

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Институт Информатики и кибернетики   
Кафедра Программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
  
к лабораторной работе №5 по дисциплине «Автоматизация тестирования»

Обучающийся группы 6232-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Александров А.А.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лобанков А.А.

Самара 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc152765567)

[2 Результаты работы 4](#_Toc152765568)

[2.1 Результаты задания 4](#_Toc152765569)

[3 Вывод 7](#_Toc152765570)

[ПРИЛОЖЕНИЕ А Листинг программы 8](#_Toc152765571)

[ПРИЛОЖЕНИЕ Б Cypress тесты 13](#_Toc152765572)

1. Постановка задачи

**Цель и постановка задания**

В процессе выполнения заданий ознакомиться с библиотеками для сквозного тестирования front-end приложений. За основу взять приложение из 4 лабораторной работы. В качестве фреймворка можно использовать Protractor (selenium) или cypress.

**Задание**

Необходимо описать несколько тест-кейсов. Для обращения к одним и тем же элементам страницы рекомендуется использовать паттерн Page Object.

1) Проверить, что у нас на странице существуют инпуты, дропдаун и кнопка для получения результата.

2) Проверить, что корректно работают все арифметические действия.

3) Проверить, что в поля ввода можно ввести только цифры.

4) Проверить, что при выборе деления во второй инпут нельзя ввести 0.

5) Проверить, что при выборе шестнадцатеричной системы счисления в инпуты можно вводить соответствующие буквы.

6) Проверить, цвета результата в зависимости от знака.

1. Результаты работы
   1. Результаты задания

За основу взято приложение из 4 лабораторной работы и протестировано при помощи сквозного (end-to-end) тестирования. В качестве фреймворка для сквозного тестирования использовался Cypress. Код фронтенд приложения описан в приложении А.

Файлы тестов находятся в e2e каталоге. Каждый тест представляет собой файл TypeScript с расширением .cy.ts. Сами тесты структурированы с помощью тестовой платформы Mocha. Утверждения (также называемые ожиданиями) написаны с использованием Chai.

Сами тесты запускаются командой ng e2e в терминале.

1) Проверяем, что у нас на странице существуют инпуты, дропдаун и кнопка для получения результата.

2) Проверяем, что корректно работают все арифметические действия.

3) Проверяем, что в поля ввода можно ввести только цифры.

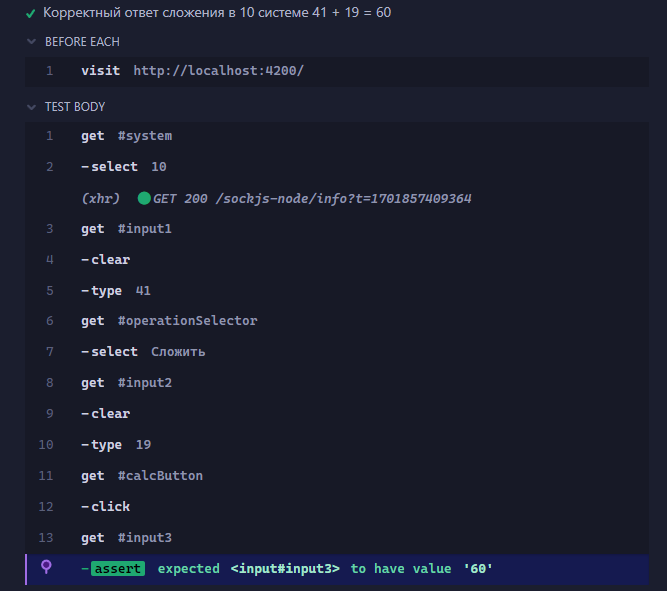
4) Проверяем, что при выборе деления во второй инпут нельзя ввести 0.

5) Проверяем, что при выборе шестнадцатеричной системы счисления в инпуты можно вводить соответствующие буквы.

