

Правила оформления программ

Этот список правил поможет вам оформлять свои программы красиво и правильно. За несоблюдение данных правил решение может не засчитываться и получать статус «Отклонено» или «Нарушение правил оформления» в тестирующей системе. Список в будущем может пополняться новыми правилами. Текущая версия была создана 15 ноября 2015 года.

I. Блоки

1.1. Каждый оператор языка программирования записывается на отдельной строке.

Неправильно	Правильно
<pre>a := 3; write(a); b := b div 100;</pre>	<pre>a := 3; write(a); b := b div 100;</pre>

1.2. Содержимое конструкций `begin-end`, `repeat-until`, `case-end`, `record-end` выделяется отступом в виде некоторого количества дополнительных пробелов в начале строки или табуляции.

Неправильно	Правильно
<pre>if a <> 3 then begin a := 3; end; repeat a := 4; until a = 4;</pre>	<pre>if a <> 3 then begin a := 3; end; repeat a := 4; until a = 4;</pre>

1.3. Содержимое конструкций `if-then`, `if-then-else`, `while-do`, `with-do` допускается писать в одну строку, если суммарная длина строки остается приемлемой. В случае, если содержимое перенести необходимо, то его можно вынести в отдельную строку, дополнив отступом.

Неправильно	Правильно
<pre>if (a = 3) and (b = 4) and (c = 5) and (d = 7) then c := -13;</pre>	<pre>if a = 3 then b := 4 else b := -1; if (a = 3) and (b = 4) and (c = 5) and (d = 7) then c := a + b + c + d else d := -11;</pre>

1.4. Конструкцию `if-then-else` можно писать в две части, выделяя действие в случае невыполнения условия (`else`) на отдельную строку, но тогда `else` должен выделяться дополнительным отступом. Также по предыдущему пункту можно писать всю конструкцию в одну строку или в четыре, но никак иначе.

Неправильно	Правильно
<pre>if a = 3 then b := 4 else if a = 4 then b := 5 else b := 6;</pre>	<pre>if a = 3 then b := 4 else c := 5; if a = 3 then b := 4 else if a = 4 then b := 5 else b := 6</pre>

- 1.5. Если в любую конструкцию (кроме if-then-else) вложен begin-end, то begin-end можно писать на том же уровне вложенности, что и конструкцию, в которую он вложен. Более того, допускается не переносить begin на отдельную строку.

Неправильно	Правильно
<pre>if a = 3 then begin b := 3 end else begin b := 5; end;</pre>	<pre>for i := 1 to 5 do begin write(i); end; for i := 1 to 5 do begin write(i); end; if a = 3 then begin a := 4; end;</pre>

- 1.6. Содержимое секций uses, type, const, var должно выделяться отступом. Сами названия секций при этом должны идти без отступа в отдельной строке.

Неправильно	Правильно
<pre>const year = 2014; var a: longint; b: integer;</pre>	<pre>const year = 2014; var a: longint; b: integer;</pre>

- 1.7. Выбранный вид отступа должен быть единым для всей программы: не допускается менять ширину отступа в пробелах в середине программы, а также менять табуляцию на пробелы и наоборот.

Неправильно	Правильно
<pre>var a: integer; begin a := 3; if a = 3 then a := 4 else a := 5; end.</pre>	<pre>var a: integer; begin a := 3; if a = 3 then a := 4 else a := 5; end.</pre>

II. Пробелы внутри строк

- 2.1. Любая бинарная операция (<, >, <=, >=, <>, =, +, -, *, div, mod, /, and, or, xor, и т.д.) должна выделяться пробелом как слева, так и справа от себя. В том числе это относится и к знаку присваивания :=.

Неправильно	Правильно
<pre>if (a mod 3=b div 7)and c then d:=a*3 + b*7;</pre>	<pre>if not(a mod 3 = b div 7) and c then d := a * 3 + b * 7;</pre>

- 2.2. Однострочный комментарий может идти в той же строке, что и код, но при этом он обязан отделяться от кода двумя пробелами. Во всех иных случаях комментариев не может идти в той же строке, что и код.

