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

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 массивами (одномерный и двумерный массив)

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, рефлексию.

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. Массивы (Сумма и произведение элементов массива)

Задача

Заполнить массив вещественных чисел вводом с клавиатуры. Посчитать сумму и произведение элементов массива. Вывести на экран сам массив, полученные сумму и произведение его элементов.

Пояснение к задаче и алгоритм решения

1. Присвоить переменной, в которой будет храниться сумма, значение 0, а переменной для произведения - значение 1.
2. В цикле увеличивая индекс массива от начала до конца
    1. считывать с клавиатуры число и записывать его в текущую ячейку массива,
    2. увеличивать переменную с суммой на введенное число.
    3. умножать переменную с произведением на введенное число.
3. В цикле увеличивая индекс массива от начала до конца вывести все элементы массива.
4. Вывести сумму и произведение.

Исходный код на языке программирования Pascal

```
const N = 10;
var
a: array[1..N] of real;
sum, mult: real;
i: integer;
begin
sum := 0;
mult := 1;
or i:=1 to N do begin

read(a[i]);
sum := sum + a[i];
mult := mult * a[i];
end;
for i:=1 to N do write(a[i]:5:2);
writeln;
writeln(sum:5:2);
writeln(mult:5:2);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
4.3 2.3 4.5 3.3 8.3 1.2 8.3 9.1 2.3 3.6
 4.30 2.30 4.50 3.30 8.30 1.20 8.30 9.10 2.30 3.60
47.20
914812.11
```

Занятие 2. Массив. Количество положительных, отрицательных и равных нулю элементов массива

Задача

Сгенерировать 20 случайных целых чисел в диапазоне от -5 до 4, записать их в ячейки массива. Посчитать сколько среди них положительных, отрицательных и нулевых значений. Вывести на экран элементы массива и посчитанные количества.

Пояснение к задаче и алгоритм решения

1. Присвоить счетчикам положительных, отрицательных и нулевых чисел значения 0.
2. В цикле
    1. генерировать случайное число и записывать его в соответствующую ячейку массива,
    2. выводить на экран,
    3. сравнивать с нулем и в зависимости от результата увеличить на 1 либо счетчик пложительных чисел, либо отрицательных, либо нулевых.
3. Вывести на экран значения счетчиков.

Исходный код на языке программирования Pascal

Процедура **Randomize** используется в Паскаль для включения генератора случайных чисел. Функция **Random** определяет диапазон случайных чисел. Процедура Randomize и функция Random очень часто используются для демонстрации работы массивов в Паскаль.

```
const N = 20;
var
a: array[1..N] of integer;
i, pos, neg, zero: byte;
begin
randomize;
pos := 0;
neg := 0;
zero := 0;
for i:=1 to N do begin
a[i] := random(10)-5;
write(a[i]:3);
if a[i] < 0 then
neg := neg + 1
else
if a[i] > 0 then
pos := pos + 1
else
zero := zero + 1;
end;
writeln;
writeln('Положительных: ', pos);
writeln('Отрицательных: ', neg);
writeln('Равных нулю: ', zero);
end.
```

Пример(ы) выполнения программы на языке Pascal

0 0 -4 -2 4 0 2 -4 3 0 0 -4 2 2 0 -3 -3 3 -3 -4
Положительных: 6
Отрицательных: 8
Равных нулю: 6

Занятие 3. Массив. Сколько элементов массива больше по модулю максимального

Задача

Создать массив из 20 элементов в диапазоне значений от -15 до 14 включительно. Определить количество элементов по модулю больших, чем максимальный.

Пояснение к задаче и алгоритм решения

Суть задачи сводится к тому, что сначала надо найти максимальный элемент массива, а затем сравнить с ним абсолютные значения отрицательных элементов и посчитать те, которые окажутся больше него.

