TALLINNA TEHNIKAÜLIKOOL
School of Information Technologies

Ahmed Abdullajev 192816IADB

HOMEWORK 1 LEG 4 – CALCULATOR ON TYPESCRIPT(BABEL, WEBPACK)

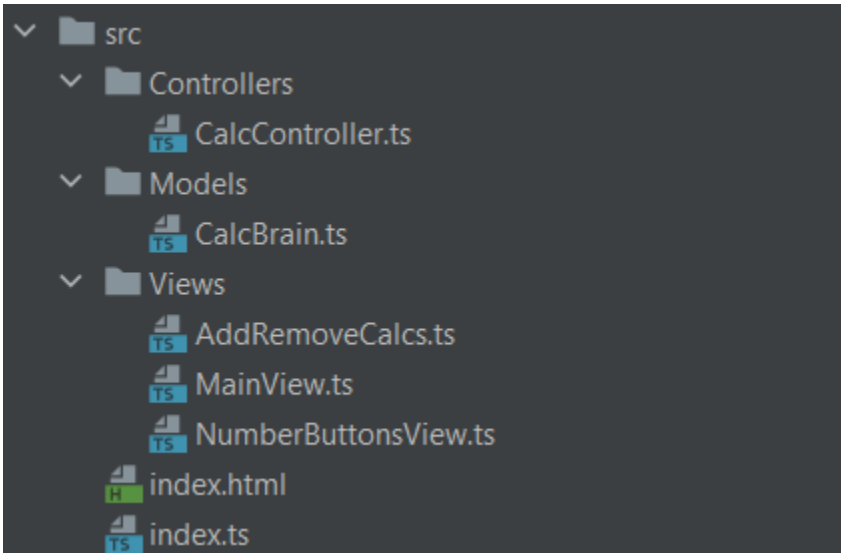
JavaScript

TALLINN 2022

SCREENSHOOTS

You can check my code below or in my school repository:
https://gitlab.cs.ttu.ee/ahabdu/icd0006-21-22-s

PROJECT STRUCTURE



HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css
" rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <title>Calc</title>
</head>
<body>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.mi
n.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
</html>
```

JAVASCRIPT CODE:

Index.ts

```
import AddRemoveCalcs from "./Views/AddRemoveCalcs";

let addRemoveCalcs = new AddRemoveCalcs();

addRemoveCalcs.addCalc();
addRemoveCalcs.removeCalc();
```

Calcbrain.ts

```
export default class CalcBrain {
    private numberOne: number | string | null;
    private numberTwo: number | string | null;
    private result: number | string | null;
    private additionalNumber: number | string | null;
    private plus: boolean;
    private minus: boolean;
    private divide: boolean;
    private multiply: boolean;
    private lastOperation: string;

    constructor() {
        //Numbers
        this.numberOne = null;
        this.numberTwo = null;
        this.result = null;
        this.additionalNumber = null;

        //Operations
        this.plus = false
        this.minus = false
        this.divide = false
        this.multiply = false

        this.lastOperation = '';
    }

    getNumberOne() {
        return this.numberOne;
    }

    setNumberOne(num : string) {
        this.numberOne = num;
    }

    getNumberTwo() {
        return this.numberTwo;
    }
```

```
    setNumberTwo(num : string) {
        this.numberTwo = num;
    }

    getResult() {
        return this.result;
    }

    getAdditionalNumber() {
        return this.additionalNumber;
    }

    setAdditionalNumber(num : string) {
        this.additionalNumber = num;
    }

    setDefault() {
        this.numberOne = null;
        this.numberTwo = null;
        this.result = null;
        this.additionalNumber = null;
        this.plus = false
        this.minus = false
        this.divide = false
        this.multiply = false
        this.lastOperation = '';
    }
    changeNumberNegativeOrPositive(number : string){
        if(number == "first" && this.numberOne != null) {
            if (typeof this.numberOne !== "number" &&
this.numberOne?.indexOf("-") >= 0) {
                this.numberOne = this.numberOne.substring(1)
            } else {
                this.numberOne = "-" + this.numberOne
            }
            return this.numberOne
        }
        if(number == "second" && this.numberTwo != null) {
            if (typeof this.numberTwo !== "number" &&
this.numberTwo?.indexOf("-") >= 0) {
                this.numberTwo = this.numberTwo.substring(1)
            } else {
                this.numberTwo = "-" + this.numberTwo
            }
            return this.numberTwo
        }
        if(number == "additional" && this.additionalNumber != null) {
            if (typeof this.additionalNumber !== "number" &&
this.additionalNumber?.indexOf("-") >= 0) {
                this.additionalNumber = this.additionalNumber.substring(1)
            } else {
                this.additionalNumber = "-" + this.additionalNumber
            }
            return this.additionalNumber
        }
        return "";
```

