Write a Java program to create a class called "Airplane" with a flight number, destination, and departure time attributes, and methods to check flight status and delay.

Sample Solution:

Java Code:

```
// Airplane.java
// Import the LocalTime class from the java.time package
import java.time.LocalTime;

// Define the Airplane class
public class Airplane {
  // Private field to store the flight number
  private String flightNumber;
  // Private field to store the destination
  private String destination;
  // Private field to store the scheduled departure time
  private LocalTime scheduledDeparture;
  // Private field to store the delay time in minutes
  private int delayTime;

  // Constructor to initialize the flight number, destination, and schedule
  public Airplane(String flightNumber, String destination, LocalTime schedu
    // Assign the flight number parameter to the flightNumber field
    this.flightNumber = flightNumber;
    // Assign the destination parameter to the destination field
    this.destination = destination;
    // Assign the scheduled departure time parameter to the scheduledDepart
    this.scheduledDeparture = scheduledDeparture;
    // Initialize the delay time to 0
    this.delayTime = 0;
  }

  // Getter method for the flight number
  public String getFlightNumber() {
    // Return the value of the flightNumber field
    return flightNumber;
  }

  // Setter method for the flight number
```

```
public void setFlightNumber(String flightNumber) {
  // Assign the flight number parameter to the flightNumber field
  this.flightNumber = flightNumber;
}

// Getter method for the destination
public String getDestination() {
  // Return the value of the destination field
  return destination;
}

// Setter method for the destination
public void setDestination(String destination) {
  // Assign the destination parameter to the destination field
  this.destination = destination;
}

// Getter method for the scheduled departure time
public LocalTime getScheduledDeparture() {
  // Return the value of the scheduledDeparture field
  return scheduledDeparture;
}

// Setter method for the scheduled departure time
public void setScheduledDeparture(LocalTime scheduledDeparture) {
  // Assign the scheduled departure time parameter to the scheduledDepart
  this.scheduledDeparture = scheduledDeparture;
}

// Getter method for the delay time
public int getDelayTime() {
  // Return the value of the delayTime field
  return delayTime;
}

// Method to set a delay and update the scheduled departure time
public void delay(int minutes) {
  // Assign the delay time parameter to the delayTime field
  this.delayTime = minutes;
  // Update the scheduled departure time by adding the delay time
```

```
        this.scheduledDeparture = this.scheduledDeparture.plusMinutes(minutes);
    }

    // Method to check the status of the flight
    public void checkStatus() {
        // Check if there is no delay
        if (delayTime == 0) {
            // Print a message indicating the flight is on time
            System.out.println("Flight " + flightNumber + " is on time.");
        } else {
            // Print a message indicating the flight is delayed
            System.out.println("Flight " + flightNumber + " is delayed by " + del
        }
    }
}
```

The above class represents an airplane with a flight number, destination, and scheduled departure time. It has getter and setter methods for these attributes. The class also two methods "delay()" and "checkStatus()" to delay the flight and check its status. The "delay()" method takes an integer value representing the number of minutes the flight will be delayed and updates the scheduled departure time accordingly. By using “checkStatus()” method, you can determine whether the flight has been delayed or is on time.

```
// Import the LocalTime class from the java.time package                Copy
import java.time.LocalTime;

// Define the Main class
public class Main {
    // Main method, entry point of the program
    public static void main(String[] args) {
        // Create a new Airplane object with flight number "CDE345", destinatio
        Airplane flight1 = new Airplane("CDE345", "London", LocalTime.of(10, 30
        // Create a new Airplane object with flight number "KUI765", destinatio
        Airplane flight2 = new Airplane("KUI765", "New York", LocalTime.of(14,
        // Create a new Airplane object with flight number "JUY456", destinatio
        Airplane flight3 = new Airplane("JUY456", "Paris", LocalTime.of(14, 0));
        // Print the initial flight status
        System.out.println("Flight Status:");
        // Check and print the status of flight1
        flight1.checkStatus();
```

```
        // Check and print the status of flight2
        flight2.checkStatus();
        // Check and print the status of flight3
        flight3.checkStatus();
        // Delay flight1 by 40 minutes
        flight1.delay(40);
        // Delay flight2 by 110 minutes
        flight2.delay(110);
        // Print the current flight status after delays
        System.out.println("\nCurrent Flight Status:");
        // Check and print the status of flight1
        flight1.checkStatus();
        // Check and print the status of flight2
        flight2.checkStatus();
        // Check and print the status of flight3
        flight3.checkStatus();
    }
}
```

In the main() function, we create three "Airplane" objects and set flight numbers, destinations and scheduled departure times. It then calls the "checkStatus()" method to display the initial flight status of each flight. The program then delays flight1 and flight2 by calling the "delay()" method on these objects, and then calls the "checkStatus()" method again to display the updated flight status of each flight.

Sample Output:

```
Flight Status:
Flight CDE345 is on time.
Flight KUI765 is on time.
Flight JUY456 is on time.

Current Flight Status:
Flight CDE345 is delayed by 40 minutes.
Flight KUI765 is delayed by 110 minutes.
Flight JUY456 is on time.
```

Flowchart: