Exercise: Observer - Typescript

Assumptions

Node.js and ts-node are installed on your computer, as described in the “Typescript Setup Instructions”.

Description

The provided zip file, Observer-Typescript.zip, contains the Typescript code for a program named FlightMonitor that selects a random airline flight that is currently in the air, and monitors its progress until it lands. The program’s main function is in the FlightMonitor.ts file. Most of the program’s functionality is in the FlightFeed class. When you call the “start” method on a FlightFeed object, it downloads information about all currently in-air flights, and selects one of them to monitor. The selected flight’s status is stored in the FlightFeed object’s “flight” property. After initializing its “flight” property, every 60 seconds the FlightFeed object downloads the current status of the flight being monitored, and if the flight’s status has changed, it updates the value of its “flight” property to reflect the new status.

The program can be run as follows:

```
> ts-node src/FlightMonitor.ts
```

After reading the provided Typescript code, modify it to use the Observer pattern as follows:

1. Create and implement a Subject base class and an Observer interface, as described in the Observer pattern.
2. Modify the FlightFeed class to be “observable” such that it notifies its observers each time its “flight” property changes.
3. Write an observer class that displays the status of the monitored flight every time its status changes, including the flight’s transponder id (icao24), call sign, country of origin, longitude, latitude, velocity, and altitude.
4. Write a second observer class that displays the deltas (or changes) in longitude, latitude, velocity, and altitude between status updates.
5. Integrate your observers into the FlightMonitor program so the output of the observers is displayed each time the flight status changes.
6. Submit your Typescript code on Canvas.