Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №4

по дисциплине «Базовые информационные технологии»

Тема: «Обработка действий пользователя в браузере. Асинхронное программирование. JavaScript анимация»

Выполнил:

студент гр. ПРИ-123

А.Ц. Нямаа

Принял:

Ст. пр. кафедры ИСПИ

О.Н. Шамышев

Владимир, 2025 г.

ЦЕЛЬ РАБОТЫ:

Научиться создавать веб-страницы, обрабатывающие движение мыши и нажатие кнопок на клавиатуре, решать задачи на асинхронное программирование, промисификацию функций, написание циклов и условий с использованием функций с callback и Promise и создавать анимации слайдера с помощью JavaScript.

ЗАДАНИЕ 1 (вариант 12).

Создать веб-страницу и добавить для нее обработчик событий для мыши (минимум 3 события), для клавиатуры (минимум 1), кнопки и формы. Так же предотвратите копирование и выделение текста.

ВЫПОЛНЕНИЕ РАБОТЫ:

Для выполнения данного задания был изменен веб-сайт с предыдущей лабораторной работы: добавлено необходимое количество событий для мыши (в т.ч. *onclick* и *onmousedown*), для клавиатуры (на сочетание *ctrl + b* открывается меню покупки). Кнопки и формы были реализованы ранее. Также было предотвращено копирование текста с «титульника» сайта и выделение текста с блока заказов.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header onmousedown="return false">

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain" onmousedown="return false">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="textInMain" style="margin-top: 2%;" onmousedown="return false">

            <p>ЗАКАЗЫ</p>

        </div>

        <div class="orderContainer" oncopy="console.log('Копировать нельзя'); return false">

            <div class="orderHeader">

                <p>Имя</p>

                <p>Фамилия</p>

                <p>Почта</p>

                <p>Адрес</p>

                <p>Количество</p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

//Генерация html

const productListContainer = document.querySelector(".productsList");

const orderListContainer = document.querySelector(".orderContainer");

productList.forEach((product) => {

    const productUnit = document.createElement("div");

    productUnit.classList.add("productUnit");

    const imgProduct = document.createElement("img");

    imgProduct.src = product.getImgPath;

    const nameProduct = document.createElement("p");

    nameProduct.textContent = product.getName;

    const typeProduct = document.createElement("p");

    typeProduct.textContent = "Тип продукта: " + product.getType;

    const priceProduct = document.createElement("p");

    priceProduct.textContent = `${product.getPrice}$`;

    product.whatIsIt();

    const buyButtonProduct = document.createElement("button");

    buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

    buyButtonProduct.textContent = "Купить!";

    buyButtonProduct.dataset.price = product.price;

    productUnit.appendChild(imgProduct);

    productUnit.appendChild(nameProduct);

    productUnit.appendChild(typeProduct);

    productUnit.appendChild(priceProduct);

    productUnit.appendChild(buyButtonProduct);

    productListContainer.appendChild(productUnit);

});

orderList.forEach((order) => {

    const orderUnit = document.createElement("div");

    orderUnit.classList.add("orderRow");

    const nameOrder = document.createElement("p");

    nameOrder.textContent = order.getName;

    const surnameOrder = document.createElement("p");

    surnameOrder.textContent = order.getSurname;

    const emailOrder = document.createElement("p");

    emailOrder.textContent = order.getEmail;

    const addressOrder = document.createElement("p");

    addressOrder.textContent = order.getAddress;

    const quantityOrder = document.createElement("p");

    quantityOrder.textContent = order.getQuantity;

    orderUnit.appendChild(nameOrder);

    orderUnit.appendChild(surnameOrder);

    orderUnit.appendChild(emailOrder);

    orderUnit.appendChild(addressOrder);

    orderUnit.appendChild(quantityOrder);

    orderListContainer.appendChild(orderUnit);

});

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

const openModels = document.querySelectorAll(".openModel");

function openModelScript() {

    dialog.showModal();

    html.classList.add("changehtml");

    priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

}

for (let i = 0; i < openModels.length; i++) {

    openModels[i].addEventListener("click", openModelScript);

}

document.querySelector("#closeModel").addEventListener("click", function () {

    dialog.close();

    html.classList.remove("changehtml");

});

addEventListener("keydown", function (event) {

    if (event.ctrlKey && event.key === "b") {

        openModelScript();

