Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего профессионального образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

Кафедра информационные системы и программная инженерия

Колледж инновационных технологий и предпринимательства

Дисциплина «Разработка интерфейсов пользователя»

Лабораторная работа №5

«Циклы и функции. Их применение»

Выполнила:

Студентка гр. ИРсп-122

Смыгина К.М

Проверил:

Фисина И.С.

Владимир 2024**Цель работы:**

1. Рассмотреть какие задачи решают циклы и функции в JavaScript;

2. Изучить виды циклов, когда применять каждый из видов;

3. Возможности прерывания циклов программно и переход к следующей итерации;

4. Передача параметров в функцию и возвращаемые значения.

5. Закрепить изученный материал путем самостоятельного выполнения заданий к практическому занятию.

**Задание:**

1. Даны целые числа K и N (N > 0). Вывести на странице N раз число K.

2. Даны два целых числа A и B ( A<B ). Вывести в порядке убывания все целые числа, расположенные между A и B (не включая числа A и B), а также количество N этих чисел.

3. Даны два целых числа A и B (A < B). Найти произведение всех целых чисел от A до B включительно.

4. Дано целое число N (> 0). Найти сумму 1 + 1/2 + 1/3 + … + 1/N.

5. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1, 2, …, 10 кг конфет.

6. Даны два целых числа A и B (A < B). Найти произведение всех целых чисел от A до B включительно. Произведение всех чисел не должно превышать 100 (используйте break).

7. Пользователь вводит положительное число N. Нужно найти и вывести на экран сумму всех четных и произведение всех нечетных чисел в интервале от 1 до N.

**Тема:**

Танцевальная студия.

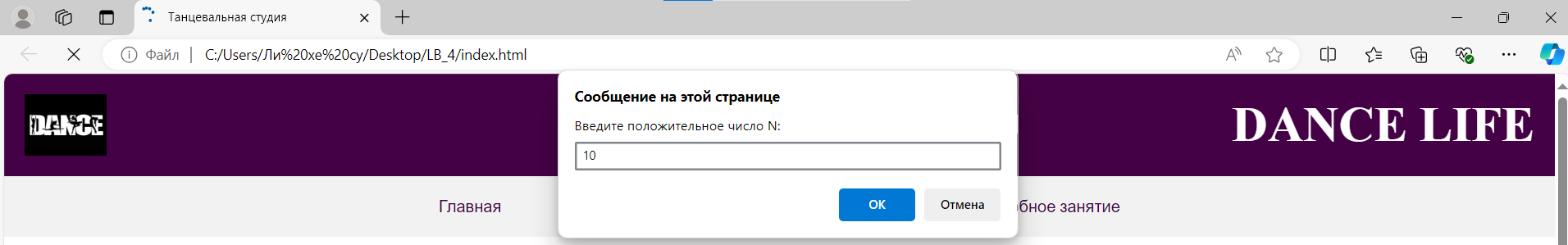


Рисунок 1. Результат работы

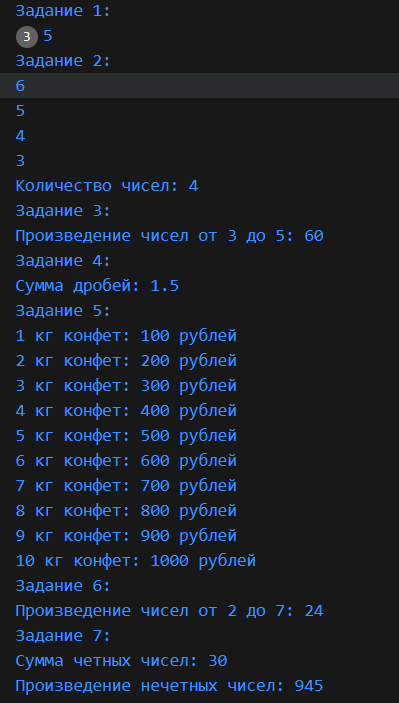


Рисунок 2. Результат работы

Краткое описание теории.

1. Цикл while

Синтаксис:

while (condition) {

// код

// также называемый "телом цикла"

}

Код из тела цикла выполняется, пока условие condition истинно. Например, пользователю предлагается ввести его любимый цвет, пока пользователь не введёт слово stop.

1. Цикл do…while

Синтаксис:

do {

// тело цикла

} while (condition);

Цикл сначала выполнит тело, а затем проверит условие condition, и пока его значение равно true, он будет выполняться снова и снова. Такая форма синтаксиса оправдана, если вы хотите, чтобы тело цикла выполнилось хотя бы один раз, даже если условие окажется ложным.

1. Цикл for

Синтаксис:

for (начало; условие; шаг) {

// ... тело цикла ...

}

For цикл очень гибкий. Вы можете использовать его для перебора элементов массива, работы со строками, решения математических задач и многих других целей. Цикл for будет выполняться только до тех пор, пока условие истинно. Если условие не изменится, цикл будет выполняться бесконечно.