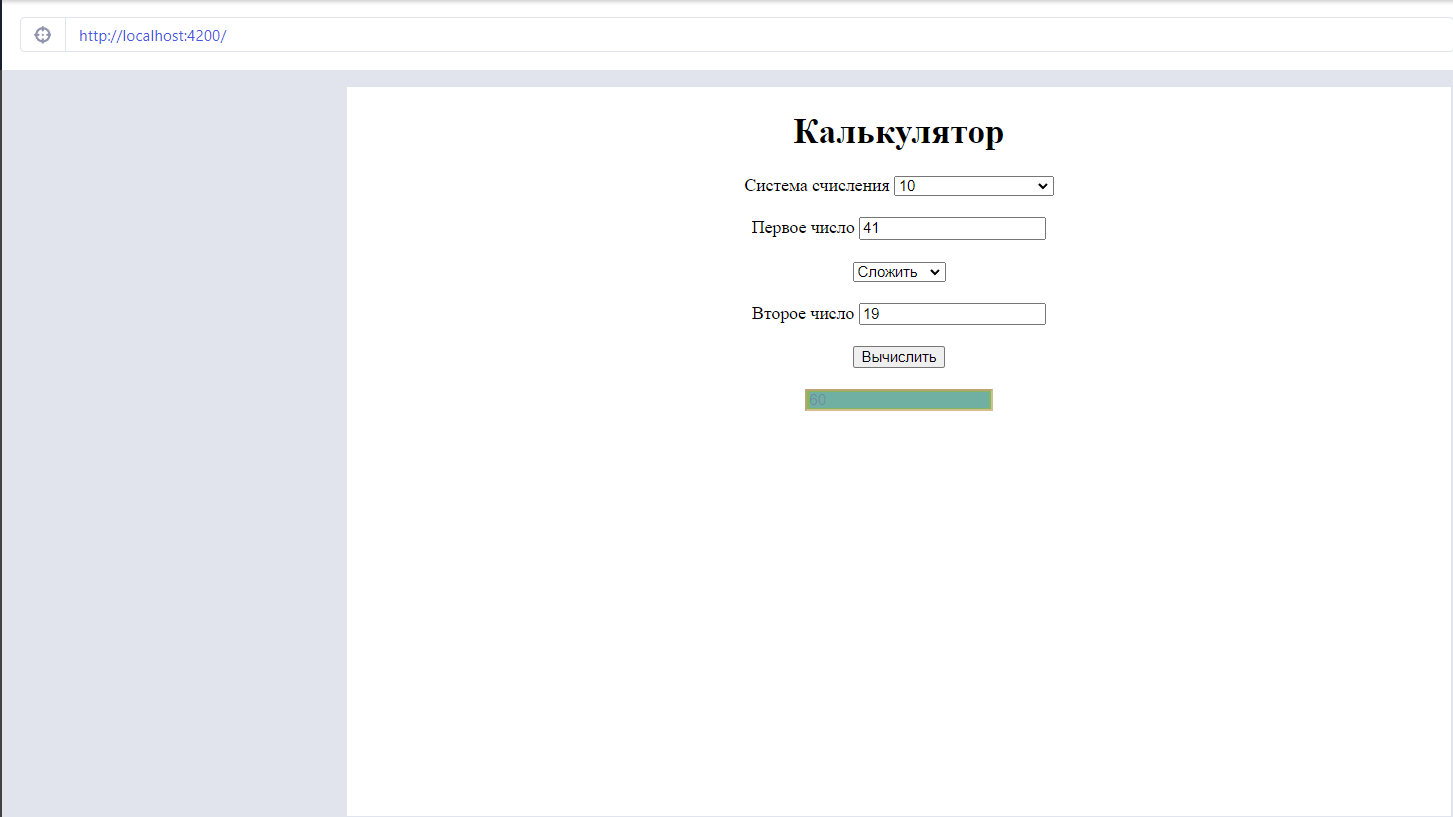
6) Проверяем, цвета результата в зависимости от знака.

Все проведенные тесты описаны в приложении Б.