    } else if (event.ctrlKey && event.key === "q") {

        window.close();

    }

});

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

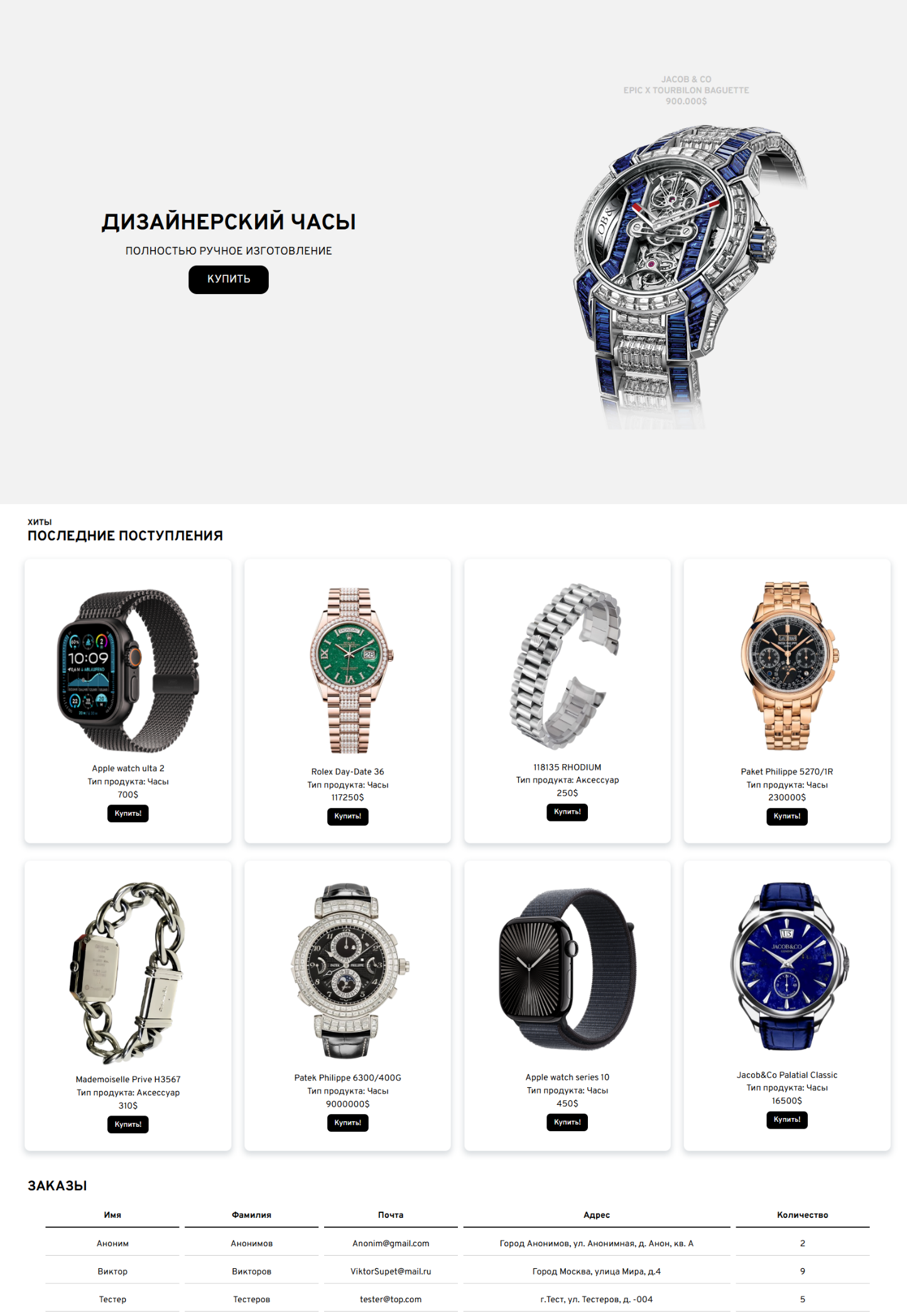
    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

  
Рисунок 1. Созданный веб-сайт.

ЗАДАНИЕ 2 (вариант 12).

Написать циклы и условия с использованием функций с callback и Promise, промисификацию функций и асинхронную функцию. Обработать исключения при использовании колбеков, промисов и асинхронных функций.

