МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Циклова комісія програмування***

**Практична робота №3**

**Робочий план**

# з курсу «Front-end-програмування на React JS»

Воробйова Валентина Юріївна

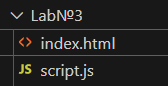
*ПІБ студента*

студента групи **1П-20**

|  |  |  |
| --- | --- | --- |
| **Вид роботи** | **Дата** | **Оцінка** |
| *Пром. контроль* |  |  |
| *Залік* |  |  |

Викладач   Сухенко А. С.

1. Створити папку з файлами index.html та script.js



1. Створіть рекурсивну функцію яка знаходить число фібоначі за переданим її номером. Формула знаходження числа фібоначчі:

F(n) = F(n-1) + F(n-2)

function Fibonaci(n) {

    if (n <= 1) {

        return n;

    } else {

        return fibonacci(n - 1) + fibonacci(n - 2);

    }

}

let number = 12

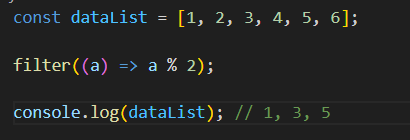
const fib\_number = Fibonaci(number);

console.log(`Число Фібоначчі з номером ${number}: ${fibNumber}`);

opera_EjYrEnJ26y

1. Створити чисту функцію filter яка приймає функцію handler і фільтрує масив в залежості від результату функції handler викликаної для кожного елемента масиву. Результатом роботи функції handler повинен бути масив відфільтрованих данних.

Приклад роботи ф-ції filter



\* Зверніть увагу, що данний приклад не задовольняє всі умови задачі. Що саме не задовольняє умові задачі треба визначити самостійно.

\*\* Для виконання цієї задачі ви не можете використовувати методи масиву які його ітерують, окрім методу forEach, cторонні бібліотеки також заборонені для використання.

function Filter(handler, arr) {

    const result = [];

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

        if (handler(arr[i])) {

            result.push(arr[i]);

        }

    }

    return result;

}

const datalist = [1, 2, 3, 4, 5, 6];

const filteredList = filter((a) => a % 2, datalist);

console.log(filteredList);

opera_7puRUAaSIK

1. Створіть функцію pow яка приймає число і ступінь до якого треба піднести число.

\*Для виконання цієї задачі ви не можете використовувати об’єкт Math або сторонні бібліотеки.

function pow(base, exponent) {

    if (exponent === 0) {

        return 1;

    } else if (exponent > 0) {

        let result = base;

        for (let i = 1; i < exponent; i++) {

            result \*= base;

        }

        return result;

    } else {

        return "Exponent should be a non-negative integer.";

    }

}

const result = pow(2, 3);

console.log(result);

opera_gCZupalE5p

1. Створіть чисту функцію powList яка приймає масив чисел першим аргументом, та значення ступеню другим. Фукція має повернутати масив чисел піднесених до степеню за допомогою фукції pow з 4 завдання.

\*Для виконання цього завдання обовязково використання одного з ітеруючих методів масиву, проте цей метод не може бути методом forEach або reduce

1. Створіть чисту функцію powAndFilter яка приймає 3 аргументи: масив чисел, ступінь, та найменше число. Функція має піднести всі числа масиву для ступеню, а потім відфільтрувати всі значення які менші за задане число і повернути відфільтрований масив

\* Для виконня цього завдання обовязково використувувати функції створенні в завданнях 3 та 5

1. Створіть чисту функцію sum. Функція sum як параметр приймає функцію handler, яка в свою чергу при виклику повертає масив чисел (для спрощення вам не треба перівяти чи дійсно вона повертає масив чисел). Функція sum має повертати функцію calc, яка в свою чергу при виклику, обраховує сумму всіх числе масиву який повернула функція handler, та повертає результат.

\* Для підрахунку суми числе треба використати метод масиву reduce

Приклад викликів фукцій sum, handler та calc

