Technical Interview

Fullstack: focus on Java, Web and Docker

# Java

Please explain what the command below does.

java -Xmx512m -cp my-app.jar ch.solenix.myapp.Program Hello World

This command runs a Java program named Program with classpath “ch.solenix.myapp.Program” from the my-app.jar file with a maximum heap size of 512 megabytes, passing "Hello" and "World" as arguments to the program.

# Java Programming

## Location Service

##### Objective

Create a Spring Boot application in Java that uses the provided JSON files to combine spacecraft onboard events with latitude and longitude positions to display the accurate geographic positions of each event. **This exercise is supported by 3 .json files provided as annex to this document (see Annex 1,2 and 3).**

##### Requirements

* Use Java and Spring Boot to create the application.
* The provided JSON files should not be modified. They contain time-stamped latitude and longitude positions that change over time, and spacecraft onboard events with *occurrence\_time, event\_name, id, and severity.*
* The application should provide a REST interface offering its users with two features:
  + Deliver the latitude/longitude for a given event-id
  + Deliver the whole list of events and associated positions
* The latitude and longitude positions for events should be determined by matching the event's occurrence\_time with the closest available positions in the latitudes.json and longitudes.json files. The candidate is invited to find ways to reduce and indicate the potential error for each position.

### Docker

Dockerize the application you wrote in 2.1. You can create one or multiple Dockerfiles, and if justified, a docker-compose.yml. These can be delivered in the same directory as the source-code.

# Angular

Write the declaration to use the following component in a template.

import { Component, Input } from '@angular/core';

@Component({

selector: 'welcome',

template: `<h1>Welcome to {{name}}!</h1>`,

styles: [`h1 { font-family: Lato; }`]

})

export class WelcomeComponent {

@Input() name: string;

}

1. Import the WelcomeComponent into the module where you want to use it. This is typically done in the app.module.ts file
2. Inside   
   @NgModule({   
    imports: [BrowserModule],  
    declarations: [WelcomeComponent],   
    bootstrap: [AppComponent] // Assuming AppComponent is main component  
   })
3. Now, you can use the <welcome> selector in any template within your Angular application. For example, in your app.component.html: example   
   <welcome [name]="'Your Name'"></welcome>

Describe, in general terms, the following snippet.

this.tableService.getTable(this.componentId).pipe(

tap(table =>

this.table = table

),

filter(table => table > 5)

).subscribe(

tableData => {

console.log(tableData)

})

it fetches table data from a service, assigns it to a property, filters out values greater than 5, and logs the filtered data to the console.

Annex 1 – events.json

[

    {

        "occurrence\_time": "2023-10-01T10:15:00",

        "event\_name": "System Startup",

        "id": "E001",

        "severity": "Info"

    },

    {

        "occurrence\_time": "2023-10-01T10:35:00",

        "event\_name": "Navigation Calibration",

        "id": "E002",

        "severity": "Info"

    },

    {

        "occurrence\_time": "2023-10-01T10:50:00",

        "event\_name": "Warning Light On",

        "id": "E003",

        "severity": "Warning"

    },

    {

        "occurrence\_time": "2023-10-01T11:05:00",

        "event\_name": "Data Transmission Error",

        "id": "E004",

        "severity": "Error"

    },

    {

        "occurrence\_time": "2023-10-01T11:25:00",

        "event\_name": "System Restart",

        "id": "E005",

        "severity": "Info"

    },

    {

        "occurrence\_time": "2023-10-01T11:45:00",

        "event\_name": "Low Battery",

        "id": "E006",

        "severity": "Warning"

    },

    {

        "occurrence\_time": "2023-10-01T12:05:00",

        "event\_name": "Data Loss",

        "id": "E007",

        "severity": "Error"

    },

    {

        "occurrence\_time": "2023-10-01T12:20:00",

        "event\_name": "Critical System Failure",

        "id": "E008",

        "severity": "Error"

    },

    {

        "occurrence\_time": "2023-10-01T12:45:00",

        "event\_name": "System Reboot",

        "id": "E009",

        "severity": "Info"

    },

    {

        "occurrence\_time": "2023-10-07T13:00:00",

        "event\_name": "Sensor Malfunction",

        "id": "E010",

        "severity": "Error"

    }

]

