**Problem Statement: Air Traffic Monitoring System**

Airtraffice in modern airlines is monitored using advanced tracking systems. Each aircraft is equipped with a tracking device that sends information about its altitude and distance traveled at specific intervals. A warning is issued if the average altitude during this interval exceeds a certain value, H meters.

Complete a class **AirTrafficMonitoringService** which implements an interface **IAirTrafficMonitoringService**:

- **void registerAircraft(int aircraftId)**: Create an object of class **Aircraft** using **aircraftId**. The definition of the **Aircraft** class is given in the code stub. This class stores the values **aircraftId**, **lastPolledInfo**, and **numberOfWarningsIssued**.
- **Aircraft getAircraftInfo(int aircraftId)**: Returns the object that represents the aircraft with the **aircraftId**.
- **boolean polledAircraftInfo(int aircraftId, long distanceTraveledInMeters, long altitudeInMeters, long epochTime)**: Calculate and check if the average altitude is greater than H units. If it is, return true; otherwise, return false.

Average altitude = altitudeInMeters / (epochTime - lastPolledEpochTime)

- **List<Long> warningHistory(int aircraftId, int K)**: Returns a list of the last K timestamps when a warning was issued for this aircraft. The list should be in descending order.

```
import java.util.ArrayList;

import java.util.List;


public class Aircraft {

   private int aircraftId;

   private long lastPolledInfo; // Last time this aircraft was polled

   private int numberOfWarningsIssued;

   private List<Long> warningTimestamps; // Store timestamps of warnings


   public Aircraft(int aircraftId) {

      this.aircraftId = aircraftId;

      this.lastPolledInfo = 0; // Initialize as needed

      this.numberOfWarningsIssued = 0;

      this.warningTimestamps = new ArrayList<>(); // Initialize the list for warning timestamps
```

```
    }

    public long getLastPolledInfo() {
        return lastPolledInfo;
    }

    public void updateLastPolledInfo(long altitude, long time) {
        this.lastPolledInfo = time; // Update with the current time
        // Store altitude if needed
    }

    public void incrementWarningCount(long timestamp) {
        this.numberOfWarningsIssued++;
        this.warningTimestamps.add(timestamp); // Store the timestamp of the warning
    }

    public List<Long> getWarningTimestamps() {
        return warningTimestamps;
    }
}



public interface IAirTrafficMonitoringService {
    void registerAircraft(int aircraftId);
    Aircraft getAircraftInfo(int aircraftId);
    boolean polledAircraftInfo(int aircraftId, long distanceTraveledInMeters, long altitudeInMeters,
long epochTime);
    List<Long> warningHistory(int aircraftId, int K);
}
```

```java
import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.stream.Collectors;


public class AirTrafficMonitoringService implements IAirTrafficMonitoringService {

    private Map<Integer, Aircraft> aircrafts = new HashMap<>();

    private static final long H = 10000; // Set the altitude limit (H) in meters


    @Override

    public void registerAircraft(int aircraftId) {

        // Empty method

    }


    @Override

    public Aircraft getAircraftInfo(int aircraftId) {

        // Empty method

        return null;

    }


    @Override

    public boolean polledAircraftInfo(int aircraftId, long distanceTraveledInMeters, long
altitudeInMeters, long epochTime) {

        // Empty method

        return false;

    }


    @Override

    public List<Long> warningHistory(int aircraftId, int K) {

        // Empty method
        return null;
    }
}




import java.util.List;

public class Main {
    public static void main(String[] args) {
        // Create an instance of the AirTrafficMonitoringService
        AirTrafficMonitoringService service = new AirTrafficMonitoringService();

        // Register some aircraft
        service.registerAircraft(1);
        service.registerAircraft(2);
        service.registerAircraft(3);

        // Simulate polling aircraft info
        long currentTime = System.currentTimeMillis() / 1000; // Get current time in seconds

        // First Polling: Aircraft 1
        boolean warning1 = service.polledAircraftInfo(1, 10000, 12000, currentTime);
        System.out.println("Aircraft 1 Warning Issued: " + warning1); // Expect false

        // Update last polled info for aircraft 1
        service.getAircraftInfo(1).updateLastPolledInfo(12000, currentTime);

        // Second Polling: Aircraft 1 with higher altitude
        currentTime += 3600; // Simulate 1 hour later
        boolean warning2 = service.polledAircraftInfo(1, 10000, 15000, currentTime);
```

```
System.out.println("Aircraft 1 Warning Issued: " + warning2); // Expect true

// Check warning history for aircraft 1
List<Long> warningHistory1 = service.warningHistory(1, 5);
System.out.println("Aircraft 1 Warning History: " + warningHistory1);

// First Polling: Aircraft 2
boolean warning3 = service.polledAircraftInfo(2, 10000, 9000, currentTime);
System.out.println("Aircraft 2 Warning Issued: " + warning3); // Expect false

// Update last polled info for aircraft 2
service.getAircraftInfo(2).updateLastPolledInfo(9000, currentTime);

// Second Polling: Aircraft 2 with higher altitude
currentTime += 3600; // Simulate 1 hour later
boolean warning4 = service.polledAircraftInfo(2, 10000, 11000, currentTime);
System.out.println("Aircraft 2 Warning Issued: " + warning4); // Expect true

// Check warning history for aircraft 2
List<Long> warningHistory2 = service.warningHistory(2, 5);
System.out.println("Aircraft 2 Warning History: " + warningHistory2);

// First Polling: Aircraft 3
boolean warning5 = service.polledAircraftInfo(3, 10000, 8000, currentTime);
System.out.println("Aircraft 3 Warning Issued: " + warning5); // Expect false

// Update last polled info for aircraft 3
service.getAircraftInfo(3).updateLastPolledInfo(8000, currentTime);

// Second Polling: Aircraft 3 with higher altitude
currentTime += 3600; // Simulate 1 hour later
```

```
        boolean warning6 = service.polledAircraftInfo(3, 10000, 12000, currentTime);
        System.out.println("Aircraft 3 Warning Issued: " + warning6); // Expect true


        // Check warning history for aircraft 3
        List<Long> warningHistory3 = service.warningHistory(3, 5);
        System.out.println("Aircraft 3 Warning History: " + warningHistory3);
    }
}
```