**Лабораторна робота №10**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

*ДУ «Житомирська політехніка».23.121.03.00 – Лр10*

Розроб.

Білотіл Я. О.

Перевір.

Дмитренко І. А.

*Т. контр.*

Н. контр.

Затверд.

Звіт

з лабораторної роботи

Літ.

Аркушів

*11*

*ФІКТ Гр. IПЗк-23-1*

**Тема: LocalStorage. JSON. Наслідування класу. Статичні властивості та методи**

**Мета роботи:** вивчити засоби по роботі з cookie, localStorage, JSON, освоїти роботу з класами в JavaScript.

**Завдання на лабораторну роботу**

**Завдання 0:** Ретельно вивчити теорію за посиланнями:

[Методи JSON, toJSON](https://uk.javascript.info/json)

[LocalStorage, sessionStorage](https://uk.javascript.info/localstorage)

[Базовий синтаксис класу](https://uk.javascript.info/class)

[Наслідування класу](https://uk.javascript.info/class-inheritance)

[Статичні властивості та методи](https://uk.javascript.info/static-properties-methods)

**Завдання 1. LOCALSTORAGE**

Дана форма з **input**, **textarea**, **checkbox**, **radio**, **select** і тп. Користувач вводить якісь дані і закриває сторінку (не факт, що він заповнив всю форму). Зробіть так, щоб при наступному заході на сторінку введені ним раніше дані стояли на своїх місцях.

Код програми:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Local Storage Form</title>

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

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

</head>

<body>

<form id="myForm">

<label for="inputField">Input:</label>

<input type="text" id="inputField"><br>

<label for="textareaField">Textarea:</label>

<textarea id="textareaField"></textarea><br>

<label for="checkboxField">Checkbox:</label>

<input type="checkbox" id="checkboxField"><br>

<label for="radioField1">Radio 1:</label>

<input type="radio" id="radioField1" name="radioGroup"><br>

<label for="radioField2">Radio 2:</label>

<input type="radio" id="radioField2" name="radioGroup"><br>

<label for="selectField">Select:</label>

<select id="selectField">

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

</form>

</body>

</html>

style.css

label {

display: block;

margin-bottom: 5px;

}

script.js

const inputElement = document.getElementById('inputField');

const textareaElement = document.getElementById('textareaField');

const checkboxElement = document.getElementById('checkboxField');

const radio1Element = document.getElementById('radioField1');

const radio2Element = document.getElementById('radioField2');

const selectElement = document.getElementById('selectField');

function saveToLocalStorage() {

localStorage.setItem('inputValue', inputElement.value);

localStorage.setItem('textareaValue', textareaElement.value);

localStorage.setItem('checkboxValue', checkboxElement.checked);

localStorage.setItem('radioValue1', radio1Element.checked);

localStorage.setItem('radioValue2', radio2Element.checked);

localStorage.setItem('selectValue', selectElement.value);

}

function loadFromLocalStorage() {

inputElement.value = localStorage.getItem('inputValue') || '';

textareaElement.value = localStorage.getItem('textareaValue') || '';

checkboxElement.checked = localStorage.getItem('checkboxValue') === 'true';

radio1Element.checked = localStorage.getItem('radioValue1') === 'true';

radio2Element.checked = localStorage.getItem('radioValue2') === 'true';

selectElement.value = localStorage.getItem('selectValue') || '';

}

inputElement.addEventListener('input', saveToLocalStorage);

textareaElement.addEventListener('input', saveToLocalStorage);

checkboxElement.addEventListener('change', saveToLocalStorage);