```
    }
    clearNumber(number : string){
        if(number == "first" && this.numberOne != null) {
            if (typeof this.numberOne !== "number" &&
this.numberOne?.replace(/\D/g, '').length > 1) {
                this.numberOne = this.numberOne.substring(0,
this.numberOne.length - 1)
            } else {
                this.numberOne = "0";
            }
            return this.numberOne
        }
        if(number == "second" && this.numberTwo != null) {
            if (typeof this.numberTwo !== "number" &&
this.numberTwo?.replace(/\D/g, '').length > 1) {
                this.numberTwo = this.numberTwo.substring(0,
this.numberTwo.length - 1)
            } else {
                this.numberTwo = "0";
            }
            return this.numberTwo
        }
        if(number == "additional" && this.additionalNumber != null) {
            if (typeof this.additionalNumber !== "number" &&
this.additionalNumber?.replace(/\D/g, '').length > 1) {
                this.additionalNumber = this.additionalNumber.substring(0,
this.additionalNumber.length - 1)
            } else {
                this.additionalNumber = "0";
            }
            return this.additionalNumber
        }

        return "";

    }
    doMath(operation : string) : string | null {
        if (operation == 'minus') {
            this.setMinusTrueOthersFalse()
            if (this.minus == true && this.numberOne != null &&
this.numberTwo != null && this.result == null) {
                this.numberOne = this.numberOne.toString();
                this.numberTwo = this.numberTwo.toString();
                    this.result = parseFloat(this.numberOne) -
parseFloat(this.numberTwo);
                return this.result!.toString();
            } else if (this.minus == true && this.numberOne != null &&
this.numberTwo != null && this.result != null
                && this.additionalNumber != null) {
                this.additionalNumber = this.additionalNumber.toString()
                this.result = this.result.toString();
                    this.result = parseFloat(this.result) -
parseFloat(this.additionalNumber);
                this.additionalNumber = null;
                return this.result!.toString();
            }
        } else if (operation == 'plus') {
```

```
            console.log("ples")
            this.setPlusTrueOthersFalse()
            if (this.plus == true && this.numberOne != null && this.numberTwo
!= null && this.result == null) {
                this.numberOne = this.numberOne.toString();
                this.numberTwo = this.numberTwo.toString();
                    this.result = parseFloat(this.numberOne) +
parseFloat(this.numberTwo);
                return this.result!.toString();
            } else if (this.plus == true && this.numberOne != null &&
this.numberTwo != null && this.result != null
                && this.additionalNumber != null) {
                this.additionalNumber = this.additionalNumber.toString()
                this.result = this.result.toString();
                this.result = parseFloat(this.result) +
parseFloat(this.additionalNumber);
                this.additionalNumber = null;
                return this.result!.toString();
            }
        } else if (operation == 'divide') {
            this.setDivideTrueOthersFalse()
            if (this.divide == true && this.numberOne != null &&
this.numberTwo != null && this.result == null) {
                this.numberOne = this.numberOne.toString();
                this.numberTwo = this.numberTwo.toString();
                    this.result = parseFloat(this.numberOne) /
parseFloat(this.numberTwo);
                return this.result!.toString();
            } else if (this.divide == true && this.numberOne != null &&
this.numberTwo != null && this.result != null
                && this.additionalNumber != null) {
                this.additionalNumber = this.additionalNumber.toString()
                this.result = this.result.toString();
                    this.result = parseFloat(this.result) /
parseFloat(this.additionalNumber);
                this.additionalNumber = null;
                return this.result!.toString();
            }
        } else if (operation == 'multiply') {
            this.setMultiplyTrueOthersFalse()
            if (this.multiply == true && this.numberOne != null &&
this.numberTwo != null && this.result == null) {
                this.numberOne = this.numberOne.toString();
                this.numberTwo = this.numberTwo.toString();
                    this.result = parseFloat(this.numberOne) *
parseFloat(this.numberTwo);
                return this.result!.toString();
            } else if (this.multiply == true && this.numberOne != null &&
this.numberTwo != null && this.result != null
                && this.additionalNumber != null) {
                this.additionalNumber = this.additionalNumber.toString()
                this.result = this.result.toString();
                    this.result = parseFloat(this.result) *
parseFloat(this.additionalNumber);
                this.additionalNumber = null;
                return this.result!.toString();
            }
```

