

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДВОРЕЦ ДЕТСКОГО (ЮНОШЕСКОГО) ТВОРЧЕСТВА ВЫБОРГСКОГО РАЙОНА
САНКТ-ПЕТЕРБУРГА
(ГБУ ДО ДДОТ)

Практические занятия
Модуль «Программирование на языке Pascal»

«Основы компьютерных технологий»

Второй год обучения
Группа № 1
Возраст учащихся 9 — 12 лет

Ухтина Елена Николаевна,
педагог дополнительного образования

Санкт-Петербург
2016 / 2017 учебный год

Цель занятий: познакомить учащихся с циклами (цикл с параметром, цикл с предусловием, цикл с постусловием)

Задачи:

Ⓢ *Обучающие:*

Ⓢ сформировать у учащихся первичные знания по применению изученного материала.

Ⓢ *Развивающие:*

Ⓢ учить анализировать, обобщать и систематизировать;

Ⓢ обогащать словарный запас учащихся.

Ⓢ *Воспитательные:*

Ⓢ развивать информационную культуру учащихся, способность к самостоятельной и коллективной деятельности, рефлексия.

Тип занятия: изучения и первичного закрепления новых знаний.

Оборудование: мультимедийный проектор, компьютеры с установленной средой программирования Pascal, настроенная локальная сеть, конспект для учащегося раздаточные карточки-задания.

Занятие 1. Циклы. Сумма и произведение цифр числа

Задача

Найти сумму и произведение цифр, введенного натурального числа. Например, если введено число 325, то сумма его цифр равна 10 ($3+2+5$), а произведение 30 ($3*2*5$).

Пояснение к задаче и алгоритм решения

1. Присвоить переменной, в которой будет накапливаться сумма цифр, значение 0, а переменной для накопления произведения - значение 1 (если присвоить 0, то при умножении результат обнулится).
2. Пока введенное число не уменьшится до нуля выполнять следующие действия:
 1. В переменную для суммы цифр добавлять последнюю цифру числа, извлеченную при нахождении остатка от деления на 10.
 2. Значение переменной для хранения произведения цифр умножить на последнюю цифру числа.
 3. Избавиться от последней цифры числа, разделив его нацело на 10.

Исходный код на языке программирования Pascal

```
var
n, sum: word;
mult: longint;
begin
readln(n);
sum := 0;
mult := 1;
while n > 0 do begin
sum := sum + n mod 10;
mult := mult * (n mod 10);
n := n div 10;
end;
writeln('Sum: ', sum);
writeln('Mult: ', mult);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
462
Sum: 12
Mult: 48
```

Занятие 2. Циклы. Вывести ряд чисел в диапазоне с шагом

Задача

Вывести на экран ряд натуральных чисел от минимума до максимума с шагом. Например, если минимум 10, максимум 35, шаг 5, то вывод должен быть таким: 10 15 20 25 30 35. Минимум, максимум и шаг указываются пользователем (считываются с клавиатуры).

Пояснение к задаче и алгоритм решения

Для решения данной задачи лучше всего использовать цикл со счетчиком (for), если в нем можно использовать шаг. Такие циклы есть в языках Basic, КуМир и Си. В Python for является циклом перебора, а не циклом со счетчиком, однако данная задача легко решается с помощью for, если использовать функцию range(). В Pascal цикл for не предусматривает использование шага, поэтому задачу проще решить с помощью иных циклов.

Алгоритм же решения очень прост: пока минимум не достигнет максимума выводить его на экран и затем увеличивать на значение шага.

Исходный код на языке программирования Pascal

```
var a,b,c: word;
begin
  readln(a);
  readln(b);
  readln(c);
  while a<=b do begin
    write(a, ' ');
    a := a+c;
  end;
  writeln;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
6
162
16
6 22 38 54 70 86 102 118 134 150
```

Особенности решения на языке программирования Pascal

Поскольку в Паскале нет возможности указывать шаг в цикле for и в теле цикла нельзя менять переменную-счетчик, то проще решить данную задачу с помощью цикла while.