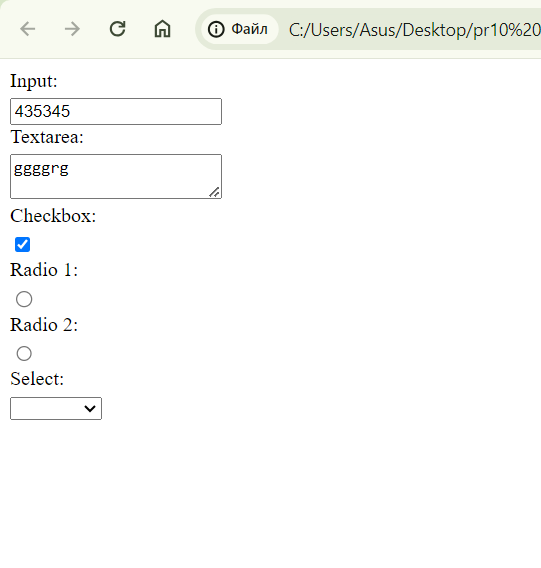
radio1Element.addEventListener('change', saveToLocalStorage);

radio2Element.addEventListener('change', saveToLocalStorage);

selectElement.addEventListener('change', saveToLocalStorage);

window.addEventListener('load', loadFromLocalStorage);

Результат роботи програми:



Мал. 1 – Завдання 1

**Завдання 2. JSON**

Створіть на сторінці текстовий input, який буде приймати імена файлів як масив в форматі JSON (наприклад, [" file1.jpg "," file2.gif "," file34.gif "] )

* По натисканню на кнопку повинні з'явитися зображення в маленькому розмірі, по натисканню на які відкриватися ці зображення в реальному розмірі
* Якщо в текстовому полі введені дані не відповідають правильному формату JSON, необхідно повідомити про це користувача

Код програми:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Image Gallery</title>

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

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

</head>

<body>

<input type="text" id="fileInput" placeholder="Введіть масив файлів у форматі JSON">

<button onclick="showImages()">Показати зображення</button>

<div id="imageGallery"></div>

<div class="lightbox" id="lightbox">

<img id="fullImage" src="" alt="Full Size Image">

</div>

</body>

</html>

style.css

.thumbnail {

width: 100px;

height: 100px;

margin: 5px;

cursor: pointer;

}

.lightbox {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.8);

justify-content: center;

align-items: center;

}

.lightbox img {

max-width: 90%;

max-height: 90%;

}

script.js

function showImages() {

const fileInput = document.getElementById('fileInput').value.trim();

let files;

try {

files = JSON.parse(fileInput);

if (!Array.isArray(files)) {

throw new Error('Введіть масив файлів у форматі JSON');

}

const gallery = document.getElementById('imageGallery');

gallery.innerHTML = '';

files.forEach(file => {

const img = document.createElement('img');

img.src = file.trim();

img.classList.add('thumbnail');

img.addEventListener('click', () => {

const lightbox = document.getElementById('lightbox');

const fullImg = document.getElementById('fullImage');

fullImg.src = file.trim();

lightbox.style.display = 'flex';

});

gallery.appendChild(img);

});

} catch (error) {

alert(error.message);

}

}

window.addEventListener('click', (event) => {

const lightbox = document.getElementById('lightbox');

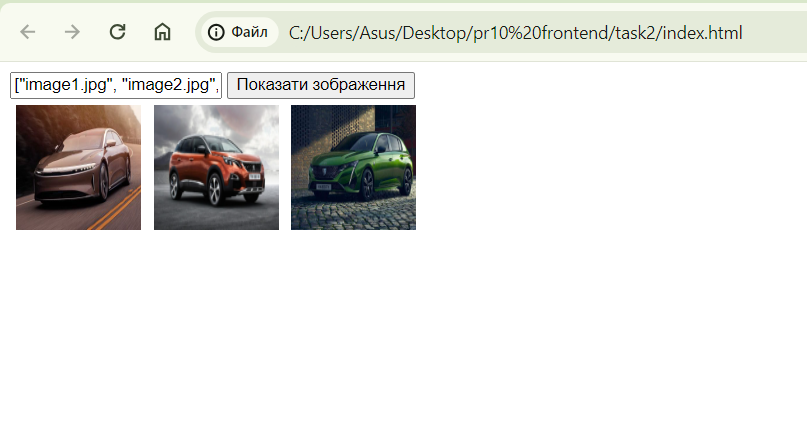
if (event.target === lightbox) {

lightbox.style.display = 'none';