```
        }
        console.warn("here: ", operation)
        if(this.result == null){
            return null;
        }
        return this.result!.toString();
    }

    getPlus() {
        return this.plus;
    }

    getMinus() {
        return this.minus;
    }

    getMultiply() {
        return this.multiply;
    }

    getDivide() {
        return this.divide;
    }

    getLastOperation() {
        return this.lastOperation;
    }

    setPlusTrueOthersFalse() {
        this.plus = true;
        this.lastOperation = 'plus'

        this.minus = false;
        this.multiply = false;
        this.divide = false;
    }

    setMinusTrueOthersFalse() {
        this.minus = true;
        this.lastOperation = 'minus'

        this.plus = false;
        this.multiply = false;
        this.divide = false;
    }

    setMultiplyTrueOthersFalse() {
        this.multiply = true;
        this.lastOperation = 'multiply'

        this.plus = false;
        this.minus = false;
        this.divide = false;
    }

    setDivideTrueOthersFalse() {
        this.divide = true;
```

```
        this.lastOperation = 'divide'

        this.plus = false;
        this.minus = false;
        this.multiply = false;
    }


}
```

MainView.ts

```
export default function MainView(btns : string, order : string) {
    let content = document.createElement('div');
    content.className = "card text-center w-50";
    content.id = "calc"+order;
    content.style.margin = "auto";
    let spanDiv = document.createElement('div');
    spanDiv.className = "card-header bg-dark w-100 h-100";
    content.append(spanDiv)
    let span = document.createElement('span');
    span.className = "d-block p-2 bg-dark text-white";
    span.id="monitor"+order;
    span.style.height = "24px";
    spanDiv.append(span)

    let body = document.createElement('div');
    body.className = "card-body";
    content.append(body)

    let btnGroup = document.createElement('div');
    btnGroup.className = "btn-group w-100 justify-content-center
quickmath"+order;

    btnGroup.innerHTML = btns;
    body.append(btnGroup);

    let footer = document.createElement('div');
    footer.className = "card-footer";

    let plus = document.createElement('button');
    plus.className = "btn btn-success plusik"+order;
    plus.innerText = "+";

    let minus = document.createElement('button');
    minus.className = "btn btn-success minusik"+order;
    minus.innerText = "-";


    let equals = document.createElement('button');
    equals.className = "btn btn-success equals"+order;
    equals.innerText = "=";

    let multiply = document.createElement('button');
    multiply.className = "btn btn-success multiply"+order;
    multiply.innerText = "*";

    let divide = document.createElement('button');
    divide.className = "btn btn-success divide"+order;;
    divide.innerText = "/";

    let todefault = document.createElement('button');
    todefault.className = "btn btn-warning todefault"+order;;
    todefault.style.marginLeft = "5px";
    todefault.innerText = "AC";

    let dot = document.createElement('button');
```

```
    dot.className = "btn btn-secondary dot"+order;
    dot.style.marginRight = "5px";
    dot.innerText = ".";

    let toplusorminus = document.createElement('button');
    toplusorminus.className = "btn btn-secondary toplusorminus"+order;;
    toplusorminus.style.marginRight = "10px";
    toplusorminus.innerText = "+/-";

    let clearnum = document.createElement('button');
    clearnum.className = "btn btn-warning clearnum"+order;;
    clearnum.style.marginLeft = "5px"
    clearnum.innerText = "C";

    footer.append(dot)
    footer.append(toplusorminus)
    footer.append(plus)
    footer.append(divide)
    footer.append(minus)
    footer.append(multiply)
    footer.append(equals)
    footer.append(todefault)
    footer.append(clearnum)

    content.append(footer)

    return content;
}
```

NumberButtonsView.ts

```
export default function NumberButtonsView() {
    let numberButtons = "";
    let iter = 0;
    for (let i = 0; i < 3; i++) {
        numberButtons = numberButtons + '<div class="btn-group-vertical w-
100">';
        for (let j = 0; j < 3; j++) {
            numberButtons = numberButtons +
                `<button class="btn btn-primary num" value="${iter}"
number="${iter}">${iter}</button>`
            iter++;
            if (iter == 8) {
                numberButtons = numberButtons +
                    `<button class="btn btn-primary num" value="${iter}"
number="${iter}">${iter}</button>`
                iter++;
            }
        }
        numberButtons = numberButtons + '</div>';
    }

    return numberButtons;

