**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**Практична робота №3**

**Поняття алгоритму. Властивості алгоритмів. Способи подання алгоритмів. Складність алгоритмів**

Виконав студент 1 курсу

спеціальності “Інженерія

програмного забезпечення”

Гарабаджіу Крістіан

**Ужгород-2025**

**Мета:** набути практичних навичок подання алгоритмів різними способами та визначення їх складності.

Завдання до роботи:

1. Написати код для виконання алгоритму відповідно до варіанту завдання. Аргументуйте використання обраного алгоритму. Визначте складність алгоритму.
2. Примітка. Не використовувати готові методи та функції Javascript
3. Оформити звіт, де подати алгоритм чотирма способами: словесним, псевдокодом, графічно (використовуючи доступні програмні продукти) та безпосередньо кодом програми.
4. Завантажити звіт в системі електронного навчання ДВНЗ «УжНУ» в установлений термін.
5. Підготувати відповіді на контрольні питання.

**Хід роботи:**

**Завдання:** Напишіть алгоритм, який знаходить перші n чисел Фібоначчі.

* Словесний спосіб:

1. Функція отримує число num (порядковий номер шуканого числа), а значення reg встановлюється за замовчуванням;
2. Визначити режим функції:
   1. Якщо reg дорівнює 1, то вивестив консоль результат виконання функції з параметрами num та reg = 2;
   2. Якщо ж reg дорівнює 2, то визначити значення, яке потрібно повернути:
      1. Якщо отримане num має значення 0 або 1, то повернути це значення;
      2. В іншому випадку, повернути суму результатів виклику функції для пошуку двох попередніх чисел Фібоначчі;
   3. В іншому випадку, вивести повідомлення про неправильний режим;

* Псевдокод:

Початок

Ввід: num, reg;

Якщо reg== 1:

Вивід результату Fibonacci(num, 2);

Якщо ж reg == 2:

Якщо num == 0 або num == 1:

Повернути num;

Інакше:

Повернути Fibonacci(num - 1, 2) + Fibonacci(num - 2, 2);

Інакше:

Повернути повідомлення про помилку;

Кінець

* Графічна візуалізація:

Початок

num, reg

**+**

reg == 1

**-**

**-**

**-**

Fibonacci(num, 2)

**+**

num == 0 || num == 1

reg == 2

**+**

`Це число ${Fibonacci(num, 2)}`

**-**

Fibonacci(num-1, 2)

"Неправильний режим"

num

Fibonacci(num-2, 2)

Fibonacci(num-1, 2) + Fibonacci(num-2, 2)

Кінець

* Код:

var Fibonacci = function (num, reg = 1){

if (reg == 1){

console.log(`Це число ${Fibonacci(num, 2)}`)

}else if (reg == 2){

if (num == 0 || num == 1){

return num;

}else{

return Fibonacci(num-1, 2) + Fibonacci(num-2, 2);

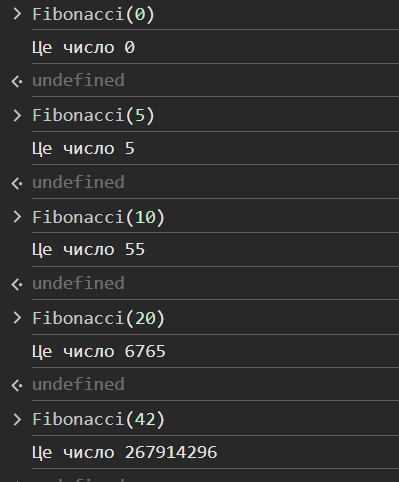
}

}else{

console.log("Неправильний режим");

}

}



**Висновок:**

У ході виконання практичної роботи було розглянуто алгоритм знаходження чисел Фібоначчі та його подання чотирма способами: словесним, псевдокодом, графічною візуалізацією (блок-схемою) та кодом. Реалізація алгоритму здійснюється через функцію Fibonacci, яка використовує рекурсивний підхід для обчислення чисел ряду Фібоначчі. Для управління процесом обчислення передбачено параметр reg, що дозволяє контролювати режим роботи функції: обчислювальний або кінцевий. У разі введення некоректного значення reg, алгоритм виводить відповідне повідомлення про помилку. Основним методом знаходження чисел є виклик функції для двох попередніх значень, що відповідає класичному рекурсивному підходу. Отримані результати підтверджують правильність роботи алгоритму та його відповідність заданим вимогам. Рекурсивний метод не найшвидший у плані виконання та може бути досить затребуваним у потужності пристрою, але є простим в реалізації та наглядним способом пошуку відповіді у закономірностях.