}

});

Результат роботи програми:



Мал. 2 – Завдання 2

**Завдання 3. КЛАСИ. НАСЛІДУВАННЯ КЛАСУ**

1. Реалізувати клас **Student**, що описує студента, та створити масив студентів (прізвище, ім'я, оцінки з предметів)
2. Реалізувати клас **ListOfStudents** для генерації html-коду таблиці зі списком студентів:

* масив студентів необхідно передавати через конструктор
* таблицю отримувати з допомогою методу **getTableList()**

Створити об'єкт класу **ListOfStudents** і вивести результат роботи методу **getTableList()**.

1. Реалізувати клас **StylesTable**, який успадковується від класу **ListOfStudents**

* додати метод **getStyles()**, який повертає рядок зі стилями таблиці в тегах style.
* перевизначити метод **getTableList(),** який додає стилі до того що повертає метод **getTableList()** з батьківського класу.
* реалізувати метод **getAvg()**, що підраховує середній бал кожного студента і додає його до кінця рядка кожному студенту.
* реалізувати метод **getTotalAvg(),** що підраховує загальний середній бал групи

Створити об'єкт класу **StylesTable** і вивести результат роботи методу **getTableList(), getAvg(), getTotalAvg().**

Код програми:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Student Table</title>

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

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

</head>

<body>

<div id="output"></div>

</body>

</html>

style.css

table {

border-collapse: collapse;

width: 100%;

}

th,

td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

script.js

class Student {

constructor(lastName, firstName, grades) {

this.lastName = lastName;

this.firstName = firstName;

this.grades = grades;

this.averageGrade = this.calculateAverageGrade();

}

calculateAverageGrade() {

const sum = this.grades.reduce((acc, grade) => acc + grade, 0);

return sum / this.grades.length;

}

}

class ListOfStudents {

constructor(students) {

this.students = students;

this.groupAverageGrade = this.calculateGroupAverageGrade();

}

calculateGroupAverageGrade() {

const allGrades = this.students.flatMap(student => student.grades);

const sum = allGrades.reduce((acc, grade) => acc + grade, 0);

return sum / allGrades.length;

}

getTableList() {

let table = '<table><thead><tr><th>Прізвище</th><th>Ім\'я</th><th>Оцінки</th><th>Середній бал</th></tr></thead><tbody>';

this.students.forEach(student => {

table += `<tr><td>${student.lastName}</td><td>${student.firstName}</td><td>${student.grades.join(', ')}</td><td>${student.averageGrade.toFixed(2)}</td></tr>`;

});

table += `</tbody></table><p>Середній бал по групі: ${this.groupAverageGrade.toFixed(2)}</p>`;

return table;