ВЫПОЛНЕНИЕ РАБОТЫ:

В ходе выполнения данного задания были написаны функции с использованием callback, Promise, промисификации и асинхронности. Также были обработаны все исключения.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

rel="stylesheet">

<title>Clock Shop</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<div class="leftText">

<h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

<p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

<button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

<dialog class="dialogMain">

<form class="orderForm">

<h2>Оформление заказа</h2>

<p class="hidden" id="validSuccess"></p>

<p class="hidden" id="validError"></p>

<p><label for="name">Имя:</label>

<input type="text" id="name" name="name" placeholder="Имя" required>

<p class="hidden text-danger" id="nameP"></p>

</p>

<p><label for="secondName">Фамилия:</label>

<input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

<p class="hidden text-danger" id="secondNameP"></p>

</p>

<p><label for="thirdName">Отчество:</label>

<input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

<p class="hidden text-danger" id="thirdNameP"></p>

</p>

<p><label for="address">Адрес доставки:</label>

<input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

</p>

<p><label for="quantity">Количество:</label>

<input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

</p>

<p><label for="promo">Промокод:</label>

<input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

</p>

<p class="priceHeader"></p>

<input type="submit" value="Заказать!">

</form>

<button id="closeModel" class="closeModel">&#x2715 </button>

</dialog>

</div>

<div class="rightText">

<p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

<img src="imgs/mainPhotoWatches.png">

</div>

</header>

<main>

<div class="textInMain">

<p>ХИТЫ</p>

<p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

</div>

<div class="productsList"> </div>

<div class="textInMain" style="margin-top: 2%;">

<p>ЗАКАЗЫ</p>

</div>

<div class="orderContainer">

<div class="orderHeader">

<p>Имя</p>

<p>Фамилия</p>

<p>Почта</p>

<p>Адрес</p>

<p>Количество</p>

</div>

</div>

  </main>

    <script src="script\_v2.js"></script>

</body>

</html>

Листинг script.js:

//Генерация html

const productListContainer = document.querySelector(".productsList");

const orderListContainer = document.querySelector(".orderContainer");

productList.forEach((product) => {

    let promise =  loadProducts(product);

    promise

        .then((result) => console.log(result))

        .catch((error) => console.log(error));

});

function loadProducts(product){

    return new Promise(function(resolve, reject){

        if (!product.getImgPath || !product.getName || !product.getType || !product.getPrice){

            reject(new Error(`Товар ${product.getId} не может быть добавлен!`));

            return;

        }

        const productUnit = document.createElement("div");

        productUnit.classList.add("productUnit");

        const imgProduct = document.createElement("img");

        imgProduct.src = product.getImgPath;

        const nameProduct = document.createElement("p");

        nameProduct.textContent = product.getName;

        const typeProduct = document.createElement("p");

        typeProduct.textContent = "Тип продукта: " + product.getType;

        const priceProduct = document.createElement("p");

        priceProduct.textContent = `${product.getPrice}$`;

        product.whatIsIt();

        const buyButtonProduct = document.createElement("button");

        buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

        buyButtonProduct.textContent = "Купить!";

        buyButtonProduct.dataset.price = product.price;

        buyButtonProduct.addEventListener("click", function () {

            dialog.showModal();

            html.classList.add("changehtml");

            priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

        });

        productUnit.appendChild(imgProduct);

        productUnit.appendChild(nameProduct);

        productUnit.appendChild(typeProduct);

        productUnit.appendChild(priceProduct);

        productUnit.appendChild(buyButtonProduct);

        productListContainer.appendChild(productUnit);

        resolve(`Товар ${product.getId} успешно добавлен`);

    });

};

function loadOrder(order, callback){

    if (!order.getName || !order.getSurname || !order.getEmail || !order.getAddress || !order.getQuantity){

        callback(new Error(`Ошибка! Заказ ${order.getId} не может быть загружен!`));