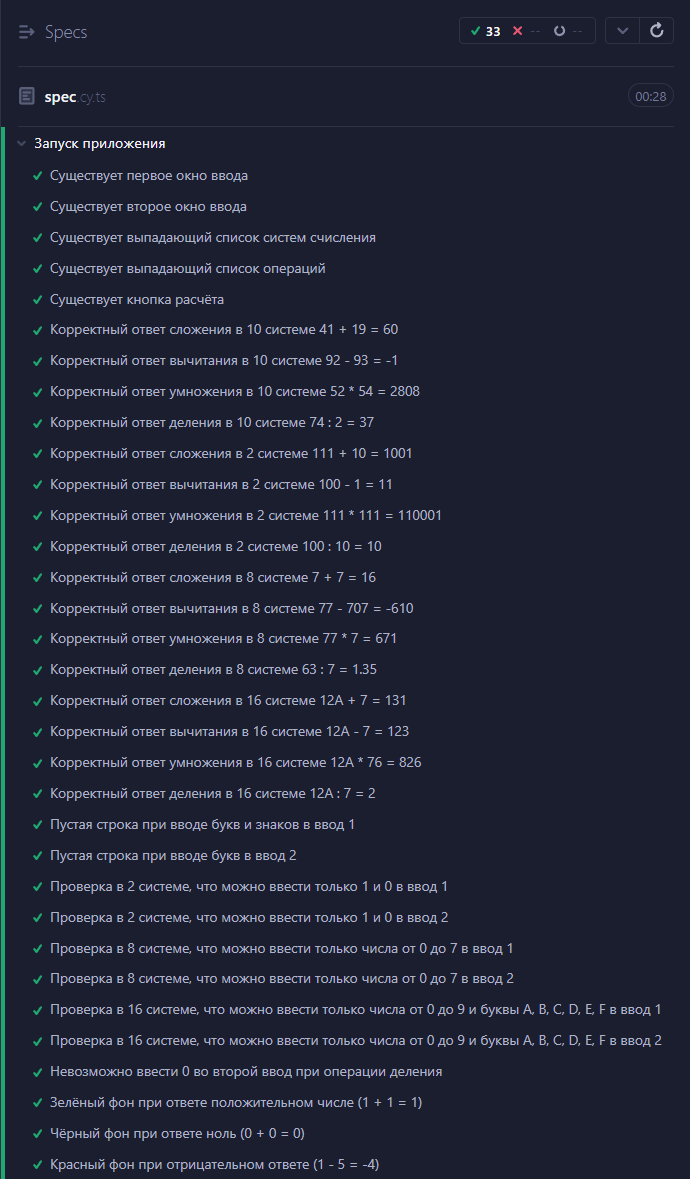
На рисунке 1 представлен пример описания действий в Cypress для операции сложения (41+19) в 10 системе счисления с ответом 60.

  
Рисунок 1 – Описание действий нашего теста

На рисунке 2 выведен результат для данного тест-кейса.

  
Рисунок 2 – Результат нашего тест-кейса

На рисунке 3 представлены все результаты тестов, которые выполнены при помощи Cypress

  
Рисунок 2 – Результат выполнения сквозного тестирования при помощи фреймворка Cypress

1. Вывод

В результате выполнения лабораторной работы на основании реализованого приложения «Калькулятор систем счисления» при помощи фреймворка Angular в лабораторной работе 4.

Описаны необходимые тест-кейсы при помощи фреймворка для сквозного тестирования (end-to-end) Cypress.

1) Проверено, что у нас на странице существуют инпуты, дропдаун и кнопка для получения результата.

2) Проверено, что корректно работают все арифметические действия.

3) Проверено, что в поля ввода можно ввести только цифры.

4) Проверено, что при выборе деления во второй инпут нельзя ввести 0.

5) Проверено, что при выборе шестнадцатеричной системы счисления в инпуты можно вводить соответствующие буквы.

6) Проверены, цвета результата в зависимости от знака.

Написан отчёт.

[Исходный код доступен по ссылке](https://github.com/Allexeey/autotests-lab1/tree/lab-45)

ПРИЛОЖЕНИЕ А  
Листинг программы

<div style="text-align:center">

<h1>

{{ title }}

</h1>

<inputs (calcDataEvent)="calc($event)" (resultEvent)="updateResult($event)" ></inputs>

<!--<input [input]="num3 | rounding:2" [redblackgreen] id="input3" name="input3" type="text" />-->

<input [input]="num3" [redblackgreen]="" id="input3" name="input3" type="text" />

</div>

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

import { AppService } from './service/app.component.service';

@Component({

selector: 'app-root',

templateUrl: './app.component.html'

})

export class AppComponent {

constructor(private appService: AppService){};

title: string = 'Калькулятор';

updateResult(result: string): void {

this.num3 = (result);

}

num3: string = '2';

operations = [

"Сложить",

"Вычесть",

"Поделить",

"Умножить"

];

calc(calcData: CalcData) {

switch (calcData.selectedValue) {

case (this.operations)[0]: {

this.appService.addition(calcData.num1, calcData.num2, calcData.system).subscribe()

break;

}

case (this.operations)[1]: {

this.appService.subtraction(calcData.num1, calcData.num2, calcData.system).subscribe()

break;

}

case (this.operations)[2]: {

this.appService.division(calcData.num1, calcData.num2, calcData.system).subscribe()

break;

