Міністерство Освіти України Державний університет Телекомунікацій

Практичне заняття 2. Створення програми з циклічними обчислювальними процесами

Підготував: студент групи ПД-21

Гапей Максим Юрійович

Перевірив: викладач

Яскевич В.О.

Мета роботи:

- Закріплення знань про структуру програми на мові JavaScript.
- Поглибити та закріпити знання про циклічні алгоритмічні структури.
- Придбати практичні навички створення програм, що містять повторюванні команди.
- Вивчення оператора циклу for, while, do while.
- Створення програм згідно завдання.

Завдання 1.

Трикутник

Програма має побудувати трикутник з елементів #. Кількість ліній задає користувач.

Вхідні дані: ціле невід'ємне число n.

Результат роботи:

```
#
##
###
####
#####
######
```

Зробити одну веб-сторінку, в якій підключити файли з зовнішніми скриптами. Кожне завдання розмістити в окремому файлі task_#.js. Введення даних здійснити за допомогою **prompt()**, виведення – в документ за допомогою **document.write()**.

Код програми:

```
var n = parseFloat(window.prompt("Enter the number a: "));
for(var i = 0; i < n; i++)
{
    for(var k = 0; k <= i; k++)
    {
        document.write("#");
    }
    document.write("<br>};
}
```

Завдання 2.

Два трикутники

Програма має побудувати трикутник з елементів #. Кількість ліній задає користувач.

Вхідні дані: ціле невід'ємне число n.

Результат роботи:

```
# #
## ##
### ###
#### ####
```

Зробити одну веб-сторінку, в якій підключити файли з зовнішніми скриптами. Кожне завдання розмістити в окремому файлі task_#.js. Введення даних здійснити за допомогою **prompt()**, виведення – в документ за допомогою **document.write()**.

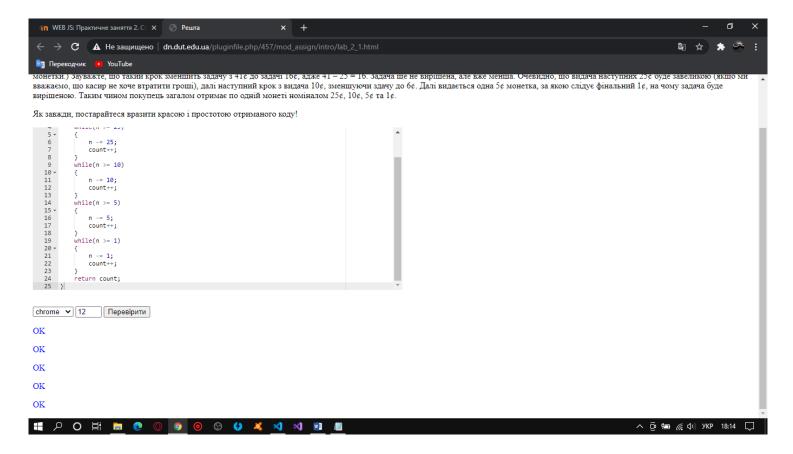
Код програми:

```
var n = parseFloat(window.prompt("Enter the number a:"));
for(var i = 0; i < n; i++)
{
    for(var k = n - i - 1; k > 0; k--) document.write("&nbsp;&nbsp;");
    for(var k = 0; k <= i; k++) document.write("#");
    document.write("&nbsp;&nbsp;");
    for(var k = 0; k <= i; k++) document.write("#");
    document.write("<br>});
}
```

Завдання 3.

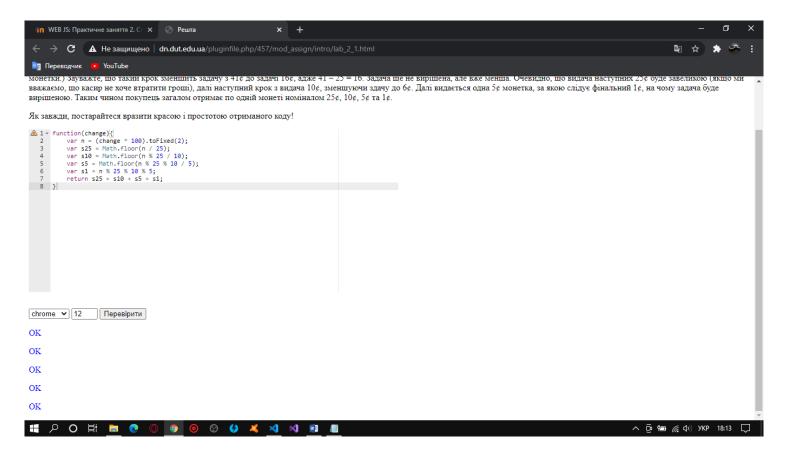
Рішення через ланцюжок циклів while:

```
function(change){
   var n = (change * 100).toFixed(2);
   var count = 0;
   while(n >= 25)
       n -= 25;
       count++;
   while(n >= 10)
       n -= 10;
       count++;
   while(n >= 5)
       n -= 5;
       count++;
   while(n >= 1)
       n -= 1;
        count++;
   return count;
```



Рішення за допомогою ділення по модулю:

```
function(change){
    var n = (change * 100).toFixed(2);
    var s25 = Math.floor(n / 25);
    var s10 = Math.floor(n % 25 / 10);
    var s5 = Math.floor(n % 25 % 10 / 5);
    var s1 = n % 25 % 10 % 5;
    return s25 + s10 + s5 + s1;
}
```



Висновок: поглибив та закріпив знання про циклічні алгоритмічні структури, придбав практичні навички створення програм, що містять повторюванні команди, а також вивчив оператори циклу *for, while, do while*.