesticFlight represents flights within the same country, and InternationalFlight represents flights that cross country borders.
    - **Methods**:
        - isBaggageAllowed(): Returns whether baggage is allowed for the flight type.
        - getDetails(): Returns specific flight details depending on whether it's a domestic or international flight.
5. **AirportData**
    - **Purpose**: Contains predefined lists of domestic and international airports.
    - **Attributes**:
        - DOMESTIC_AIRPORTS: Array of domestic airports.
        - INTERNATIONAL_AIRPORTS: Array of international airports.
    - **Methods**:
        - N/A - This class simply contains static data about airports.
6. **Airport**
    - **Purpose**: Represents an airport, including the origin and destination airports and a list of possible destinations.
    - **Attributes**:
        - originAirport: Name of the origin airport.
        - destinationList: List of destinations that can be reached from the origin airport.
    - **Methods**:
        - getOriginAirport(): Returns the origin airport.
        - getDestinationList(): Returns the list of destination airports from the origin.
7. **BinaryTree**

- **Purpose**: A generic class used to store flight objects in a binary tree, allowing for efficient searching based on specific attributes (e.g., flight date).
- **Attributes**:
  - root: The root node of the binary tree.
- **Methods**:
  - insert(T value): Inserts a flight object into the binary tree.
  - search(Node root, String date): Searches for flights on a given date.

4. How to Run the Program

- **Compile the Program**:

To compile the Java classes, use the following command in your terminal:

javac -d src src/*.java

- **Run the Program**:

To run the program after compiling, use the following command:

java -cp src Main

- This command runs the Main class, which will start the flight search application.

5. Example Run

Welcome to Flight Search!

Do you want domestic or international flights? (Enter 'd' or 'i'): d

---
Flight Search Menu:
---
1. Check available domestic airports
2. Check where you can fly from your origin airport
3. Check all available flights on a certain date
4. Search for flights based on origin, destination, and date
Enter your choice (1-4): 1

Domestic Airports:

ATL | LAX | ORD | DFW | DEN | JFK | SFO | LAS | SEA | MIA | EWR | CLT | IAH | PHX | MSP | DTW | BOS | LGA | MDW | PDX | HOU |

Enter 6 to continue or 5 to exit.