Занятие 3. Циклы. Вывести ряд чисел в диапазоне с шагом

Задача

Вычислить факториал введенного числа.

Пояснение к задаче и алгоритм решения

Факториалом числа называют произведение всех натуральных чисел до этого числа включительно. Например, факториал числа 4 равен $1*2*3*4 = 24$. Записывается факториал так: $4! = 24$.

Поскольку факториал резко увеличивается с каждым следующим числом не следует вводить больших чисел.

1. Присвоим переменной, накапливающей произведение натуральных чисел, начальное значение 1.
2. Присвоим переменной-счетчику значение 2.
3. Пока переменная счетчик не достигнет числа, введенного пользователем,
 1. умножить значение переменной, в которой накапливается произведение, на значение переменной счетчика,
 2. увеличивать счетчик на 1.

Исходный код на языке программирования Pascal

```
var
n,i: byte;
f: longint;
begin
readln(n);
if n<>0 then f:=1;
for i:=2 to n do f := f*i;
writeln(f);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
8
40320
```

Занятие 4. Циклы. Таблица значений функции

Задача

Вывести таблицу значений функции $y = -0.23x^2 + x$. Значения аргумента (x) задаются минимумом, максимумом и шагом. Например, если минимум задан как 1, максимум равен 3, а шаг 0.5. То надо вывести на экран изменение x от 1 до 3 с шагом 0.5 (1, 1.5, 2, 2.5, 3) и значения функции (y) при каждом значении x .

Пояснение к задаче и алгоритм решения

1. Получить с клавиатуры значения x_1 (минимум x), x_2 (максимум x) и up (шаг).
2. Пока x_1 не превысит x_2 выполнять следующие действия:
 1. вычислить значение y ,
 2. вывести на экран текущие значения x_1 и y ,
 3. увеличить значение x_1 на значение шага (up).

Исходный код на языке программирования Pascal

```
var
x1,x2,up,y: real;
begin
  readln(x1,x2,up);
  while x1 <= x2 do begin
    y := -0.23*x1*x1 + x1;
    writeln(x1:5:2, ' -> ',y:5:2);
    x1 := x1 + up;
  end;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
1
5
0.5
1.00 -> 0.77
1.50 -> 0.98
2.00 -> 1.08
2.50 -> 1.06
3.00 -> 0.93
3.50 -> 0.68
4.00 -> 0.32
4.50 -> -0.16
5.00 -> -0.75
```

Занятие 5. Циклы. Вероятность четных случайных чисел

Задача

Проверить корректность работы генератора псевдослучайных чисел языка программирования с помощью оценки вероятности выпадения четных чисел на выборке не меньше 1000 случайных чисел.

Пояснение к задаче и алгоритм решения

Встроенный в язык программирования генератор псевдослучайных чисел должен генерировать разные числа в заданном диапазоне с равной вероятностью. Это значит, что, например, количество четных и нечетных чисел должно быть примерно одинаковым (при условии, что диапазон задан так, что в нем количество четных и нечетных одинаково). Чем больше выборка, тем точнее результат исследования.

1. Создадим счетчик четных случайных чисел и присвоим ему значение 0.
2. Сгенерируем 1000 случайных чисел (например, от 0 до 99) и проверим каждое из них на четность. Четным является то число, при делении которого на 2 остаток равен 0. Если очередное число четное, то будем увеличивать счетчик на 1.
3. Вероятность выпадения четных чисел оценим в процентах. Для этого количество четных чисел разделим на общее количество сгенерированных (в данном случае 1000) и умножим на 100.

Программа должна выдать около 50%. Можно увеличить число генераций, например до 10 тысяч. В таком случае отклонение от 50% будет меньше.

