Assignment#2 Calculator
OOP

Name: **البدري محمد السيد البدري**

ID: **20010329**

1. Problem Statement

Build a web-based calculator similar to that of windows. The buttons should be web buttons.

2. User Interface

3. Specifications

- The Calculator can perform the following operations:
  [1] +
  [2] –
  [3] X
  [4] ÷
  [5] √x
  [6] X^2
  [7] 1/x
  [8] %
- It can perform multiple operations in one statement before pressing equal
  Eg. 3+5+6+root(25)+25%
- It can perform a single operation each time before pressing equal
  Eg. 3x6
- It has a clear button to reset the calculator.

4. Assumptions

- To use an unary operator the user must enter the number first then enter the unary operator he wants

5. Sample runs

6. Code snippets

[1] Front end (Angular):

➢ app.component.ts

```
import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { HttpHeaders } from '@angular/common/http';



const httpOptions : object = {
  headers : new HttpHeaders({
    'Content-Type' : 'application/json',
    Authorization : 'my-auth-token'
  }), responseType : 'string'
};



@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  constructor(private http: HttpClient){}
  title = 'my-first-app';
  exp : string ='';
  result: string = '';
  operand1: string='';
  operand2: string='';
  operator: string='';
  mainOperator: string='';
  temp : string='';
```

```
creat(){
  this.http.post<string>(`http://localhost:8080/result/${this.mainOperator}/${this.operand1}/${this.operand2}`, JSON, httpOptions)
    .subscribe(result => {
      this.result = result.toString();}
    )
}
pressNum(num:string){
  this.result = this.result + num;
  this.exp = this.exp + num;
}
allclear(){
  this.result="";
  this.exp="";
  this.operand1='';
  this.operand2='';
  this.operator='';
  this.mainOperator='';
}
```

```
pressOperator(op:string){
  if(op=='-' && this.result == '' && this.operand1 == ''){  //for inserting a negative number
    this.result= op ;
    this.exp = this.exp + op;
    return
  }else if(op=='-' && this.result == '' && this.operand1 != '' && this.operand2 == ''){
    this.result = op;
    this.exp = this.exp + op;
    return;
  }
  if(this.mainOperator != ''){          //to perform multiple operations in a single statement
    if(op == 'x^-1' || op == 'X²' || op == '√x' || op == '±' || op == '%' ){
      this.http.post<string>(`http://localhost:8080/result/${op}/${this.result}/${'0'}`, JSON, httpOptions)
      .subscribe(result => {
        this.operand2 = result.toString();
        this.result = this.operand2;
      }
      )
    }
    else{
    this.operand2 = this.result;
    this.http.post<string>(`http://localhost:8080/result/${this.mainOperator}/${this.operand1}/${this.operand2}`, JSON, httpOptions)
    .subscribe(result => {
      this.operand1 = result.toString();}
    )
    this.result = '';
    this.operand2='';
    }
  }
```

```
    if(this.operand1 == ''){
      this.operand1 = this.result;
      this.temp = this.operand1;
    }else{
      this.operand2 = this.result;
      this.temp = this.operand2;
    }
    if(op == 'x^-1' || op == 'X²' || op == '√x' || op == '±' || op == '%'){
      this.operator= op;
    }else{
      this.mainOperator = op ;
      this.operator= op;
    }
    this.result='';
    if(this.operator == '√x'){          //for showing the experession in a good format
      this.exp = this.exp.slice(0,-this.temp.length);
      this.exp = this.exp + "root("+ this.temp + ')';

    }else if(this.operator == 'X²'){
      this.exp = this.exp.slice(0,-this.temp.length);
      this.exp = this.exp + "square("+ this.temp + ')';

    }else if(this.operator == 'x^-1'){
      this.exp = this.exp.slice(0,-this.temp.length);
      this.exp = this.exp + "1/("+ this.temp + ')';

    }else if(this.operator == '±'){
      this.exp = this.exp.slice(0,-this.temp.length);
      this.exp = this.exp + "-("+ this.temp + ')';

    }
    else{
      this.exp = this.exp + op;
    }
```

```
    //for handling inserting a uniray operator for the first operand in the expression
    if(this.operator == 'x^-1' || this.operator == 'X²' || this.operator == '√x' || this.operator == '±' || this.operator == '%' ){

      if(this.mainOperator == ''){
        this.http.post<string>(`http://localhost:8080/result/${this.operator}/${this.operand1}/${'0'}`, JSON, httpOptions)
        .subscribe(result => {
          this.result = result.toString();
          this.operand1 = this.result;

        }
        )
      }
    }

  }
```

```
  pressEqual(){

    this.operand2 = this.result;
    this.creat();
    this.operand1 = '';
    this.operand2 = '';
    this.mainOperator='';


  }
  backSpace(){
   this.result= this.result.slice(0, -1);
   this.exp= this.exp.slice(0, -1);
  }

}