}

case (this.operations)[3]: {

this.appService.multiplication(calcData.num1, calcData.num2, calcData.system).subscribe()

break;

}

}

}

}

export interface CalcData {

num1: string;

num2: string;

selectedValue: string;

system: string;

}

<label>Система счисления </label>

<select

id="system"

class="custom-select mb-3"

(change)="changeSystem($event)"

>

<option value="">Система счисления</option>

<option \*ngFor="let s of System">

{{ s }}

</option>

</select>

<br>

<br>

<label>Первое число </label>

<input [(ngModel)]="inputNum1" id="input1" name="inputNum1" type="text" (keypress)="keyPress($event, getPatternParam())">

<br>

<br>

<select [(ngModel)]="inputSelectedValue" id="operationSelector">

<option \*ngFor="let o of operations" [ngValue]="o">{{o}}</option>

</select>

<br>

<br>

<label>Второе число </label>

<input [(ngModel)]="inputNum2" (keypress)="keyPress($event, getPatternParam())" id="input2" name="inputNum2" type="text" oninput='let e = document.getElementById("operationSelector");

let text = e.options[e.selectedIndex].text;

if (text.localeCompare("Поделить") === 0)

{

if(!this.value.match("^[1-9][0-9]\*$"))

this.value="";

}

else {

if(!this.value.match("^[0-9]\*$"))

this.value="";

}'

pattern="^[1-9][0-9]\*$" required/>

<br>

<br>

<button (click)="emitCalcData()" id="calcButton">Вычислить</button>

<br>

<br>

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

import { CalcData } from "../../app.component";

@Component({

selector: 'inputs',

templateUrl: './inputs.component.html'

})

export class InputsComponent {

@Output() calcDataEvent = new EventEmitter<CalcData>();

@Output() resultEvent = new EventEmitter<string>();

inputNum1: string = '1';

inputNum2: string = '1';

pattern: RegExp = /[]/

operations = [

"Сложить",

"Вычесть",

"Поделить",

"Умножить"

];

inputSelectedValue = (this.operations)[0];

System: any = ['2', '8', '10', '16'];

param3\_val: any= this.System[0];

changeSystem(e: any): string {

console.log(e.target.value)

this.param3\_val = e.target.value

return this.param3\_val

}

keyPress(event: KeyboardEvent, pattern: any) {

const inputChar = String.fromCharCode(event.charCode);

if (!pattern.test(inputChar)) {

event.preventDefault();

}

}

getPatternParam() {

if (this.param3\_val == '2') {

this.pattern = /[0-1]/

}

if (this.param3\_val == '8') {

this.pattern = /[0-7]/

}

if (this.param3\_val == '16') {

this.pattern = /[0-9A-F]/

}

if (this.param3\_val == '10') {

this.pattern = /[0-9]/

}

return this.pattern

}

emitCalcData() {

this.inputNum2 = this.inputNum2;

const calcData: CalcData = {

num1: this["inputNum1"],

num2: this["inputNum2"],

selectedValue: this["inputSelectedValue"],

system: this["param3\_val"]

};

this.calcDataEvent.emit(calcData);

const num1Decimal = parseInt(this.inputNum1, parseInt(this.param3\_val, 10));

const num2Decimal = parseInt(this.inputNum2, parseInt(this.param3\_val, 10));

let resultDecimal: number;

switch (this.inputSelectedValue) {

case 'Сложить':

resultDecimal = num1Decimal + num2Decimal;

break;

case 'Вычесть':

resultDecimal = num1Decimal - num2Decimal;

break;

case 'Поделить':

resultDecimal = num1Decimal / num2Decimal;

break;

case 'Умножить':

resultDecimal = num1Decimal \* num2Decimal;

break;

default:

resultDecimal = 0;

break;

}

const resultInSelectedSystem = resultDecimal.toString(parseInt(this.param3\_val, 10));

this.resultEvent.emit(resultInSelectedSystem);

}

}

import { HttpClient, HttpParams } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';

@Injectable()

export class AppService {

private url: string;

constructor(private http: HttpClient) {

this.url = 'http://localhost:8080/calculations';

}

public addition(param1: string, param2: string, param3: string): Observable<any> {

return this.http.get(`${this.url + '/addition'}`, {

params: {

num1: param1,

num2: param2,

system: param3

}

})