По факту код программы получается проще, если сравнивать с максимумом все элементы массива. В результате алгоритм решения задачи сводится к следующему:

1. Присвоить переменной, в которой будет храниться максимум, значение меньше, чем возможное в массиве (в данном случае -16).
2. Перебрать все элементы массива. Если очередной элемент больше значения, которое хранится в переменной для максимума, то поменять значение этой переменной на текущий элемент.
3. Присвоить переменной для подсчета элементов больших по модулю максимальный значение 0.
4. Перебрать все элементы массива. Если абсолютное значение текущего элемента больше того, что записано в переменной для максимума, то увеличить переменную-счетчик (из предыдущего пункта) на 1.

Исходный код на языке программирования Pascal

```
const N = 20;
var
a: array[1..N] of integer;
max: integer;
i, count: byte;
begin
randomize;
max := -16;
for i:=1 to N do begin
a[i] := random(30)-15;
write(a[i],' ');
if a[i] > max then max := a[i];
end;
writeln;
count := 0;
for i:=1 to N do
if abs(a[i]) > max then
count := count + 1;
writeln(count);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
1 -6 -13 12 -3 12 -8 11 12 5 -7 -4 -9 -13 -13 -5 -6 0 -5 1
3
```

Особенности решения на языке программирования Pascal

В примере максимум равен 12. Больше него по модулю число -13, которое в массиве встречается 3 раза.

Занятие 4. Массив. Найти наибольший элемент и его порядковый номер в массиве

Задача

Заполнить одномерный массив случайными числами. Найти и вывести на экран

наибольший его элемент и порядковый номер этого элемента.

Пояснение к задаче и алгоритм решения

Заполнение массива и поиск наибольшего элемента можно выполнять в одном цикле.

Поскольку необходимо найти не только максимальный элемент, но и его индекс, то лучше искать индекс, так как по нему всегда можно получить значение из массива. Конечно, при поиске можно сохранять и индекс, и элемент в двух разных переменных. Однако этого делать не обязательно. До цикла присвоим переменной, в которой будет храниться индекс максимального элемента, значение 1. Это значит, предполагается, что максимальный элемент находится в первой ячейке массива.

Тело цикла будет состоять из следующих действий:

1. Сгенерировать случайное число и записать его в очередную ячейку массива.
2. Вывести полученное число на экран.
3. Если это число больше, чем то, что хранится под индексом, записанным в переменную-максимум, то присвоить этой переменной текущий индекс (не само число!).

После того, как индекс наибольшего элемента будет найден, вывести его на экран. Чтобы вывести элемент по данному индексу, надо использовать выражение извлечения элемента из массива. Например, если max - это индекс, а arr - массив, то выражение будет таким: arr[max].

Исходный код на языке программирования Pascal

```
const N = 10;
var
arr: array[1..N] of integer;
i, max: byte;
begin
randomize;
max := 1;
 for i:=1 to N do begin
arr[i] := random(100);
write(arr[i], ' ');
if arr[max] < arr[i] then
 max := i;
end;
writeln;
writeln('arr[',max,'] = ',arr[max]);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
64 26 99 37 57 64 6 21 48 19
arr[3] = 99
```

Занятие 5. Массив. Среднее арифметическое положительных элементов массива

Задача

Найти среднее арифметическое положительных элементов линейного массива.

Пояснение к задаче и алгоритм решения

Данная задача имеет смысл, если массив заполнен не только положительными

числами, но также содержит отрицательные числа и/или возможно нули.

Для заполнения массива можно воспользоваться генератором псевдослучайных чисел. Допустим, надо сгенерировать числа в диапазоне от -5 до 4 включительно. Всего значений 10, смещение на -5. Таким образом, с помощью стандартной функции генерируем числа от 0 до 10 и вычитаем из них 5.

Среднее арифметической находится как отношение суммы чисел к количеству этих чисел. Поскольку надо найти среднее арифметическое только положительных чисел, то, перебирая массив, нам надо определять положительные числа, добавлять их к общей сумме, а также считать их количество. Для этого потребуется две переменные (например, sum и qty), которым до цикла следует присвоить значение 0.

Осуществлять проверку с помощью условного оператора if можно как в отдельном цикле перебора массива, так и в цикле его заполнения. Если очередной элемент массива больше нуля, то следует его значение добавить к переменной sum, а значение переменной qty увеличить на 1.