```

➢ html:

```
<!doctype html>
<html lang="en">
<body >
  <div class="container">
    <div class="result">
      <span class="res">{{result}}</span>
      <footer><span class="exp">{{exp}}</span></footer>
    </div>
    <div class="buttons">
      <button class="item outer"(click)="pressOperator('%')">%</button>
      <button class="item outer"(click)="allclear()">CE</button>
      <button class="item outer"(click)="allclear()">C</button>
      <button class="item outer"(click)="backSpace()">⌫</button>
      <button class="item outer"(click)="pressOperator('x^-1')">1/x</button>
      <button class="item outer"(click)="pressOperator('X²')">X²</button>
      <button class="item outer"(click)="pressOperator('√x')">√x</button>
      <button class="item outer"(click)="pressOperator('÷')">÷</button>
      <button class="item"(click)="pressNum('7')">7</button>
      <button class="item"(click)="pressNum('8')">8</button>
      <button class="item"(click)="pressNum('9')">9</button>
      <button class="item outer"(click)="pressOperator('x')">X</button>
      <button class="item"(click)="pressNum('4')">4</button>
      <button class="item"(click)="pressNum('5')">5</button>
      <button class="item"(click)="pressNum('6')">6</button>
      <button class="item outer"(click)="pressOperator('-')">-</button>
      <button class="item"(click)="pressNum('1')">1</button>
      <button class="item"(click)="pressNum('2')">2</button>
      <button class="item"(click)="pressNum('3')">3</button>
      <button class="item outer"(click)="pressOperator('+')">+</button>
      <button class="item"(click)="pressOperator('±')">+/-</button>
      <button class="item"(click)="pressNum('0')">0</button>
      <button class="item"(click)="pressNum('.')">.</button>
      <button class="item equal"(click)="pressEqual()">=</button>
    </div>
  </div>
</body>
</html>
```

➢ CSS:

```css
.result span{
    color: rgb(234, 232, 240);
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.res{
    font-size: 50px;
    font-weight: bold;
}
.result{
    background-color: rgb(77, 138, 168);
    width: auto;
    height: 100px;
    border-radius: 5%;
    text-align: right;


}
.exp{
    font-size: 20px;
}
.buttons{
    display: grid;
    grid-template-columns: auto auto auto auto ;
    background-color: rgb(8, 35, 48);
    border-radius: 15%;
    box-shadow: 5px 3px 52px 20px rgba(233, 236, 235, 0.24);
}
.item{
    color: aliceblue;
    width: 100px;
    height: 90px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 30px;
    background-color: rgb(121, 166, 189);
    cursor: pointer;
    border-radius: 15%;

}
.item:active {
    transform: scale(0.99);
    background-color: rgb(161, 219, 255);
    box-shadow: 5px 3px 52px 1px rgba(236, 233, 233, 0.24);

}
.outer{
    background-color: rgb(77, 138, 168);
}
.equal{
    background-color: rgb(12, 212, 219);
}
```

[2] Back End(SpringBoot):

➢ Controller:

```java
package com.calculator.calculator;

import org.springframework.web.bind.annotation.*;
import java.util.Objects;

@RestController
@CrossOrigin(origins="http://localhost:4200/")

public class controller {
    @PostMapping (value = "/result/{first}/{second}/{third}")

    public String calc(@PathVariable String first, @PathVariable String second,@PathVariable String third) {
       if(Objects.equals(first, b: "+") || Objects.equals(first, b: "-") || Objects.equals(first, b: "x") || Objects.equals(first, b: "÷")){
           BinaryOperations Binary = new BinaryOperations();
           return Binary.result(first,second,third);
        }else{
           UniaryOperations un = new UniaryOperations();
           return un.result(first,second);
       }
    }
}
```

➢ UinaryOperations calls:

```java
public class UniaryOperations {
    1 usage
    public double percentage(String num1) { return Float.parseFloat(num1)/100.0 ; }
    1 usage
    public float resiprocal(String num1) { return 1/(Float.parseFloat(num1)); }
    1 usage
    public float square(String num1) { return Float.parseFloat(num1) *Float.parseFloat(num1); }
    1 usage
    public double root(String num1) { return Math.sqrt(Float.parseFloat(num1)) ; }
    1 usage
    public float invertSign(String num1) { return Float.parseFloat(num1)*-1 ; }
    1 usage
    public String result(String operation, String num1){
        if(Objects.equals(operation, b: "٪")){
            return String.valueOf(percentage(num1));
        }
        if(Objects.equals(operation, b: "x^-1")){
            return String.valueOf(resiprocal(num1));
        }
        if(Objects.equals(operation, b: "X²")){
            return String.valueOf(square(num1));
        }
        if(Objects.equals(operation, b: "√x")){
            return String.valueOf(root(num1));
        }
        if(Objects.equals(operation, b: "±")){
            return String.valueOf(invertSign(num1));
        }

        return null;
```

➢ BinaryOperations class:

```
public class BinaryOperations {
    1 usage
    public float add(String num1, String num2) { return Float.parseFloat(num1) + Float.parseFloat(num2); }
    1 usage
    public float subtract(String num1, String num2) { return Float.parseFloat(num1) - Float.parseFloat(num2); }
    1 usage
    public float multiply(String num1, String num2) { return Float.parseFloat(num1) *Float.parseFloat(num2); }
    1 usage
    public float divide(String num1, String num2){return Float.parseFloat(num1) /Float.parseFloat(num2);}
    1 usage
    public String result(String operation, String num1 , String num2){
        if(Objects.equals(operation, b: "+")){
            return String.valueOf(add(num1,num2));
        }
        if(Objects.equals(operation, b: "-")){
            return String.valueOf(subtract(num1,num2));
        }
        if(Objects.equals(operation, b: "x")){
            return String.valueOf(multiply(num1,num2));
        }
        if(Objects.equals(operation, b: "÷")){
            return String.valueOf(divide(num1,num2));
        }
       return null;
    }

}
```

7. How to open the website:
   1) Open the front-end file.
   2) On that bar that contains the directory of the file click on it and write cmd.
   3) When the cmd opens write ng serve
   4) Open the springboot file using IntelliJ IDE
   5) Press run.
   6) Open your browser and write localhost:4200