}

public subtraction(param1: string, param2: string, param3: string): Observable<any> {

return this.http.get(`${this.url + '/subtraction'}`, {

params: {

num1: param1,

num2: param2,

system: param3

}

})

}

public division(param1: string, param2: string, param3: string): Observable<any> {

return this.http.get(`${this.url + '/division'}`, {

params: {

num1: param1,

num2: param2,

system: param3

}

})

}

public multiplication(param1: string, param2: string, param3: string): Observable<any> {

return this.http.get(`${this.url + '/multiplication'}`, {

params: {

num1: param1,

num2: param2,

system: param3

}

})

}

}

import {Directive, ElementRef, Input, OnChanges, SimpleChanges} from '@angular/core';

@Directive({

selector: '[redblackgreen]'

})

export class RedblackgreenDirective implements OnChanges {

// @ts-ignore

@Input() public redblackgreen: any;

@Input() public input: any;

constructor(private el: ElementRef) {

this.highlight('red');

}

ngOnChanges(changes: SimpleChanges) {

const inputValue = parseFloat(this.input);

if (isNaN(inputValue)) {

this.highlight('yellow');

this.el.nativeElement.value = 'NaN';

} else if (inputValue < 0) {

this.highlight('red');

} else if (inputValue === 0) {

this.highlight('black');

} else if (inputValue > 0) {

this.highlight('green');

}

/\*if (inputValue) {

this.el.nativeElement.value = this.input.toUpperCase();

}\*/

}

private highlight(color: string) {

this.el.nativeElement.style.backgroundColor = color;

}

}

import {Pipe, PipeTransform} from '@angular/core';

@Pipe({name: 'rounding'})

export class RoundingPipe implements PipeTransform {

transform(value: number, decPlacesNum = 3): number {

return Number(value.toFixed(decPlacesNum));

}

}

ПРИЛОЖЕНИЕ Б  
Cypress тесты

describe('Запуск приложения', () => {

beforeEach(() => {

cy.visit('http://localhost:4200/');

});

it('Существует первое окно ввода', () => {

cy.get('#input1').should('be.visible');

});

it('Существует второе окно ввода', () => {

cy.get('#input2').should('be.visible');

});

it('Существует выпадающий список систем счисления', () => {

cy.get('#system').should('be.visible');

});

it('Существует выпадающий список операций', () => {

cy.get('#operationSelector').should('be.visible');

});

it('Существует кнопка расчёта', () => {

cy.get('#calcButton').should('be.visible');

});

it('Корректный ответ сложения в 10 системе 41 + 19 = 60', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('41');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('19');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '60');

});

it('Корректный ответ вычитания в 10 системе 92 - 93 = -1', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('92');

cy.get('#operationSelector').select('Вычесть');

cy.get('#input2').clear().type('93');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '-1');

});

it('Корректный ответ умножения в 10 системе 52 \* 54 = 2808', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('52');

cy.get('#operationSelector').select('Умножить');

cy.get('#input2').clear().type('54');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '2808');

});

it('Корректный ответ деления в 10 системе 74 : 2 = 37', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('74');

cy.get('#operationSelector').select('Поделить');

cy.get('#input2').clear().type('2');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '37');

});

it('Корректный ответ сложения в 2 системе 111 + 10 = 1001', () => {

cy.get('#system').select('2');

cy.get('#input1').clear().type('111');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('10');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '1001');

});

it('Корректный ответ вычитания в 2 системе 100 - 1 = 11', () => {

cy.get('#system').select('2');

cy.get('#input1').clear().type('100');

cy.get('#operationSelector').select('Вычесть');

cy.get('#input2').clear().type('1');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '11');

});

it('Корректный ответ умножения в 2 системе 111 \* 111 = 110001', () => {

cy.get('#system').select('2');

cy.get('#input1').clear().type('111');

cy.get('#operationSelector').select('Умножить');

cy.get('#input2').clear().type('111');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '110001');

});

it('Корректный ответ деления в 2 системе 100 : 10 = 10', () => {

cy.get('#system').select('2');

cy.get('#input1').clear().type('100');

cy.get('#operationSelector').select('Поделить');

cy.get('#input2').clear().type('10');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '10');

});

it('Корректный ответ сложения в 8 системе 7 + 7 = 16', () => {

cy.get('#system').select('8');

cy.get('#input1').clear().type('7');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '16');

});

it('Корректный ответ вычитания в 8 системе 77 - 707 = -610', () => {

cy.get('#system').select('8');

cy.get('#input1').clear().type('77');

cy.get('#operationSelector').select('Вычесть');

cy.get('#input2').clear().type('707');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '-610');