1. Break

Обычно цикл завершается при вычислении условия в false. Но мы можем выйти из цикла в любой момент с помощью специальной директивы break. Например, следующий код подсчитывает сумму вводимых чисел до тех пор, пока пользователь вводит число, если пользователь ввел не цифровой символ – выдает полученную сумму. Сочетание «бесконечный цикл + break» – отличная штука для тех ситуаций, когда условие, по которому нужно прерваться, находится не в начале или конце цикла, а посередине или даже в нескольких местах его тела. директива continue – «облегчённая версия» break. При её выполнении цикл не прерывается, а переходит к следующей итерации (если условие все ещё равно true). Её используют, если понятно, что на текущем повторе цикла делать больше нечего.

1. Функции

Функция — это "подпрограмма", которую можно вызывать из внешнего (или внутреннего, в случае рекурсии) по отношению к функции кода. Как и сама программа, функция состоит из последовательности инструкций, называемой телом функции. Значения могут быть переданы в функцию, а функция вернёт значение.

В JavaScript функции являются объектами первого класса, то есть: они являются объектами и с ними можно взаимодействовать и передавать их точно так же, как любой другой объект. Если быть точным, функции — это объекты Function. Область видимости функций.

Переменные, объявленные в функции, не могут быть доступными где-нибудь вне этой функции, поэтому переменные (которые нужны именно для функции) объявляют только в scope функции. При этом функция имеет доступ ко всем переменным и функциям, объявленным внутри её scope. Другими словами, функция объявленная в глобальном scope имеет доступ ко всем переменным в глобальном scope.

Функция, объявленная внутри другой функции, еще имеет доступ и ко всем переменным её родительской функции и другим переменным, к которым эта родительская функция имеет доступ.

Описание выполнения заданий.

**Задача 1:**

1. **Вызов функции:** запускаем код, и в нем вызывается функция p(5, 3).
2. **Передача аргументов:** Значения 5 (число K) и 3 (количество раз N) передаются в функцию p.
3. **Инициализация цикла:** Цикл for инициализируется с i = 0.
4. **Проверка условия:** Проверяется условие i < N (т.е. 0 < 3). Так как условие истинно, цикл запускается.
5. **Выполнение тела цикла:** Внутри цикла вызывается console.log(K), что выводит число 5 на консоль.
6. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 1).
7. **Повторная проверка условия:** Условие i < N (т.е. 1 < 3) снова истинно, поэтому цикл повторяется.
8. **Выполнение тела цикла:** Снова выводится 5 на консоль.
9. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 2).
10. **Повторная проверка условия:** Условие i < N (т.е. 2 < 3) снова истинно, цикл повторяется.
11. **Выполнение тела цикла:** Снова выводится 5 на консоль.
12. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 3).
13. **Повторная проверка условия:** Условие i < N (т.е. 3 < 3) становится ложным. Цикл завершается.

**Задача 2:**

1. **Вызов функции:** Вызывается функция pr(2, 7).
2. **Передача аргументов:** Значения 2 (меньшее число A) и 7 (большее число B) передаются в функцию pr.
3. **Инициализация цикла:** Цикл for инициализируется с i = B - 1 (т.е. i = 6).
4. **Проверка условия:** Проверяется условие i > A (т.е. 6 > 2). Так как условие истинно, цикл запускается.
5. **Выполнение тела цикла:** Выводится число i (6) на консоль с помощью console.log(i). Значение count увеличивается на 1.
6. **Итерация цикла:** Значение i уменьшается на 1 (теперь i = 5).
7. **Повторная проверка условия:** Условие i > A (т.е. 5 > 2) снова истинно, цикл повторяется.
8. **Выполнение тела цикла:** Выводится число i (5) на консоль. Значение count увеличивается на 1.
9. **Итерация цикла:** Значение i уменьшается на 1 (теперь i = 4).
10. **Повторная проверка условия:** Условие i > A (т.е. 4 > 2) снова истинно, цикл повторяется.
11. **Выполнение тела цикла:** Выводится число i (4) на консоль. Значение count увеличивается на 1.
12. **Итерация цикла:** Значение i уменьшается на 1 (теперь i = 3).
13. **Повторная проверка условия:** Условие i > A (т.е. 3 > 2) снова истинно, цикл повторяется.
14. **Выполнение тела цикла:** Выводится число i (3) на консоль. Значение count увеличивается на 1.
15. **Итерация цикла:** Значение i уменьшается на 1 (теперь i = 2).
16. **Повторная проверка условия:** Условие i > A (т.е. 2 > 2) становится ложным. Цикл завершается.
17. **Вывод результата:** Выводится сообщение “Количество чисел: 4”, так как count равно 4.

