Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра  автоматики та управління в технічних системах

**Лабораторна робота № 15**

по дисципліні «Основи клієнтської розробки»

Тема: JavaScript. Події. Спливання. Делегування подій

|  |  |
| --- | --- |
| Виконав:  студент групи ІА-94  Чумак Володимир Володимирович  Дата здачі \_\_\_\_  Захищено з балом \_\_\_ |  |

Київ 2020

**1. Тема**

JavaScript. Події. Спливання. Делегування подій

**2. Завдання**

Використовувати тему, обрану в 4 роботі.

Створити список або використати існуючий. Реалізувати підсвічування

елементів списку при кліку миші. Використати **event.target**. Обробник

**onclick** застоувати для списку, а не для кожного елементу.

Створити меню (кілька кнопок), додати один обробник для всього меню і

атрибути **data-\*** для кожної кнопки, в відповідності з методами, які вони

викликають.

Застосувати прийом проектування «Поведінка» (додавання елементам

поведінки **behavior** за допомогою атрибута **data-\***.

**3. Теоретичні відомості**

**Спливання**

Принцип спливання дуже простий.

Коли на елементі відбувається подія, обробники спочатку спрацьовують на

ньому, потім на його батьку, потім вище і так далі, вгору по ланцюжку предків.

Наприклад, є 3 вкладених елемента FORM > DIV > P з обробником на кожному.

Клік по внутрішньому <p> викличе обробник onclick:

1. Спочатку на самому <p>.

2. Потім на зовнішньому <div>.

3. Потім на зовнішньому <form>.

І так далі вгору по ланцюжку до самого document.

Тому якщо клікнути на <p>, то ми побачимо три оповіщення: p → div → form.

Цей процес називається «спливанням», тому що події «спливають» від внутрішнього елемента вгору через батьків.

Майже всі події спливають.

Наприклад, подія focus не спливає. Однак, варто розуміти, що це швидше

виняток, ніж правило, все-таки більшість подій спливають.

**event.target**

Завжди можна дізнатися, на якому конкретно елементі відбулася подія.

Найглибший елемент, який викликає подія, називається цільовим

елементом, і він доступний через event.target.

Відмінності від this (= event.currentTarget):

event.target - це «цільової» елемент, на якому відбулася подія, в процесі

спливання він незмінний.

this - це «поточний» елемент, до якого дійшло спливання, на ньому зараз

виконується обробник.

**Припинення спливання**

Спливання йде з «цільового» елемента прямо наверх. За замовчуванням

подія буде спливати до елемента <html>, а потім до об'єкта document, а іноді

навіть до window, викликаючи всі обробники на своєму шляху.

Але будь-який проміжний обробник може вирішити, що подія повністю

оброблено, і зупинити спливання.

Для цього потрібно викликати метод event.stopPropagation ().

**Занурення**

Існує ще одна фаза з життєвого циклу події - «занурення» (іноді її

називають «перехоплення»). Вона дуже рідко використовується в реальному

коді.

**Делегування подій**

Спливання і перехоплення подій дозволяють реалізувати один з

найважливіших прийомів розробки - делегування.

Ідея в тому, що якщо у нас є багато елементів, події на яких потрібно

обробляти схожим чином, то замість того, щоб призначати обробник кожному,

ми ставимо один обробник на їх загального предка.

З нього можна отримати цільовий елемент event.target, зрозуміти на якому

саме нащадку відбулася подія і обробити його.

**4. Порядок виконання лабораторної роботи**

1) ознайомитися з теоретичними відомостями;

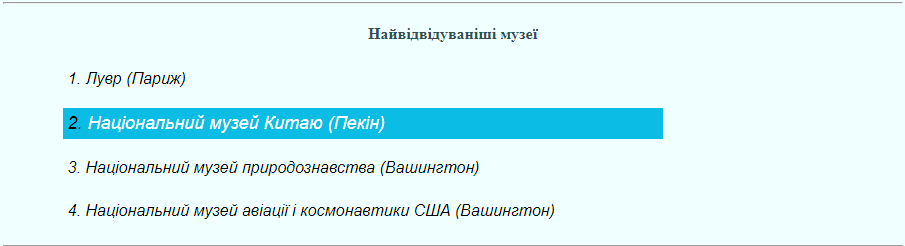
2) виконати завдання

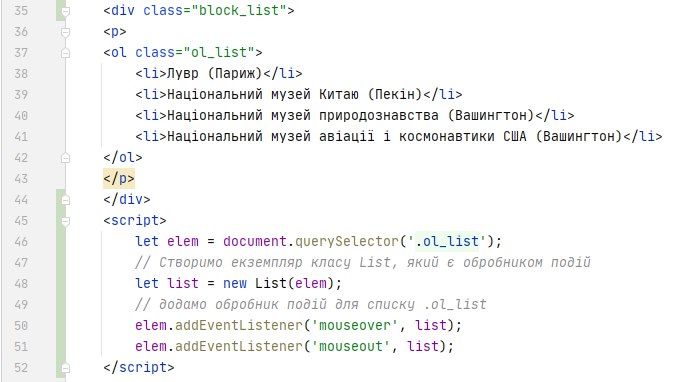
3) оформити звіт, який включатиме: титульний аркуш, завдання, теоретичні відомості, результати і висновки по роботі;

4) продемонструвати результат на комп'ютері і захистити лабораторну роботу.

**Результати виконання роботи**

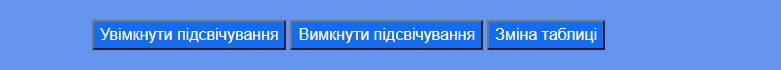
1. Список







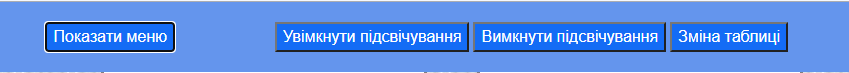
1. Меню







1. Поведінка







**Висновки**

У цій лабораторній роботі я реалізував підсвічування елементів існуючого списку за допомогою event.target, створив меню та додав до нього обробник, а також атрибути data-\* для кожної кнопки. Також я застосував прийом проектування «Поведінка» для додавання елементам поведінки behavior за допомогою атрибуту data-\*.