

## Кубики и массивы. Самостоятельная работа

С. П. Жолондевская,  
учитель информатики гимназии г. Дрогичина

Без организации самостоятельной работы учащихся на учебных занятиях невозможно создать условия для формирования умений и усвоения теоретических знаний. Так как самостоятельные работы позволяют учащимся не только повысить свой уровень обученности, но и развить критичность и гибкость мышления, сформировать компетенции, которые помогут добиться успеха в жизни, поэтому на уроках информатики используются разные самостоятельные работы с разными целями. Одним из видов самостоятельной работы является решение задач (по теме «Основы алгоритмизации и программирования»).

Тема «Основы алгоритмизации и программирования» является одной из самых сложных в школьном курсе информатики, поэтому при преподавании необходимо искать методы и приемы, которые будут способствовать ее «упрощению».

Кто из нас не играл в детстве в кубики, не строил из них дворцы и домики? Всем знакомый принцип составления целого из частей потом в усложненном виде применяется и при составлении мозаики. Почему бы его не использовать в IX классе при решении задач с использованием одномерных массивов по теме «Основы алгоритмизации и программирования»? Как это сделать? Очень просто. Почти все задачи с использованием одномерных массивов можно разбить на части (блоки, кубики), которые будут в той или иной комбинации встречаться в решении. Для своих учащихся я выделила следующие блоки:

<b>Ввод элементов массива с помощью клавиатуры</b>	
<b>1</b>	<i>Writeln('ww. razm. massiva <b>n</b>'); Readln (<b>n</b>);</i> <i>For i:=1 to <b>n</b> do</i> <i>  Begin</i> <i>    Write ('ww. x['i,']=');</i> <i>    Readln (x[i]);</i> <i>  End;</i>
<b>Ввод элементов массива с помощью функции генерации случайных чисел</b>	
<i>Элементы массива действительные числа</i>	
<b>2</b>	<i>Writeln('ww. razm. massiva <b>n</b>'); Readln (<b>n</b>);</i> <i>Randomize;</i> <i>For i:=1 to <b>n</b> do</i> <i>  x[i]:=random(100)+random;</i>
<i>Элементы массива целые числа</i>	
<b>3</b>	<i>Writeln('ww. razm. massiva <b>n</b>'); Readln (<b>n</b>);</i> <i>Randomize;</i> <i>For i:=1 to <b>n</b> do</i> <i>  x[i]:=round(random(100)+random);</i>

Примечание. В функции *random(100)* аргумент 100 указывает на то, что в таблицу будут записаны числа из диапазона от 0 до 99.

<b>Вывод элементов массива в одну строку</b>	
<b>4</b>	<i>For i:=1 to n do Write (x[i], ', '); Writeln;</i>
<b>Вывод элементов массива в один столбец</b>	
<b>5</b>	<i>For i:=1 to n do Writeln (x[i]);</i>

А далее учащимся предлагаются блоки с решениями ключевых задач, таких как нахождение суммы, произведения, среднего арифметического значения элементов массива, максимального и минимального элементов, количества элементов с заданными свойствами, приведенные в таблице ниже.

<b>Нахождение суммы элементов массива</b>	
<b>6</b>	<i>S:=0; For i:=1 to n do S:=s+x[i];</i>
<b>Нахождение произведения элементов массива</b>	
<b>7</b>	<i>P:=1; For i:=1 to n do P:=P*x[i];</i>
<b>Нахождение среднего арифметического элементов массива</b>	
<b>8</b>	<i>S:=0; For i:=1 to n do S:=s+x[i]; Sr:=s/n;</i>
<b>Нахождение максимального элемента массива и его местоположение в массиве</b>	
<b>9</b>	<i>max:=x[1]; nommax:=1; For i:=2 to n do If max&lt;=x[i] then begin max:=x[i]; nommax:=i end;</i>
<b>Нахождение минимального элемента массива и его местоположение в массиве</b>	
<b>10</b>	<i>min:=x[1]; nommin:=1; For i:=2 to n do If min&gt;=x[i] then begin min:=x[i]; nommin:=i end;</i>
<b>Нахождение количества элемента массива с заданными свойствами</b>	
<b>11</b>	<i>K:=0; For i:=1 to n do If x[i]{<i>характеристика элемента</i>} then K:=k+1;</i>

## Замена элементов массива с заданными свойствами

**12**    *For i:=1 to n do*  
           *If условие then*  
               *X[i]:=...;*

Теперь, имея такие блоки, остается только разбить исходную задачу на подзадачи, подобрать и выстроить в необходимой последовательности те, которые подходят. При этом следует помнить, что почти любое решение задачи строится в соответствии со следующим алгоритмом:

1. Ввод исходных данных.
2. Собственно решение задачи.
3. Вывод полученных результатов.