}
```

CalcController.ts

```
import CalcBrain from "../Models/CalcBrain";

export default class CalcController {
    private calc: CalcBrain;
    private order: string;
    constructor(calcBrain : CalcBrain, order : string) {
        this.calc = calcBrain
        this.order = order;
    }

    showNum(num : string | null = "") {
        let docm = document.querySelector("#monitor"+this.order)
        if (num != null) {
            docm!.innerHTML = "";
            let nums = num + "";
            if(nums.indexOf(".") >= 0){
                docm!.innerHTML = nums
            }else{ // to not return number starting with 0
                docm!.innerHTML = parseFloat(nums) + "";
            }
        } else {
            docm!.innerHTML = "";
        }

    }

    setValues(number : number , dot = false, toplusorminus = false, clearnum
= false) {

        if (this.calc.getNumberOne() == null && toplusorminus == false &&
clearnum == false) { // we are in first number scope and setting first number
for it
            this.calc.setNumberOne(number + "");
            this.showNum(number.toString())
        } else if (this.calc.getPlus() == false && this.calc.getMinus() ==
false && this.calc.getDivide() == false
            && this.calc.getMultiply() == false && this.calc.getNumberOne()
!= null
            && this.calc.getResult() == null) {  // we are in first number
scope and adding next numbers for it
            if (dot == true &&
this.calc!.getNumberOne()!.toString().indexOf(".") <= -1 && toplusorminus ==
false) {
                let numStr = this.calc.getNumberOne() + "."; // adding one
dot
                this.calc.setNumberOne(numStr)
                this.showNum(numStr)
            } else if (dot == false && toplusorminus == false && clearnum ==
false) { // adding next numbers for first number
                let numStr = this.calc.getNumberOne() + "" + number;
                this.calc.setNumberOne(numStr)
                this.showNum(numStr)
            }
            if(toplusorminus == true){ // change num to positive or negative
               let num = this.calc.changeNumberNegativeOrPositive('first')
```

```
                this.showNum(num)
            }
            if(clearnum == true){
                let num = this.calc.clearNumber('first')
                this.showNum(num)
            }
        }
        if (this.calc.getNumberTwo() == null && (this.calc.getPlus() == true
|| this.calc.getMinus() == true
            || this.calc.getDivide() == true || this.calc.getMultiply() ==
true) && this.calc.getResult() == null && toplusorminus == false && clearnum
== false) {
            this.calc.setNumberTwo(number + "");
            this.showNum(number.toString())
        } else if ((this.calc.getPlus() == true || this.calc.getMinus() ==
true || this.calc.getDivide() == true
            || this.calc.getMultiply() == true) && this.calc.getNumberTwo()
!= null && this.calc.getResult() == null) {
            if (dot == true &&
this.calc.getNumberTwo()!.toString().indexOf(".") <= -1 && toplusorminus ==
false) {
                let numStr = this.calc.getNumberTwo() + ".";
                this.calc.setNumberTwo(numStr)
                this.showNum(numStr)
            } else if (dot == false && toplusorminus == false && clearnum ==
false) {
                let numStr = this.calc.getNumberTwo() + "" + number;
                this.calc.setNumberTwo(numStr);
                this.showNum(numStr)
            }
            if(toplusorminus == true){
                let num = this.calc.changeNumberNegativeOrPositive('second')
                this.showNum(num)
            }
            if(clearnum == true){
                let num = this.calc.clearNumber('second')
                this.showNum(num)
            }
        }

        if (this.calc.getResult() != null && (this.calc.getPlus() == true ||
this.calc.getMinus() == true
                || this.calc.getDivide() == true || this.calc.getMultiply()
== true)
            && this.calc.getAdditionalNumber() == null && toplusorminus ==
false) {
            this.calc.setAdditionalNumber(number + "");
            this.showNum(number.toString())
        } else if (this.calc.getResult() != null && (this.calc.getPlus() ==
true || this.calc.getMinus() == true
                || this.calc.getDivide() == true || this.calc.getMultiply()
== true)
            && this.calc.getAdditionalNumber() != null) {
            if (dot == true &&
this.calc.getAdditionalNumber()!.toString().indexOf(".") <= -1 &&
toplusorminus == false) {
                let numStr = this.calc.getAdditionalNumber() + ".";
```