После того как все элементы массива проверены, надо найденную сумму положительных чисел разделить на их количество. Тем самым будет найдено среднее арифметическое.

Исходный код на языке программирования Pascal

```
const N = 20;
var
arr: array[1..N] of integer;
i, qty: byte;
sum: word;
begin
sum := 0;
qty := 0;
randomize;
for i:=1 to N do begin
arr[i] := random(10)-5;
write(arr[i],' ');
 if arr[i] > 0 then begin
sum := sum + arr[i];
qty := qty + 1;
end;
 end;
writeln;
writeln(sum/qty:7:2);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
-1 2 -5 -5 -3 1 4 -2 -2 -1 -4 3 -2 1 -2 -1 1 -4 4 4
2.50
```

Занятие 6. Массив. Номер минимального по модулю элемента массива

Задача

Найти номер минимального по модулю элемента массива.

Например, в массиве [10, -3, -5, 2, 5] минимальным по модулю элементом является число 2. Его номер 4 (в языках, в которых индексация массивов начинается с нуля, его индекс будет равен 3).

Пояснение к задаче и алгоритм решения

Если стоит задача найти минимальный (или максимальный) элемент по модулю, то

значит при поиске нужно сравнивать не сами элементы массива, а их абсолютные значения (модули). Большинство языков программирования имеют встроенную функцию (например, abs()), которая возвращают модуль числа.

Поскольку требуется найти номер минимального по модулю элемента, а не сам элемент (его значение), то при поиске необходимо сохранять индекс найденного на данный момент элемента.

Алгоритм поиска индекса минимального по модулю элемента массива следующий:

1. Вводим переменную (например, num) и присваиваем ей индекс первого элемента массива (0 или 1 в зависимости от особенностей языка программирования). Тем самым предполагаем, что первый элемент массива и является минимальным по модулю.
2. Начинаем в цикле перебор массива со второго элемента и до конца. При этом в теле цикла в заголовке условного оператора (if) сравниваем модуль текущего элемента с модулем элемента, чей индекс хранится в переменной num.
3. Если абсолютное значение текущего элемента массива меньше, чем элемента с индексом num, то в теле условного оператора присваиваем num индекс текущего элемента.
4. После того, как цикл закончит свою работу, num будет содержать индекс минимального по модулю элемента. Выводим его на экран как есть или увеличиваем на единицу (номер элемента равен индексу при индексации массива с единицы, и на 1 больше индекса при индексации с нуля).

Данное решение задачи не совсем верное, так как находит только первый минимальный элемент. Однако в массиве их может быть несколько (равных между собой или равных между собой только по модулю). Например, в массе [1, 12, 4, 1, 5, 8, 3] минимальных два элемента: первый и четвертый.

Если стоит задача найти номера всех минимальных по модулю элементов, то алгоритм решения задачи будет иным:

1. Сначала ищется и сохраняется в переменной абсолютное значение минимального по модулю элемента (именно значение, а не индекс).
2. В следующем цикле каждый элемент (его модуль) сравнивается с ранее найденным минимальным и при совпадении номер текущего элемента выводится на экран.

Ниже в примерах решения задачи на языках программирования реализован первый упрощенный алгоритм (находится только первый минимальный по модулю элемент). Здесь приведем вариант решение задачи по второму алгоритму на языке Pascal:

```
const N = 20;
var
arr: array[1..N] of integer;
i: byte;
min: integer;
begin
randomize;
for i:=1 to N do begin
arr[i] := random(30)-15;
```

```
write(arr[i]:4);
end;
 writeln;
min := 100; // хранит минимальное по модулю значение
 for i:=1 to N do
 if abs(arr[i]) < abs(min) then
 min := abs(arr[i]);
writeln('Минимальное по модулю значение: ', min);
write('Номера элементов: ');
for i:=1 to N do
 if abs(arr[i]) = min then
 write(i:3);
writeln;
end.
```

Пример выполнения кода:

```
  10  -8  12   3 -14  -3 -14 -15   9  -7   6  -3  -1 -11  -2 -13   1  -7   8 -10
Минимальное по модулю значение: 1
Номера элементов:  13 17
```

Занятие 7. Массив. Разложить положительные и отрицательные числа по разным массивам

Задача

Случайные числа в диапазоне от -5 до 5 разложить по двум массивам: в одни помещать только положительные, во второй - только отрицательные. Числа, равные нулю, игнорировать. Вывести на экран все сгенерированные случайные числа и элементы обоих массивов.