Annex 2 – latitudes.json

[

    {"timestamp": "2023-10-01T10:00:00", "position": 34.0522},

    {"timestamp": "2023-10-01T10:10:00", "position": 35.1234},

    {"timestamp": "2023-10-01T10:20:00", "position": 36.2456},

    {"timestamp": "2023-10-01T10:30:00", "position": 37.3245},

    {"timestamp": "2023-10-01T10:40:00", "position": 38.4321},

    {"timestamp": "2023-10-01T10:50:00", "position": 39.5567},

    {"timestamp": "2023-10-01T11:00:00", "position": 40.6742},

    {"timestamp": "2023-10-01T11:10:00", "position": 41.7789},

    {"timestamp": "2023-10-01T11:20:00", "position": 42.8865},

    {"timestamp": "2023-10-01T11:30:00", "position": 43.9687},

    {"timestamp": "2023-10-01T11:40:00", "position": 45.0123},

    {"timestamp": "2023-10-01T11:50:00", "position": 46.1021},

    {"timestamp": "2023-10-01T12:00:00", "position": 47.2134},

    {"timestamp": "2023-10-01T12:10:00", "position": 48.3290},

    {"timestamp": "2023-10-01T12:20:00", "position": 49.4257},

    {"timestamp": "2023-10-01T12:30:00", "position": 50.5567},

    {"timestamp": "2023-10-01T12:40:00", "position": 51.6321},

    {"timestamp": "2023-10-01T12:50:00", "position": 52.7122},

    {"timestamp": "2023-10-01T13:00:00", "position": 53.8112},

    {"timestamp": "2023-10-01T13:10:00", "position": 54.9223},

    {"timestamp": "2023-10-01T13:15:00", "position": 55.9891},

    {"timestamp": "2023-10-01T13:25:00", "position": 56.9877},

    {"timestamp": "2023-10-01T13:35:00", "position": 57.9855},

    {"timestamp": "2023-10-01T13:45:00", "position": 59.0765},

    {"timestamp": "2023-10-01T13:55:00", "position": 60.0998},

    {"timestamp": "2023-10-01T14:05:00", "position": 61.1267},

    {"timestamp": "2023-10-01T14:15:00", "position": 62.2254},

    {"timestamp": "2023-10-01T14:25:00", "position": 63.3232},

    {"timestamp": "2023-10-01T14:35:00", "position": 64.4234},

    {"timestamp": "2023-10-01T14:45:00", "position": 65.5231},

    {"timestamp": "2023-10-01T14:55:00", "position": 66.6234},

    {"timestamp": "2023-10-01T15:05:00", "position": 67.7236},

    {"timestamp": "2023-10-01T15:15:00", "position": 68.8233},

    {"timestamp": "2023-10-01T15:25:00", "position": 69.9236},

    {"timestamp": "2023-10-01T15:35:00", "position": 71.0238},

    {"timestamp": "2023-10-01T15:45:00", "position": 72.1235},

    {"timestamp": "2023-10-01T15:55:00", "position": 73.2238},

    {"timestamp": "2023-10-01T16:05:00", "position": 74.3239},

    {"timestamp": "2023-10-01T16:15:00", "position": 75.4236}

]

Annex 3 – longitudes.json

[

    {"timestamp": "2023-10-01T10:00:00", "position": -118.2437},

    {"timestamp": "2023-10-01T10:10:00", "position": -117.2437},

    {"timestamp": "2023-10-01T10:20:00", "position": -116.2437},

    {"timestamp": "2023-10-01T10:30:00", "position": -115.2437},

    {"timestamp": "2023-10-01T10:40:00", "position": -114.2437},

    {"timestamp": "2023-10-01T10:50:00", "position": -113.2437},

    {"timestamp": "2023-10-01T11:00:00", "position": -112.2437},

    {"timestamp": "2023-10-01T11:10:00", "position": -111.2437},

    {"timestamp": "2023-10-01T11:20:00", "position": -110.2437},

    {"timestamp": "2023-10-01T11:30:00", "position": -109.2437},

    {"timestamp": "2023-10-01T11:40:00", "position": -108.2437},

    {"timestamp": "2023-10-01T11:50:00", "position": -107.2437},

    {"timestamp": "2023-10-01T12:00:00", "position": -106.2437},

    {"timestamp": "2023-10-01T12:10:00", "position": -105.2437},

    {"timestamp": "2023-10-01T12:20:00", "position": -104.2437},

    {"timestamp": "2023-10-01T12:30:00", "position": -103.2437},

    {"timestamp": "2023-10-01T12:40:00", "position": -102.2437},

    {"timestamp": "2023-10-01T12:50:00", "position": -101.2437},

    {"timestamp": "2023-10-01T13:00:00", "position": -100.2437},

    {"timestamp": "2023-10-01T13:10:00", "position": -99.2437},

    {"timestamp": "2023-10-01T13:15:00", "position": -98.2437},

    {"timestamp": "2023-10-01T13:25:00", "position": -97.2437},

    {"timestamp": "2023-10-01T13:35:00", "position": -96.2437},

    {"timestamp": "2023-10-01T13:45:00", "position": -95.2437},

    {"timestamp": "2023-10-01T13:55:00", "position": -94.2437},

    {"timestamp": "2023-10-01T14:05:00", "position": -93.2437},

    {"timestamp": "2023-10-01T14:15:00", "position": -92.2437},

    {"timestamp": "2023-10-01T14:25:00", "position": -91.2437},

    {"timestamp": "2023-10-01T14:35:00", "position": -90.2437},

    {"timestamp": "2023-10-01T14:45:00", "position": -89.2437},

    {"timestamp": "2023-10-01T14:55:00", "position": -88.2437},

    {"timestamp": "2023-10-01T15:00:00", "position": -87.2437},

    {"timestamp": "2023-10-01T15:10:00", "position": -86.2437},

    {"timestamp": "2023-10-01T15:20:00", "position": -85.2437},

    {"timestamp": "2023-10-01T15:30:00", "position": -84.2437},

    {"timestamp": "2023-10-01T15:40:00", "position": -83.2437},

    {"timestamp": "2023-10-01T15:50:00", "position": -82.2437},

    {"timestamp": "2023-10-01T16:10:00", "position": -81.2437},

    {"timestamp": "2023-10-01T16:20:00", "position": -80.2437}

]