**Задача 3:**

1. **Вызов функции:** Вызывается функция pri(3, 5).
2. **Передача аргументов:** Значения 3 (начальное число A) и 5 (конечное число B) передаются в функцию pri.
3. **Инициализация цикла:** Цикл for инициализируется с i = A (т.е. i = 3).
4. **Проверка условия:** Проверяется условие i <= B (т.е. 3 <= 5). Так как условие истинно, цикл запускается.
5. **Выполнение тела цикла:** Переменная product умножается на текущее число i (т.е. product = 1 \* 3 = 3).
6. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 4).
7. **Повторная проверка условия:** Условие i <= B (т.е. 4 <= 5) снова истинно, цикл повторяется.
8. **Выполнение тела цикла:** Переменная product умножается на текущее число i (т.е. product = 3 \* 4 = 12).
9. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 5).
10. **Повторная проверка условия:** Условие i <= B (т.е. 5 <= 5) снова истинно, цикл повторяется.
11. **Выполнение тела цикла:** Переменная product умножается на текущее число i (т.е. product = 12 \* 5 = 60).
12. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 6).
13. **Повторная проверка условия:** Условие i <= B (т.е. 6 <= 5) становится ложным. Цикл завершается.
14. **Возврат значения:** Функция возвращает значение product (т.е. 60).

**Задача 4:**

1. **Вызов функции:** Вызывается функция prim(2).
2. **Передача аргумента:** Значение 2 (количество дробей N) передается в функцию prim.
3. **Инициализация цикла:** Цикл for инициализируется с i = 1.
4. **Проверка условия:** Проверяется условие i <= N (т.е. 1 <= 2). Так как условие истинно, цикл запускается.
5. **Выполнение тела цикла:** К переменной sum добавляется значение 1 / i (т.е. sum = 0 + 1/1 = 1).
6. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 2).
7. **Повторная проверка условия:** Условие i <= N (т.е. 2 <= 2) снова истинно, цикл повторяется.
8. **Выполнение тела цикла:** К переменной sum добавляется значение 1 / i (т.е. sum = 1 + 1/2 = 1.5).
9. **Итерация цикла:** Значение i увеличивается на 1 (теперь i = 3).
10. **Повторная проверка условия:** Условие i <= N (т.е. 3 <= 2) становится ложным. Цикл завершается.
11. **Возврат значения:** Функция возвращает значение sum (т.е. 1.5).

**Задание 5:**

1. **Вызов функции:** Вызывается функция prime(100), передавая ей значение 100, которое представляет собой цену за 1 кг конфет.
2. **Цикл for:** Инициализируется цикл for с начальным значением i = 1 и условием i <= 10.
3. **Проверка условия:** Цикл запускается, так как условие i <= 10 истинно.
4. **Выполнение тела цикла:** Выводится строка i кг конфет: [стоимость] рублей, где:
   * i - текущее значение счетчика цикла (например, 1, 2, 3, …).
   * [стоимость] - результат умножения price (цена за кг) на i (количество кг).
5. **Итерация цикла:** Значение i увеличивается на 1, и цикл повторяется, пока условие i <= 10 истинно.
6. **Вывод на консоль:** После каждой итерации цикла выводится стоимость для соответствующего количества кг конфет (например, 1 кг: 100 рублей, 2 кг: 200 рублей, 3 кг: 300 рублей, … 10 кг: 1000 рублей).
7. **Завершение цикла:** Цикл завершается, когда i становится больше 10.

**Задание 6:**

1. **Вызов функции:** Вызывается функция primer(2, 7), передавая значения 2 (начальное число A) и 7 (конечное число B).
2. **Инициализация переменной:** Инициализируется переменная product со значением 1, которая будет хранить произведение чисел.
3. **Цикл for:** Инициализируется цикл for с начальным значением i = A (т.е. i = 2) и условием i <= B (т.е. i <= 7).
4. **Проверка условия:** Проверяется условие product \* i > 100:
   * Если условие истинно (т.е. произведение переменной product на текущее число i превышает 100), выполняется break, чтобы прервать цикл.
   * Если условие ложно, то product умножается на текущее число i (т.е. product \*= i).
5. **Итерация цикла:** Значение i увеличивается на 1, и цикл повторяется.
6. **Вывод на консоль:** После завершения цикла (или его прерывания) выводится значение product, которое хранит произведение чисел от A до B включительно, но не больше 100.

**Задание 7:**

1. **Ввод числа N:** Пользователь вводит число N через окно ввода prompt().
2. **Преобразование ввода:** Введенное значение N преобразуется в целое число с помощью parseInt(N).
3. **Инициализация переменных:** Инициализируются переменные sum (для суммы четных чисел) и h (для произведения нечетных чисел) с начальными значениями 0 и 1 соответственно.
4. **Цикл for:** Инициализируется цикл for с начальным значением i = 1 и условием i <= N.
5. **Проверка четности:** Внутри цикла выполняется проверка if (i % 2 === 0):
   * Если число i четное (остаток от деления на 2 равен 0), то текущее значение i добавляется к sum.
   * Если число i нечетное, то текущее значение i умножается на h.
6. **Итерация цикла:** Значение i увеличивается на 1, и цикл повторяется.
7. **Вывод результатов:** После завершения цикла выводится сообщение “Сумма четных чисел: [sum]” и “Произведение нечетных чисел: [h]”.

**Вывод:** в ходе выполнения лабораторной работы я рассмотрела какие задачи решают циклы и функции в JavaScript, изучила виды циклов, когда применять каждый из видов, возможности прерывания циклов программного и переход к следующей итерации, передача параметров в функцию и возвращаемые значения, закрепила изученный материал путем самостоятельного выполнения заданий к практическому занятию.