Developer Interview programming test

The classes below model a data structure for holding Flights and Airspace's.

Problem 1

Enhance the Flight class to allow its user to determine where it is located.

Problem 2

Given an instance of a Flight, and an instance of an Airspace, enhance the code to allow users to determine if the flight is within the airspace.

Classes

```
// Copyright (C) 2017 Snowflake Software Ltd. All rights reserved.
package techtest;

import java.time.Instant;

/**
 * The class representing a Flight.
 */
public class Flight {

    /** The aerodrome the flight is arriving at. */
    private final String arrivalAerodrome;

    /** The date/time the flight is arriving. */
    private final Instant arrivalTime;

    /** The aerodrome the flight is departing from. */
    private final String departureAerodrome;

    /** The date/time the flight is departing. */
    private final Instant departureTime;

    /**
     * Construct a flight.
     *
     * @param arrivalAerodrome The aerodrome the flight is arriving at.
     * @param arrivalTime The date/time the flight is arriving.
     * @param departureAerodrome The aerodrome the flight is departing from.
     * @param departureTime The date/time the flight is departing.
     */
    public Flight(final String arrivalAerodrome, final Instant arrivalTime,
        final String departureAerodrome, final Instant departureTime) {
        this.arrivalAerodrome = arrivalAerodrome;
        this.arrivalTime = arrivalTime;
```

```
        this.departureAerodrome = departureAerodrome;
        this.departureTime = departureTime;
    }

    /**
     * Gets the aerodrome the flight is arriving at.
     *
     * @return the arrival aerodrome.
     */
    public String getArrivalAerodrome() {
        return arrivalAerodrome;
    }

    /**
     * Gets the date/time the flight is arriving.
     *
     * @return the arrival time.
     */
    public Instant getArrivalTime() {
        return arrivalTime;
    }

    /**
     * Gets the aerodrome the flight is departing from.
     *
     * @return the departure aerodrome.
     */
    public String getDepartureAerodrome() {
        return departureAerodrome;
    }

    /**
     * Gets the date/time the flight is departing.
     *
     * @return the departure time.
     */
    public Instant getDepartureTime() {
        return departureTime;
    }
}
```

```java
// Copyright (C) 2017 Snowflake Software Ltd. All rights reserved.
package techtest;

/**
 * The class representing an Airspace.
 */
public class Airspace {

    /** The bottom left coordinate of this airspace. */
    private final Coordinate bottomLeft;

    /** The top right coordinate of this airspace. */
    private final Coordinate topRight;

    /**
     * Construct an Airspace.
     *
     * @param bottomLeft the bottom left coordinate of this airspace.
     * @param topRight the top right coordinate of this airspace.
     */
    public Airspace(final Coordinate bottomLeft, final Coordinate topRight) {
        this.bottomLeft = bottomLeft;
        this.topRight = topRight;
    }

    /**
     * Gets the bottom left coordinate of this airspace.
     *
     * @return the bottom left.
     */
    public Coordinate getBottomLeft() {
        return bottomLeft;
    }

    /**
     * Gets the top right coordinate of this airspace.
     *
     * @return the top right.
     */
    public Coordinate getTopRight() {
        return topRight;
    }
}
```

```java
// Copyright (C) 2017 Snowflake Software Ltd. All rights reserved.
package techtest;

/**
 * The class representing a coordinate (x, y).
 */
public class Coordinate {

    /** The X component of this coordinate (can have values -180 to +180). */
    private final double x;

    /** The Y component of this coordinate (can have values -90 to +90). */
    private final double y;

    /**
     * Construct a coordinate.
     *
     * @param x the X component of this coordinate (can have values -180 to +180).
     * @param y the Y component of this coordinate (can have values -90 to +90).
     */
    public Coordinate(final double x, final double y) {
        this.x = x;
        this.y = y;
    }

    /**
     * Gets the X component of this coordinate (can have values -180 to +180).
     *
     * @return the x.
     */
    public double getX() {
        return x;
    }

    /**
     * Gets the Y component of this coordinate (can have values -90 to +90).
     *
     * @return the y.
     */
    public double getY() {
        return y;
    }
}
```