```
                this.calc.setAdditionalNumber(numStr)
                this.showNum(numStr)
            } else if (dot == false && toplusorminus == false && clearnum ==
false) {
                let numStr = this.calc.getAdditionalNumber() + "" + number;
                this.calc.setAdditionalNumber(numStr);
                this.showNum(numStr)
            }
            if(toplusorminus == true){
                let num =
this.calc.changeNumberNegativeOrPositive('additional')
                this.showNum(num)
            }
            if(clearnum == true){
                let num = this.calc.clearNumber('additional')
                this.showNum(num)
            }
        }
    }

    doCalculations(operation : string) {
        let res = this.calc.doMath(operation)
        this.showNum(res)
    }

    toDefaultValues() {
        this.calc.setDefault();
        this.showNum(null)
    }

}
```

AddRemoveCalcs

```
import NumberButtonsView from "./NumberButtonsView";
import MainView from "./MainView";
import CalcBrain from "../Models/CalcBrain";
import CalcController from "../Controllers/CalcController";

export default class AddRemoveCalcs {
    private globalCounter: number;
    constructor() {
        this.globalCounter = 0;
    }
    addNewCalc(globalNum : number) {
        let order = "_0"+globalNum;

        let btnsView = NumberButtonsView();
        let viewer = MainView(btnsView, order);

        let my_calc = new CalcBrain();
        let my_controller = new CalcController(my_calc, order);
        document.body.appendChild(viewer)
        let doc = document.querySelector('.quickmath'+order);
        doc!.addEventListener("click", (e: Event) => {
            if(e.target instanceof HTMLButtonElement) {
                const target = e.target;
                let num =
parseFloat(target.closest('.num')!.getAttribute('number')!.toString())
                my_controller.setValues(num, false, false)
            }
        })
        document.querySelector('.dot'+order)!.addEventListener("click", ()=>
{
            my_controller.setValues(0, true, false)
        })
        let toplusorminus = document.querySelector('.toplusorminus'+order);
        toplusorminus!.addEventListener('click', ()=> {
            my_controller.setValues(0, false, true)
        });
        let clearNum = document.querySelector('.clearnum'+order);
        clearNum!.addEventListener('click', function (){
            my_controller.setValues(0, false, false, true)
        })
        let plusik = document.querySelector('.plusik'+order);
        plusik!.addEventListener('click', function () {
            my_controller.doCalculations('plus')
        })
        let minusik = document.querySelector('.minusik'+order);
        minusik!.addEventListener('click', function () {
            my_controller.doCalculations('minus')
        })
        let multiply = document.querySelector('.multiply'+order);
        multiply!.addEventListener('click', function () {
            my_controller.doCalculations('multiply')
        })
```

```
        let divide = document.querySelector('.divide'+order);
        divide!.addEventListener('click', function () {
            my_controller.doCalculations('divide')
        })
        let todefault = document.querySelector('.todefault'+order)
        todefault!.addEventListener('click', function () {
            my_controller.toDefaultValues()
        })
        let equals = document.querySelector('.equals'+order);
        equals!.addEventListener('click',function () {
            my_controller.doCalculations(my_calc.getLastOperation());
        })

    }
    addCalc(){
        let buttonAdd = document.createElement("button");
        buttonAdd.textContent = "Add Calc";
        buttonAdd.className = "btn btn-primary"
        document.body.appendChild(buttonAdd)
        buttonAdd.onclick = () => {
            this.globalCounter++;
            this.addNewCalc(this.globalCounter);
        }
    }
    removeOldCalc(calc:string) {
        let removeCalc = document.body.querySelector(`#calc_0${calc}`);
        removeCalc!.remove()
    }
    removeCalc(){
        let buttonRemove = document.createElement("button");
        buttonRemove.textContent = "Remove Calc";
        buttonRemove.className = "btn btn-warning"
        document.body.appendChild(buttonRemove)
        buttonRemove.onclick = () => {
            this.removeOldCalc(this.globalCounter.toString());
            this.globalCounter--;
        }
    }

}
```