    }

    else{

        const orderUnit = document.createElement("div");

        orderUnit.classList.add("orderRow");

        const nameOrder = document.createElement("p");

        nameOrder.textContent = order.getName;

        const surnameOrder = document.createElement("p");

        surnameOrder.textContent = order.getSurname;

        const emailOrder = document.createElement("p");

        emailOrder.textContent = order.getEmail;

        const addressOrder = document.createElement("p");

        addressOrder.textContent = order.getAddress;

        const quantityOrder = document.createElement("p");

        quantityOrder.textContent = order.getQuantity;

        orderUnit.appendChild(nameOrder);

        orderUnit.appendChild(surnameOrder);

        orderUnit.appendChild(emailOrder);

        orderUnit.appendChild(addressOrder);

        orderUnit.appendChild(quantityOrder);

        orderListContainer.appendChild(orderUnit);

        callback(null, `Заказ ${order.getId} загружен успешно!`);

    };

};

let loadOrderPromise = function(order){

    return new Promise((resolve, reject) => {

        loadOrder(order, (error, payload) => {

            if (error) reject (error)

            else resolve(payload);

        });

    });

};

orderList.forEach((order) => {

    loadOrderPromise(order)

        .then(payload => console.log(payload))

        .catch(error => console.log(error));

});

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

};

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

const openModels = document.querySelectorAll(".openModel");

for (let i = 0; i < openModels.length; i++) {

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

};

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

};

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

};

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    let hasError = false;

    const toCheck = [

        {param1: name, param2: "nameP"},

        {param1: surname, param2: "secondNameP"},

        {param1: thirdName, param2: "thirdNameP"}

    ]

    toCheck.forEach((item) => {

        checker(item.param1, (error) => {

            const element = document.getElementById(item.param2);

            if (error){

                element.textContent = error.message;

                element.classList.remove("hidden");

                hasError = true;

            } else{

                element.textContent = "";

                element.classList.add("hidden");

            }

        });

    });

    if (hasError) {

        document.getElementById("validError").textContent = "Ошибка! Некоторые данные были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    };

});

function checker(str, callback) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        callback(new Error("Не должно быть цифр!"));

        return;

    }

    if (str.length > 50 || str.length < 3) {

        callback(new Error("От 3 до 50 символов!"));

        return;

    }

    callback(null, "Покупка прошла успешно!");

};

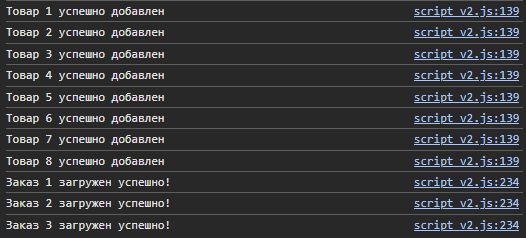


Рисунок 2. Все товары и покупки успешно загружены

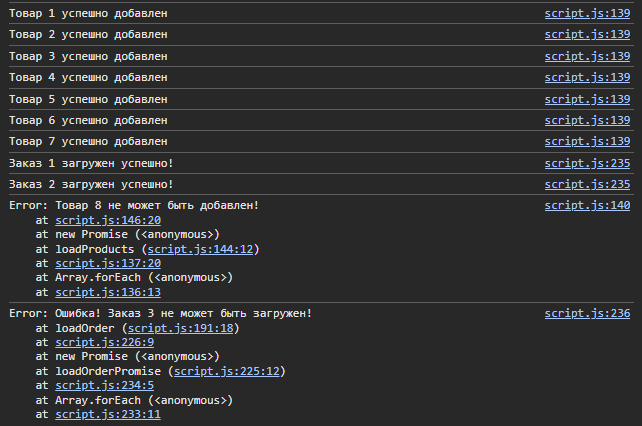


Рисунок 3. Произошла ошибка при загрузке товара и покупки.

