**Отчёт по лабораторной работе №3**

**Сниппеты автотестов:**

**Text Input:**

var input = document.getElementById('newButtonName');

input.focus();

for (var ch of 'Hello World!') {

input.dispatchEvent(new KeyboardEvent('keydown', {key: ch}));

input.value += ch;

input.dispatchEvent(new InputEvent('input', {data: ch, inputType: 'insertText'}));

input.dispatchEvent(new KeyboardEvent('keyup', {key: ch}));

}

input.dispatchEvent(new Event('change', {bubbles: true}));

document.getElementById('updatingButton').click();

**Scrollbars:**

var div = document.querySelector('div[style="height:150px;overflow-y: scroll;width:300px;overflow-x:scroll"]');

div.scrollTop = div.scrollHeight;

div.scrollLeft = div.scrollWidth;

setTimeout(() => {

var btn = document.getElementById('hidingButton');

btn.click();

}, 300);

**Dynamic Table:**

var table = document.querySelector('div[role="table"]');

var header = table.querySelector('[role="rowgroup"]');

var cpuIndex = [...header.querySelectorAll('[role="columnheader"], [role="cell"]')]

.findIndex(c => c.textContent.trim() === 'CPU');

var rows = table.querySelectorAll('div[role="rowgroup"] div[role="row"]');

var chromeRow = [...rows].find(r => r.querySelector('[role="cell"]')?.textContent.trim() === 'Chrome');

var cpuValue = chromeRow?.querySelectorAll('[role="cell"]')[cpuIndex]?.textContent.trim();

var pValue = document.querySelector('p.bg-warning')?.textContent.match(/[\d.]+%/)?.[0];

console.log('Из таблицы:', cpuValue, '| Из <p>:', pValue, '| Совпадает:', cpuValue === pValue);

**Progress Bar:**

document.getElementById('startButton').click();

var progress = document.querySelector('div.progress');

var observer = new MutationObserver(() => {

var value = progress.textContent.trim();

if (value === '75%') {

document.getElementById('stopButton').click();

observer.disconnect();

}

});

observer.observe(progress, { childList: true, subtree: true, characterData: true });

**Visibility:**

var table = document.querySelector('table');

var buttons = table.querySelectorAll('button');

var hideButton = document.getElementById('hideButton');

hideButton.click();

buttons.forEach(btn => {

if (btn === hideButton) return;

var r = btn.getBoundingClientRect();

var style = getComputedStyle(btn);

let visible =

document.body.contains(btn) &&

style.display !== 'none' &&

style.visibility !== 'hidden' &&

style.opacity !== '0' &&

r.width > 0 && r.height > 0 &&

r.bottom > 0 && r.right > 0 &&

r.top < window.innerHeight && r.left < window.innerWidth;

if (visible) {

var cx = r.left + r.width / 2;

var cy = r.top + r.height / 2;

var el = document.elementFromPoint(cx, cy);

visible = el === btn || btn.contains(el);

}

console.log(btn.id || btn.textContent.trim(), visible ? 'видна' : 'скрыта');

});

**Animated Button:**

document.getElementById('animationButton').click();

var target = document.getElementById('movingTarget');

var observer = new MutationObserver(() => {

if (!target.classList.contains('spin')) {

observer.disconnect();

target.click();

}

});

observer.observe(target, { attributes: true, attributeFilter: ['class'] });

**Disabled Input:**

document.getElementById('enableButton').click();

var input = document.getElementById('inputField');

var observer = new MutationObserver(() => {

if (!input.disabled) {

observer.disconnect();

input.focus();

input.value = 'Hello world';

input.dispatchEvent(new Event('input', { bubbles: true }));

}

});

observer.observe(input, { attributes: true, attributeFilter: ['disabled'] });