}

}

const students = [

new Student('Петров', 'Петро', [80, 75, 90]),

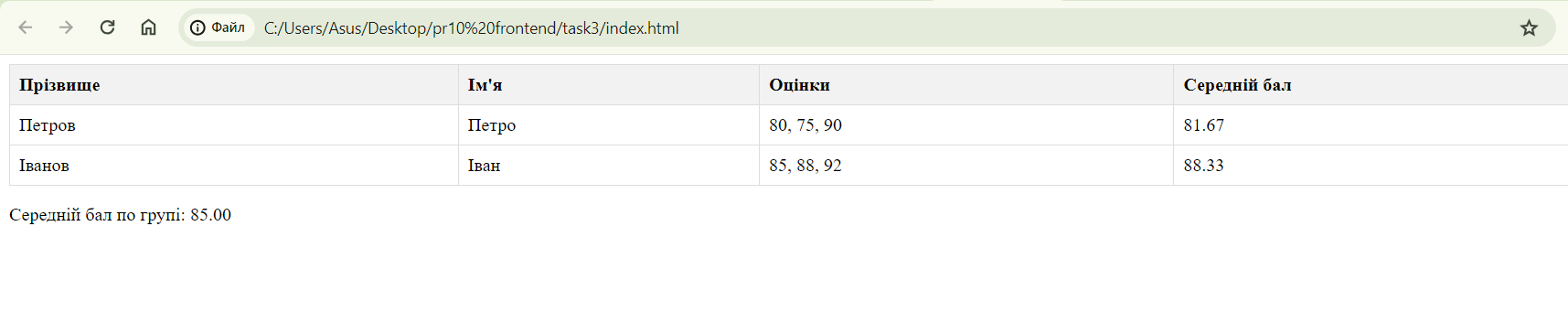
new Student('Іванов', 'Іван', [85, 88, 92]),

];

const studentList = new ListOfStudents(students);

document.getElementById('output').innerHTML = studentList.getTableList();

Результат роботи програми:



Мал. 3 – Завдання 3

**Завдання 4. CТАТИЧНІ МЕТОДИ ТА ВЛАСТИВОСТІ**

Створіть клас **Shape**, який має 2 властивості та 2 методи:

* статична властивість **total** – кількість фарби
* а - радіус кола або довжина сторони квадрата)
* статичний метод **fill()**, який заправлятиме фарбу до 100%
* метод **draw()**, який малюватиме фігуру

При створенні об'єкта (кнопка **draw**) малюється фігура певного розміру з кількістю кольору, що дорівнює змінній **total**. Кожна нова фігура витрачає певну кількість фарби і настане момент, коли фарби не залишиться. При натисканні кнопки **fill** кількість фарби поповнюється на 100%. Продемонструвати роботу методів.

Код програми:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Shape Drawing</title>

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

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

</head>

<body>

<h1>Shape Drawing</h1>

<label for="size">Розмір фігури:</label>

<input type="number" id="size" min="1" value="1">

<button onclick="drawShape()">Намалювати</button>

<button onclick="fillPaint()">Заправити фарбу</button>

<p id="result"></p>

<div id="shape"></div>

</body>

</html>

style.css

.square {

width: 50px;

height: 50px;

display: inline-block;

margin-right: -4px;

margin-bottom: -4px;

}

script.js

class Shape {

static total = 100;

constructor(a) {

this.a = a;

this.drawn = 0;

}

static fill() {

Shape.total = 100;

document.getElementById('result').innerText = 'Фарбу заправлено до 100%';

document.getElementById('shape').innerHTML = ''; // Очистимо фігуру

}

draw() {

if (Shape.total <= 0) {

document.getElementById('result').innerText = 'Недостатньо фарби для малювання.';

return;

}

const squaresToDraw = Math.min(this.a - this.drawn, Shape.total);

for (let i = 0; i < squaresToDraw; i++) {

const shapeElement = document.createElement('div');

shapeElement.classList.add('square');

shapeElement.style.backgroundColor = 'blue';

document.getElementById('shape').appendChild(shapeElement);

}

Shape.total -= squaresToDraw;

this.drawn += squaresToDraw;

if (this.drawn < this.a) {

document.getElementById('result').innerText = `Малюємо ${this.drawn}/${this.a} квадратиків. Залишилося фарби: ${Shape.total}.`;

} else {

document.getElementById('result').innerText = `Фігура малюється повністю. Залишилося фарби: ${Shape.total}.`;

}

}

}

function drawShape() {

const size = parseInt(document.getElementById('size').value);

const shape = new Shape(size);

shape.draw();

}

function fillPaint() {

Shape.fill();

document.getElementById('result').innerText = 'Фарбу заправлено до 100%';

}

Результат роботи програми:



Мал. 4 – Завдання 4