Например, дана задача «Класс участвовал в соревнованиях по бегу на короткие дистанции. Данные каждого участника заносились в линейную таблицу. Помогите судье найти суммарный результат класса и номера и результаты самого худшего и самого лучшего бегунов класса»

### План решения задачи

1. Ввод элементов массива (блок 1)	<b>1</b> <i>Writeln ('vv. razm. massiva n'); Readln (n);</i> <i>For i:=1 to n do</i> <i>  Begin</i> <i>    Write ('vv. x[', i, ']=');</i> <i>    Readln (x[i]);</i> <i>  End;</i>
2. Нахождение суммы элементов массива (блок 6 + вывод полученного результата)	<b>6</b> <i>S:=0;</i> <i>For i:=1 to n do S:=s+x[i];</i>
3. Нахождение номера и значение максимального элемента массива (блок 9 + вывод полученных результатов)	<b>9</b> <i>max:=x[1]; nommax:=1;</i> <i>For i:=2 to n do</i> <i>  If max&lt;=x[i] then</i> <i>    begin</i> <i>      max:=x[i];</i> <i>      nommax:=i</i> <i>    end;</i>
4. Нахождение номера и значение минимального элемента массива (блок 10 + вывод полученных результатов)	<b>10</b> <i>min:=x[1]; nommin:=1;</i> <i>For i:=2 to n do</i> <i>  If min&gt;=x[i] then</i> <i>    begin</i> <i>      min:=x[i];</i> <i>      nommin:=i</i> <i>    end;</i>
5. Оформление программы, в том числе описание используемых переменных и массива в разделе описания переменных	

В результате может получиться, например, такой вариант решения:

```

Program primer1;
Var n,i:integer;
      X:array [1..25] of real;
```

```

S, max, min: real;
Nommax, nommin: integer;
Begin
  Writeln ('ww. kol-vo n=');
  Readln (k);
  For i:=1 to n do
    Begin
      Write ('ww. X['i,']=');
      Readln (X[i]);
    End;
  S:=0;
  For i:=1 to n do
    S:=s+X[i];
  Writeln ('s=',s:9:2);
  min:=X[1]; nommin:=1;
  For i:=2 to n do
    If min>=X[i] then
      begin
        min:=X[i];
        nommin:=i
      end;
  Writeln ('nommin=', nommin);
  Writeln ('min=', min:9:2);
  max:=X[1]; nommax:=1;
  For i:=2 to n do
    If max<=X[i] then
      begin
        max:=X[i];
        nommax:=i
      end;
  Writeln ('nommax=', nommax);
  Writeln ('max=', max:9:2);
  Readln()
End.

```

The diagram consists of four green brackets on the right side of the code, each pointing to a specific block of code. The first bracket groups the first three lines of the inner loop (Write, Readln, End) and is labeled with a box containing the number 1. The second bracket groups the four lines of the summing loop (S:=0, For i:=1 to n do, S:=s+X[i], Writeln) and is labeled with a box containing the number 6. The third bracket groups the four lines of the minimum-finding loop (For i:=2 to n do, If min>=X[i] then, begin, min:=X[i]; nommin:=i, end;) and is labeled with a box containing the number 10. The fourth bracket groups the four lines of the maximum-finding loop (For i:=2 to n do, If max<=X[i] then, begin, max:=X[i]; nommax:=i, end;) and is labeled with a box containing the number 9.

Аналогичным образом получают решения остальных задач по теме «Основы алгоритмизации и программирования» школьного курса «Информатика. IX класс». После того как учащиеся научатся собирать решения из блоков, им можно показать пути оптимизации полученных решений.

На первом этапе мои учащиеся записывали каждый блок на отдельном листочке, а потом при решении задач выбирали подходящие листочки и выкладывали их на парте в необходимом порядке. Позже при изучении темы «Основы алгоритмизации и программирования», удивили меня тем, что как только им была задана задача, в которой требовалась работа с массивами, достали листочки и стали «собирать» решение. В итоге с заданием все справились успешно.

Во время самостоятельной работы по решению задач происходит активное закрепление изученного материала, развивается умение применять дедуктивный метод, формируются алгоритмический стиль мышления, умения записывать программы, работать с компилятором.