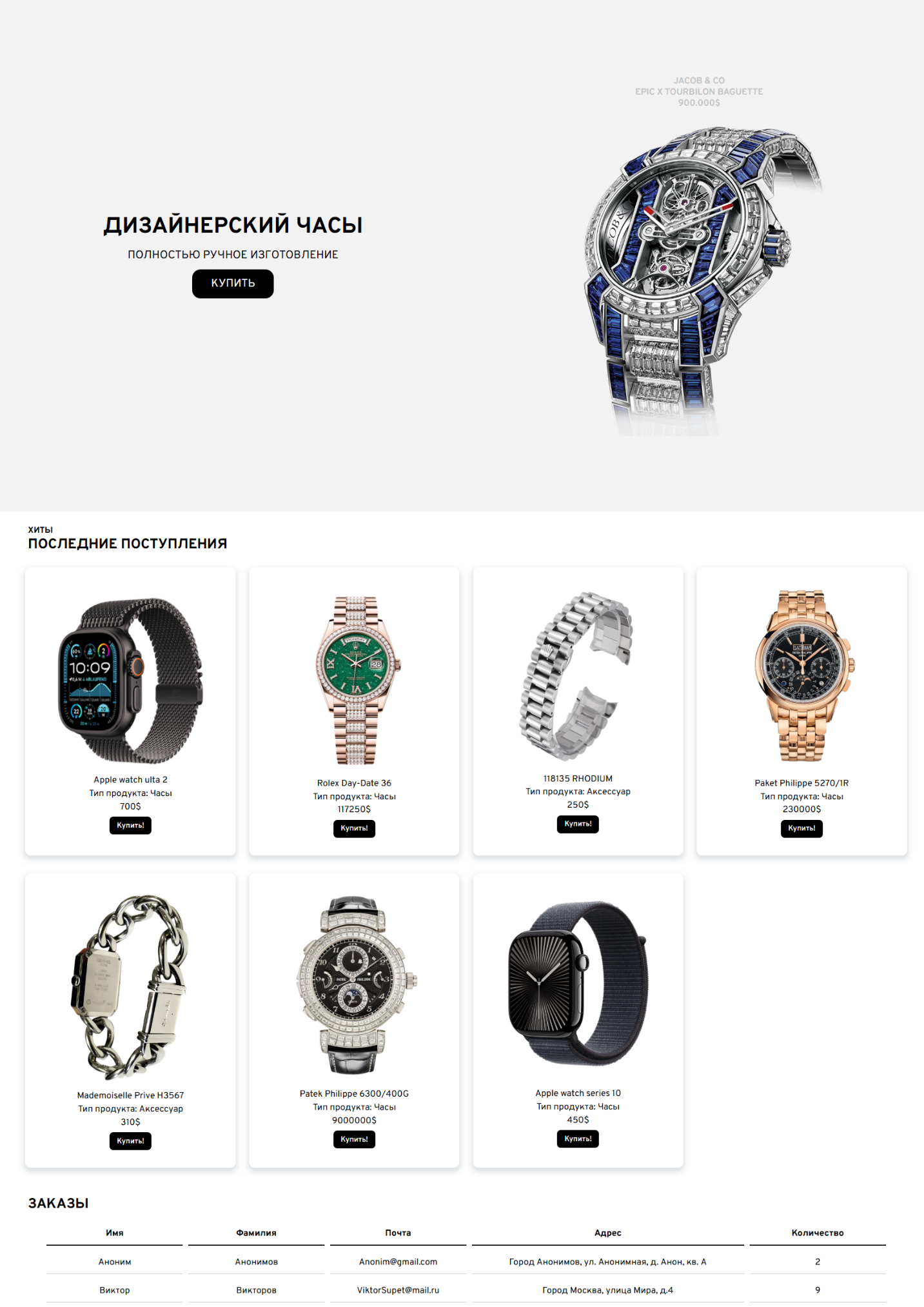


Рисунок 4. Разработанный веб-сайт с ошибкой загрузки.

Далее функция генерации товаров на сайт была изменена таким образом, что стала асинхронной.

Листинг script\_v2.js:

const productListContainer = document.querySelector(".productsList");

productList.forEach(async (product) => {

    try {

        const payloadProduct = await loadProducts(product);

        console.log(payloadProduct);

    } catch (error){

        console.log(error);

    }

});

async function loadProducts(product){

    if (!product.getImgPath || !product.getName || !product.getType || !product.getPrice){

        throw new Error(`Товар ${product.getId} не может быть добавлен!`);

    }

    const productUnit = document.createElement("div");

    productUnit.classList.add("productUnit");

    const imgProduct = document.createElement("img");

    imgProduct.src = product.getImgPath;

    const nameProduct = document.createElement("p");

    nameProduct.textContent = product.getName;

    const typeProduct = document.createElement("p");

    typeProduct.textContent = "Тип продукта: " + product.getType;

    const priceProduct = document.createElement("p");

    priceProduct.textContent = `${product.getPrice}$`;

    product.whatIsIt();

    const buyButtonProduct = document.createElement("button");

    buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

    buyButtonProduct.textContent = "Купить!";

    buyButtonProduct.dataset.price = product.price;

    buyButtonProduct.addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

    });

    productUnit.appendChild(imgProduct);

    productUnit.appendChild(nameProduct);

    productUnit.appendChild(typeProduct);

    productUnit.appendChild(priceProduct);

    productUnit.appendChild(buyButtonProduct);

    productListContainer.appendChild(productUnit);

    return `Товар ${product.getId} успешно добавлен`;

};

ЗАДАНИЕ 3 (вариант 12).

