

Лабораторная работа №8

Повторяющиеся вычисления - циклы

Для успешного решения широкого круга задач требуется многократно повторить некоторую последовательность действий, записанную в программе один раз. В том случае, когда число повторений последовательности действий нам неизвестно, либо число повторений зависит от некоторых условий, можно воспользоваться оператором цикла вида:

```
while (B) {s}
```

где B - выражение логического типа; s - операторы, называемые телом цикла. Операторы s в фигурных скобках выполняются до тех пор, пока условие B не станет ложным.

Пример 1. Нахождение общего делителя

Напишем программу, которая для двух заданных чисел определяет наибольший общий делитель.

При решении задачи воспользуемся алгоритмом Евклида. Если значение m равно нулю, то наибольший общий делитель чисел n и m равен n:

$$\text{НОД}(n, 0) = n.$$

В остальных случаях верно следующее соотношение:

$$\text{НОД}(n, m) = \text{НОД}(m, n \% m).$$

В функции nod переменная p используется для получения остатка от деления чисел n и m (листинг 1). Выполнение цикла продолжается до тех пор, пока значение p не станет равным нулю. Последнее вычисленное значение m равно наибольшему общему делителю.

Листинг 1. Наибольший общий делитель двух чисел

```
<HTML>
<HEAD>
<TITLE>Наибольший общий делитель двух чисел</TITLE>
<script language="JavaScript">
<!-- //
function nod(obj)
{ var n=obj.num1.value
  var m=obj.num2.value
  var p = n%m
  while (p!=0)
  { n=m
    m=p
    p=n%m
  }
}
```

```

obj.res.value=m
}
//-->
</script>
</HEAD>
<BODY>
Наибольший общий делитель двух заданных чисел
<FORM name="form1">
Введите число <input type="text" name="num1" size="8"><br>
Введите число <input type="text" name="num2" size="8"><br>
<input type="button" value="Вычислить" onClick="nod(form1)"><br>
Наибольший общий делитель <input type="text" name="res"
size="8"><hr>
<input type="reset" value="Отменить">
</FORM>
</BODY>
</HTML>

```

Если число повторений заранее известно, то можно воспользоваться следующим оператором цикла, который часто называют оператором цикла арифметического типа. Синтаксис этого оператора таков:

```
for (A; B; I){S}
```

Выражение A служит для инициализации параметра цикла и вычисляется один раз в начале выполнения цикла. Выражение B (условие продолжения) управляет работой цикла. Если значение выражения ложно, то выполнение цикла завершается, если истинно, то выполняется оператор S, составляющий тело цикла. Выражение I служит для изменения значения параметра цикла. После выполнения тела цикла S вычисляется значение выражения I, затем опять вычисляется значение выражения B и т.д. Цикл может прекратить свою работу в результате выполнения оператора break в теле цикла.

Пример 2. Совершенные числа

Напишем программу, определяющую, является ли заданное число n совершенным.

Совершенным называется число n, равное сумме своих делителей, не считая самого числа. Например, число 6 является совершенным, т. к. верно $6 = 1 + 2 + 3$, где 1, 2, 3 - делители числа 6. Число 28 также является совершенным, справедливо равенство $28 = 1 + 2 + 4 + 7 + 14$. При решении задачи воспользуемся только функцией `sumdei` (листинг 2).

Листинг 2. Итерационные методы. Совершенные числа

```

<HTML>
<HEAD>
<TITLE>Итерационные методы. Совершенные числа</TITLE>
<script language="JavaScript">
<!-- //
function sumdel(n)
{ var s=1;
for (var i=2; i<=n/2; i++)
{ if (n % i == 0) s += i }
return s
}
function sov(obj)
{ var n=obj.numb.value;
var s=""
if (n==sumdel(n)) s="совершенное"
else s="не является совершенным"
return s
}
//-->
</script>
</HEAD>
<BODY>
<P> Итерационные методы. Совершенные числа</P>
<FORM name="form0">
Введите натуральное число: <input type="text" size=8 name="numb">
<input type="button" value=Выполнить
onClick="this.form.res.value=sov(form0)"><hr>
Данное число: <input type="text" size=24 name="res"><hr>
<input type="reset" value=Отменить>
</FORM>
</BODY>
</HTML>

```

Обратите внимание на значение параметра обработки события. В данном случае это оператор присваивания, в правой части которого вызов функции sov.

Оператор for...in используется для анализа свойств объекта. Синтаксис оператора:

```
for (i in t) {s}
```

где i - переменная цикла; t - объект; s - последовательность операторов.

В результате выполнения оператора цикла производится перебор свойств объекта. Переменная цикла при каждом повторении содержит значение свойства объекта. Количество повторений тела цикла s равно числу свойств, определенных для объекта t.

Пример 3. Определение свойств элемента формы

Напишем сценарий, с помощью которого можно определить свойства элемента формы "поле ввода многострочного текста".

Свойства объекта с помощью оператора цикла формируются в строке result, затем после просмотра всех свойств значение строки result помещается в поле ввода многострочного текста.

Сценарий определения свойств текстового поля приведен в листинге 3.

Листинг 3. Операции над объектами. Свойства текстового поля

```
<HTML>
<HEAD>
<TITLE>Операции над объектами. Свойства текстового поля</TITLE>
<script language="JavaScript">
<!-- //
function propobj (obj)
{ var result = ""
for (var i in obj)
{ result += obj.data.value + "." + i + " = " +(obj.data)[i]+"\r\n" }
result += "\n\r"
form1.data.value=result
}
//-->
</script>
</HEAD>
<BODY bgcolor=F8F8FF>
<CENTER>
<H4>Определение свойств объектов</H4>
<FORM name="form1">
<input type="button" value=Выполнить onClick="propobj (form1)"><hr>
<textarea name="data" cols=30 rows=10 id=1>Текст</textarea><hr>
<input type="reset" value=Очистить>
</CENTER>
</FORM>
</BODY>
</HTML>
```

Задания

1. Проверить примеры из лабораторной работы.
2. Добавьте на ваш сайт страницу «Лотерея». На странице задайте кнопку «Играть» при нажатии на кнопку случайным образом появляется шестизначное число. Выпавшее число является выигрышным если сумма первых трех чисел равна сумме вторых трех чисел.

Например, выпало число 586748

где $5+8+6=7+4+8$ =вы повезло, для вас скидка на занятия 50%

3. Напишите программу, которая "переворачивает" случайное число.

Пример: 586748=847685