Исходный код на языке программирования Pascal

```
const N = 1000;
var i,q: integer;
begin
randomize;
q := 0;
for i:=1 to N do
if random(100) mod 2 = 0 then
q := q + 1;
writeln(q / N * 100:4:2,'%');
end.
```

Пример(ы) выполнения программы на языке Pascal

48.80%

Занятие 6. Циклы. Простейший калькулятор

Задача

Написать программу, которая будет складывать, вычитать, умножать или делить два числа. Числа и знак операции вводятся пользователем. После выполнения вычисления программа не должна завершаться, а должна запрашивать новые данные для вычислений. Завершение программы должно выполняться при вводе символа '0' в качестве знака операции. Если пользователь вводит неверный знак (не '0', '+', '-', '*', '/'),

то программа должна сообщать ему об ошибке и снова запрашивать знак операции. Также сообщать пользователю о невозможности деления на ноль, если он ввел 0 в качестве делителя.

Пояснение к задаче и алгоритм решения

Запустить бесконечный цикл.

1. Запросить ввод знака операции.
2. Если был введен '0', то прервать цикл операцией break.
3. Если был введен один из знаков арифметических операций, то в зависимости от того, какой знак был введен, выполнить соответствующее ему арифметическое действие. При этом при делении проверить не был ли введен ноль. Если это так, то вывести сообщение, иначе выполнить деление.
4. Иначе сообщить о неверно введенном знаке.

Исходный код на языке программирования Pascal

```
var
x,y,z: real;
sign: char;
begin
while True do begin
write('sign: '); readln(sign);
if sign = '0' then break;
if sign in ['+', '-', '*', '/'] then begin
write('x='); readln(x);
write('y='); readln(y);
case sign of
'+': begin z := x + y; writeln(z:5:2) end;
'-': begin z := x - y; writeln(z:5:2); end;
'*': begin z := x * y; writeln(z:5:2); end;
'/': if y <> 0 then begin z := x / y; writeln(z:5:2)
end
else
writeln('Division by zero!');
end;
end
else writeln('Bad sign!')
end;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
sign: 8
Bad sign!
sign: 4
Bad sign!
sign: -
x=5
y=6
-1.00
```


sign: +
x=4.45
y=10.66
15.11
sign: *
x=-3.4
y=-67
227.80
sign: /
x=-78
y=123
-0.63
sign: /
x=9
y=0
Division by zero!
sign: i
Bad sign!
sign: 0

Занятие 7. Циклы. Посчитать четные и нечетные цифры числа

Задача

Посчитать четные и нечетные цифры введенного натурального числа. Например, если введено число 34560, то у него 3 четные цифры (4, 6 и 0) и 2 нечетные (3 и 5).

Пояснение к задаче и алгоритм решения

1. Присвоить переменным-счетчикам четных (*even*) и нечетных (*odd*) цифр

- значение 0.
2. Пока введенное число не уменьшится до нуля выполнять нижеследующие действия:
 1. Если число четное (делится нацело на 2), значит последняя его цифра четная и надо увеличить на 1 переменную *even*.
 2. Иначе число нечетное и следует добавить 1 к переменной *odd*.
 3. Убрать последнюю цифру числа путем деления числа нацело на 10.

Исходный код на языке программирования Pascal

```
var n: word;  
    even, odd: byte;  
begin  
  readln(n);  
  even := 0;  
  odd := 0;  
  while n > 0 do begin  
    if n mod 2 = 0 then  
      even := even + 1  
    else  
      odd := odd + 1;  
    n := n div 10;  
  end;  
  writeln('четных ', even);  
  writeln('нечетных ', odd);  
end.
```

Пример(ы) выполнения программы на языке Pascal

```
56709  
четных 2  
нечетных 3
```

Занятие 8. Циклы. Прямоугольник из символов

Задача

Вывести на экран "прямоугольник", образованный из двух видов символов. Контур прямоугольника должен состоять из одного символа, а в "заливка" - из другого.

Пояснение к задаче и алгоритм решения

Пусть прямоугольник будет состоять из 20 символов по горизонтали и 10 по вертикали.

1. Перебираем строки друг за другом (внешний цикл).
 1. Если строка первая или последняя, то всю ее выводим одним символом.
 2. Иначе выводим один символ, затем 18 знаков другим символом, после чего снова выводим первый.
 3. Переходим на новую строку.