Пояснение к задаче и алгоритм решения

⓾ Изначально оба массива пусты. Присвоим двум разным переменным-индексам

массивов нули.

- В цикле выполняем следующие действия.
  1. Генерируем случайное число и выводим его на экран.
  2. Если очередное число положительное, то увеличиваем индекс массива на единицу и записываем в соответствующую ячейку это число.
  3. Если же генерируется отрицательное число, то увеличиваем индекс массива отрицательных чисел и записываем в массив число.
- В циклах от первого до последнего элемента выводим на экран содержимое массивов положительных и отрицательных чисел. Их переменные-индексы указывают на количество элементов в массивах.

Исходный код на языке программирования Pascal

```
const
 N = 21;
var
 b,c: array[1..N] of integer;
 num: integer;
 i, j, k: byte;
begin
randomize;
 j := 0;
 k := 0;
 for i:=1 to N do begin
 num := random(11)-5;
 write(num:3);
 if num > 0 then begin
 j := j+1;
 b[j] := num;
 end
 else if num < 0 then begin
 k := k + 1;
 c[k] := num;
end;
 end;
 writeln;
 for i:=1 to j do write(b[i]:3);
 writeln;
 for i:=1 to k do write(c[i]:3);
 writeln;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
 4 -2 -1 3 -4 5 3 0 -2 -2 -2 0 1 -5 -1 2 3 -2 5 0 -5
 4 3 5 3 1 2 3 5
-2 -1 -4 -2 -2 -2 -5 -1 -2 -5
```

Занятие 8. Массив. Поменять местами минимальный и максимальный элементы

Задача
В массиве случайных целых чисел поменять местами минимальный и максимальный элементы.

Пояснение к задаче и алгоритм решения

Эта задача состоит из двух частей:

1. Поиск минимума и максимума (а лучше их индексов).
2. Обмен минимального и максимального элемента местами.

Поиск минимума:

1. Присвоить переменной первый индекс массива.
2. Перебрать все элементы массива в цикле, начиная со второго. Проверить каждый, не меньше ли он элемента под индексом, записанным в переменной п.1. Если это так, то присвоить этой переменной текущий индекс.

Поиск максимума выполняется также, только проверяется, не больше ли текущий элемент того, чей индекс хранится в переменной.

Пример обмена местами минимума и максимума массива:

1. Присвоить буферной переменной значение минимума.
2. Записать по индексу минимума максимум массива.
3. Записать по индексу максимума значение, хранимое в буферной переменной.

Исходный код на языке программирования Pascal

```
const N = 15;
var
 arr: array[1..N] of integer;
min, max, i: byte;
 b: integer;
begin
 randomize;
 for i:=1 to N do begin
 arr[i] := random(100);
 write(arr[i],' ');
 end;
 writeln;
 min := 1;
 max := 1;
 for i:=2 to N do begin
 if arr[i] < arr[min] then
 min := i;
 if arr[i] > arr[max] then
 max := i;
 end;
 writeln('arr[',min,']=',arr[min],' arr[',max,']=',arr[max]);
 b := arr[min];
 arr[min] := arr[max];
 arr[max] := b;
 for i:=1 to N do
 write(arr[i],' ');
 writeln;
end.
```

Пример(ы) выполнения программы на языке Pascal

```
94 64 42 1 72 3 88 52 3 67 32 15 31 32 65
arr[4]=1 arr[1]=94
1 64 42 94 72 3 88 52 3 67 32 15 31 32 65
```