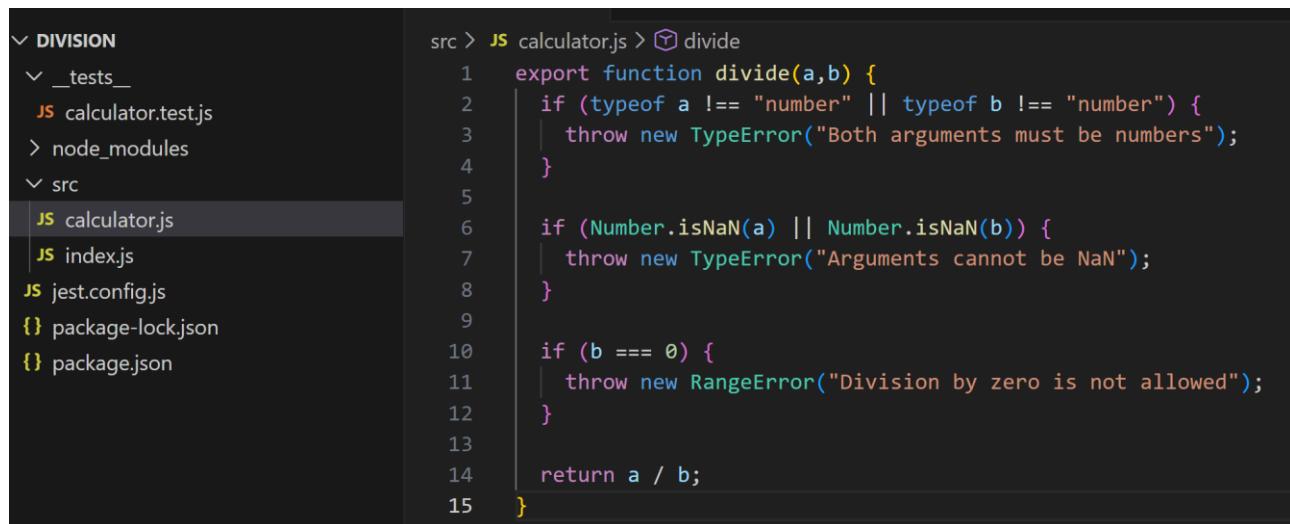


Assignment. Introduction to automated testing

Create a simple Node project based on previous walkthrough (add). Implement basic structure and configuration to run simple Node application written in JavaScript and run tests with Jest testing library.

Structure and files for the project are illustrated on the figure below. App has a simple function, that divides two numbers and does basic input validation.



The screenshot shows a file tree on the left and the content of a file on the right. The file tree includes a DIVISION folder with _tests_, node_modules, and src folders. The src folder contains calculator.js, index.js, jest.config.js, package-lock.json, and package.json. The calculator.js file is selected and shown in the code editor. The code in calculator.js is as follows:

```
src > JS calculator.js > divide
      1 export function divide(a,b) {
      2   if (typeof a !== "number" || typeof b !== "number") {
      3     throw new TypeError("Both arguments must be numbers");
      4   }
      5
      6   if (Number.isNaN(a) || Number.isNaN(b)) {
      7     throw new TypeError("Arguments cannot be NaN");
      8   }
      9
     10  if (b === 0) {
     11    throw new RangeError("Division by zero is not allowed");
     12  }
     13
     14  return a / b;
     15 }
```

Content for src/calculator.js is as follows (you may copy and paste code from here).

```
export function divide(a,b) {
  if (typeof a !== "number" || typeof b !== "number") {
    throw new TypeError("Both arguments must be numbers");
  }

  if (Number.isNaN(a) || Number.isNaN(b)) {
    throw new TypeError("Arguments cannot be NaN");
  }

  if (b === 0) {
    throw new RangeError("Division by zero is not allowed");
  }

  return a / b;
}
```

Content for the src/index.js is as follows (you may copy and paste code from here).

```
let result = divide(2,1);
console.log(result)

console.log(typeof(NaN))
```

If you run the code (npm start), code will print out the result for valid division calculation. To illustrate what is NaN, you'll notice that typeOf NaN is number, but it cannot be used in division.

```
C:\Users\jjuntune24\Downloads\division>npm start
> division@1.0.0 start
> node src/index.js
> node src/index.js

2
number
```

Write tests for the division into __tests__/calculator.test.js file. There should be one positive and at least four negative tests for various cases, where input is invalid (and this you should be able to figure out from the implementation code). Each test is 2 points and maximum number of points for this assignment is 10.

Return **pdf document**, which contains address for **Github** where the code is and a **screenshot** of running the tests in the terminal. An incorrect assignment submission results in 0 points.