Исходный код на языке программирования Pascal

```

var i,j: byte;
begin
  for i:=1 to 10 do begin
    if (i=1) or (i=10) then
      for j:=1 to 20 do
        write('-')
      else begin
        write('|');
        for j:=2 to 19 do
          write('0');
        write('|');
        end;
        writeln;
      end;
  end.

```

Особенности решения на языке программирования Pascal

Результат выполнения программы:

```

-----
|000000000000000000|
|000000000000000000|
|000000000000000000|
|000000000000000000|
|000000000000000000|
|000000000000000000|
|000000000000000000|
|000000000000000000|
-----

```

Занятие 9. Циклы. Представить натуральное число в виде простых сомножителей

Задача

Вывести на экран, из каких простых множителей состоит введенное натуральное число.

Пояснение к задаче и алгоритм решения

Чтобы найти все простые сомножители натурального числа, надо его пробовать

делить на простые числа, начиная с 2. Если заданное число делится без остатка, значит его делитель - это число, которое входит в состав сомножителей, из которых формируется заданное число. Как только такой сомножитель будет найден, заданное число следует на него разделить, т.е. получить новое заданное число и уже к нему заново подбирать простой делитель. Например, дано число 24. Первое натуральное число, на которое оно делится, - это 2. Значит 2 - это первый простой сомножитель. В результате деления получается 12. Далее снова находим, что 12 делится на 2. Далее 6 делится на 2. Далее 3 делится на 3. Таким образом получаем: $24 = 2 * 2 * 2 * 3$.

Поскольку список простых чисел заранее неизвестен, то можно подбирать делители, увеличивая каждый следующий на 1, а не искать простые числа. Ведь если число не делится на 2, то оно не разделится и на 4, 6 и т.д.

В программах ниже есть решения с использованием оператора goto и без него. С ним решение получается короче, но с точки зрения современного программирования его использование нежелательно.

Алгоритм сводится к следующему. Пока заданное число не будет сведено к 1, делить его на натуральные числа от 2 и т.д. Как только деление будет произведено без остатка, то значит был найден простой сомножитель. Он выводится на экран, заданное число на него делится и снова начинается поиск простого делителя, начиная с 2.

Исходный код на языке программирования Pascal

```
label again;
var n, i: word;
begin
  readln(n);
  again:
  while n > 1 do begin
    i := 2;
    while True do
      if n mod i = 0 then begin
        n := n div i;
        write(i, ' ');
        goto again;
      end
      else i := i + 1;
    end;
    writeln;
  end.
```

Пример(ы) выполнения программы на языке Pascal

```
507
3 13 13
```

Занятие 9. Циклы. Вывести на экран таблицу умножения

Задача

Вывести на экран таблицу умножения (от 1 до 9).

Пояснение к задаче и алгоритм решения

Перебрать во внешнем цикле числа от 1 до 9. Для каждого из них перебрать во внутреннем цикле числа от 1 до 9. Во внутреннем цикле выполнять умножение переменных-счетчиков внешнего и внутреннего циклов. Таким образом на одну итерацию внешнего цикла произойдет девять итераций внутреннего, и сформируется одна строка таблицы умножения. После каждой строки надо перейти на новую. Это делается во внешнем цикле, после того как закончится выполняться внутренний.

Для построения таблицы необходимо использовать форматированный вывод, т.е. задавать ширину столбцов, иначе произойдет сдвиг, т.к. количество цифр в каждой строке различно.

Исходный код на языке программирования Pascal

```
var i,j: byte;  
begin  
  for i:=1 to 9 do begin  
    for j:=1 to 9 do  
      write(i*j:4);  
    writeln;  
  end;  
end.
```

Пример(ы) выполнения программы на языке Pascal

```
1 2 3 4 5 6 7 8 9  
2 4 6 8 10 12 14 16 18  
3 6 9 12 15 18 21 24 27  
4 8 12 16 20 24 28 32 36  
5 10 15 20 25 30 35 40 45  
6 12 18 24 30 36 42 48 54  
7 14 21 28 35 42 49 56 63  
8 16 24 32 40 48 56 64 72  
9 18 27 36 45 54 63 72 81
```