Дописать скрипт для слайдера, чтобы можно было листать слайды по стрелкам. Написать еще один слайдер на JavaScript (любая другая реализация).

ВЫПОЛНЕНИЕ РАБОТЫ:

В начала выполнения был доделан отвечающий за слайдер скрипт из лабораторной работы: появились кнопки, которые позволяют листать слайды. Также был написан еще один слайдер с другой логикой реализации.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header onmousedown="return false">

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="slider" id="slider">

            <div class="slides" id="slides">

                <div class="slide"><img src="imgs/sliderfirst.jpg"></div>

                <div class="slide"><img src="imgs/slidersecond.jpg"></div>

                <div class="slide"><img src="imgs/sliderthird.jpg"></div>

            </div>

            <div class="sliderButtons">

                <button class="prev" id="prev">🠘</button>

                <button class="next" id="next">🠚</button>

            </div>

        </div>

        <div class="slider\_v2" id="slider\_v2">

            <div class="slide\_v2"><img src="imgs/sliderfirst.jpg"></div>

            <div class="slide\_v2"><img src="imgs/slidersecond.jpg"></div>

            <div class="slide\_v2"><img src="imgs/sliderthird.jpg"></div>

            <button class="prev\_v2" id="prev\_v2">🠘</button>

            <button class="next\_v2" id="next\_v2">🠚</button>

        </div>

        <div class="textInMain" onmousedown="return false">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="textInMain" style="margin-top: 2%;" onmousedown="return false">

            <p>ЗАКАЗЫ</p>

        </div>

        <div class="orderContainer" oncopy="console.log('Копировать нельзя'); return false">

            <div class="orderHeader">

                <p>Имя</p>

                <p>Фамилия</p>

                <p>Почта</p>

                <p>Адрес</p>

                <p>Количество</p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Слайдер1

const slider = document.getElementById("slider");

const slides = document.getElementById("slides");

const slideWidth = document.querySelector(".slide").offsetWidth;

const prevButton = document.getElementById("prev");

const nextButton = document.getElementById("next");

prevButton.addEventListener("click", slidePrev);

nextButton.addEventListener("click", slideNext);

let counter = 0;

function slideNext() {

    counter++;

    if (counter >= slides.children.length) {

        counter = 0;

    }

    slides.style.transform = `translateX(${-counter \* slideWidth}px)`;

}

function slidePrev() {

    counter--;

    if (counter < 0) {

        counter = slides.children.length - 1;

    }

    slides.style.transform = `translateX(${-counter \* slideWidth}px)`;

}

setInterval(slideNext, 3000);

// Слайдер2

const imagesSlider2 = document.querySelectorAll('.slider\_v2 img');

const prevSlider2 = document.getElementById('prev\_v2');

const nextSlider2 = document.getElementById('next\_v2');

let currentIndex = 0;

function showImage(index) {

    imagesSlider2[currentIndex].classList.remove('active');

    imagesSlider2[index].classList.add('active');

    currentIndex = index;

}

prevSlider2.addEventListener('click', function(){

    let index = currentIndex - 1;

    if (index < 0) {

        index = imagesSlider2.length - 1;

    }

    showImage(index);

});

nextSlider2.addEventListener('click', function(){

    let index = currentIndex + 1;

    if (index >= imagesSlider2.length) {

        index = 0;

    }

    showImage(index);

});

showImage(currentIndex);



Рисунок 5. Дописанный слайдер из методического указания.



Рисунок 6. Созданный слайдер с другой логикой реализации.

ВЫВОД К РАБОТЕ:

Я научился создавать веб-страницы, обрабатывающие движение мыши и нажатие кнопок на клавиатуре, решать задачи на асинхронное программирование, промисификацию функций, написание циклов и условий с использованием функций с callback и Promise и создавать анимации слайдера с помощью JavaScript.