});

it('Корректный ответ умножения в 8 системе 77 \* 7 = 671', () => {

cy.get('#system').select('8');

cy.get('#input1').clear().type('77');

cy.get('#operationSelector').select('Умножить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '671');

});

it('Корректный ответ деления в 8 системе 63 : 7 = 1.35', () => {

cy.get('#system').select('8');

cy.get('#input1').clear().type('63');

cy.get('#operationSelector').select('Поделить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '7.22222222222222222');

});

it('Корректный ответ сложения в 16 системе 12A + 7 = 131', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('12A');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '131');

});

it('Корректный ответ вычитания в 16 системе 12A - 7 = 123', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('12A');

cy.get('#operationSelector').select('Вычесть');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '123');

});

it('Корректный ответ умножения в 16 системе 12A \* 76 = 826', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('12A');

cy.get('#operationSelector').select('Умножить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '826');

});

it('Корректный ответ деления в 16 системе 12A : 7 = 2', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('12A');

cy.get('#operationSelector').select('Поделить');

cy.get('#input2').clear().type('7');

cy.get('#calcButton').click();

cy.get('#input3').should('have.value', '2A.924924924924');

});

it('Пустая строка при вводе букв и знаков в ввод 1', () => {

cy.get('#input1').clear().type('AF+-/\*ййы');

cy.get('#input1').should('have.value', '');

});

it('Пустая строка при вводе букв в ввод 2', () => {

cy.get('#input2').clear().type('AF+-/\*ййы');

cy.get('#input2').should('have.value', '');

});

it('Проверка в 2 системе, что можно ввести только 1 и 0 в ввод 1', () => {

cy.get('#system').select('2');

cy.get('#input1').clear().type('1023FA+-/\*');

cy.get('#input1').should('have.value', '10');

});

it('Проверка в 2 системе, что можно ввести только 1 и 0 в ввод 2', () => {

cy.get('#system').select('2');

cy.get('#input2').clear().type('1023FA+-/\*');

cy.get('#input2').should('have.value', '10');

});

it('Проверка в 8 системе, что можно ввести только числа от 0 до 7 в ввод 1', () => {

cy.get('#system').select('8');

cy.get('#input1').clear().type('1023456789FA+-/\*');

cy.get('#input1').should('have.value', '10234567');

});

it('Проверка в 8 системе, что можно ввести только числа от 0 до 7 в ввод 2', () => {

cy.get('#system').select('8');

cy.get('#input2').clear().type('1023456789FA+-/\*');

cy.get('#input2').should('have.value', '10234567');

});

it('Проверка в 16 системе, что можно ввести только числа от 0 до 9 и буквы A, B, C, D, E, F в ввод 1', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('1023456789ABCDEFGT+-/\*');

cy.get('#input1').should('have.value', '1023456789ABCDEF');

});

it('Проверка в 16 системе, что можно ввести только числа от 0 до 9 и буквы A, B, C, D, E, F в ввод 2', () => {

cy.get('#system').select('16');

cy.get('#input1').clear().type('1023456789ABCDEFGT+-/\*');

cy.get('#input1').should('have.value', '1023456789ABCDEF');

});

it('Невозможно ввести 0 во второй ввод при операции деления', () => {

cy.get('#system').select('10');

cy.get('#operationSelector').select('Поделить');

cy.get('#input2').clear().type('0');

cy.get('#input2').should('have.value', '');

});

it('Зелёный фон при ответе положительном числе (1 + 1 = 1)', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('1');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('1');

cy.get('#calcButton').click();

cy.get('#input3').should('have.css', 'background-color', 'rgb(0, 128, 0)');

});

it('Чёрный фон при ответе ноль (0 + 0 = 0)', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('0');

cy.get('#operationSelector').select('Сложить');

cy.get('#input2').clear().type('0');

cy.get('#calcButton').click();

cy.get('#input3').should('have.css', 'background-color', 'rgb(0, 0, 0)');

});

it('Красный фон при отрицательном ответе (1 - 5 = -4)', () => {

cy.get('#system').select('10');

cy.get('#input1').clear().type('1');

cy.get('#operationSelector').select('Вычесть');

cy.get('#input2').clear().type('5');

cy.get('#calcButton').click();

cy.get('#input3').should('have.css', 'background-color', 'rgb(255, 0, 0)');

});

})