Неправильно
<pre>{ Here is the code: } a := b + c; do_stuff; // Do we really need these spaces?</pre>
Правильно
<pre>minute := time mod 60; { Getting current hour from time } hour := time div 60; // Getting current hour from time // Here we can comment as we want { Even that way }</pre>

- 2.3. Перед символами «:», «,» и «;» никогда не должен идти пробел, и всегда должен идти после (если это не конец строки).

Неправильно	Правильно
<pre>var a,b,c :integer; begin read(a,b,c); end.</pre>	<pre>var a, b, c: integer; begin read(a, b, c); end.</pre>

- 2.4. Запрещается использовать в строке более одного пробела подряд, если эти пробелы не составляют отступ и не служат для отделения комментария от кода.

Неправильно	Правильно
<pre>a := (b + 3) div 17;</pre>	<pre>a := (b + 3) div 17;</pre>

- 2.5. Ключевые слова array, case, do, downto, else, file, if, in, of, then, to, while, with, until всегда должны содержать по обе стороны от себя хотя бы один пробельный символ (допускается перевод строки).

Неправильно	Правильно
<pre>if(a in[3..5])then b := 5;</pre>	<pre>if (a in [3..5]) then b := 5;</pre>

- 2.6. Запрещено отделять знак «..» (две точки) от соседних лексем слева и справа при объявлении диапазонов.

Неправильно	Правильно
<pre>if (a in [3 .. 5]) then b := 5;</pre>	<pre>if (a in [3..5]) then b := 5;</pre>

2.7. При обращении к элементу массива пробелы между квадратными скобками и их содержимым не ставятся. Также не ставится пробел между именем переменной и открывающейся квадратной скобкой.

Неправильно	Правильно
<code>a [3] := -17;</code>	<code>a[3] := -17;</code>

III. Идентификаторы

3.1. В программе запрещается использование кириллицы (даже в комментариях).

Неправильно
<code>a := getInfo(obj); { Получение информации об объекте }</code>
Правильно
<code>a := getInfo(obj); { Getting info from object }</code>

3.2. В программе запрещается использование транслита.

Неправильно
<code>a := poluchitInformatsiyu(obj); { Poluchenie informatsii ob ob'ekte }</code>
Правильно
<code>a := getInfo(obj); { Getting info from object }</code>

3.3. В качестве имен простых переменных/констант, смысл которых достаточно очевиден, разрешается использовать короткие название из одной-двух букв. При этом, по сложившейся традиции, имена счетчиков в циклах обычно называют *i* и *j*. Для нетривиальных понятий лучше использовать английские слова.

Неправильно
<code>const q = 50; var a: array [1..q] of 2..5;</code>
Правильно
<code>const number_of_students = 50; var marks: array [1..number_of_students] of 2..5;</code>

3.4. Не стоит злоупотреблять длинными именами переменных и констант. 10-15 символов должно быть достаточно для передачи смысла переменной.

Неправильно
<code>var amount_of_cash_on_user_account: longint;</code>
Правильно
<code>var money: longint;</code>

3.5. Имена переменных и констант должны содержать только строчные латинские буквы, цифры и нижние подчеркивания для разделения слов в названии.

Неправильно	Правильно
<pre>var N: integer; AccountLogins: array [1..100] of string;</pre>	<pre>var n: integer; account_logins: array [1..100] of string;</pre>

3.6. Имена функций и процедур именуются с использованием CamelCase, то есть содержат только латинские буквы и цифры, каждое слово, кроме первого, пишется с заглавной буквы.

Неправильно
<pre>function Square(x: double): double; begin Square := x * x end; function get_length(x1, y1, x2, y2: double): double; begin get_length := sqrt(square(x1 - x2) + square(y1 - y2)); end;</pre>
Правильно
<pre>function square(x: double): double; begin square := x * x end; function getLength(x1, y1, x2, y2: double): double; begin getLength := sqrt(square(x1 - x2) + square(y1 - y2)); end;</pre>

3.7. Все ключевые слова языка должны записываться только строчными латинскими буквами.

Неправильно	Правильно
<pre>If a mod 2 = 0 Then writeln('Even') eLsE writeln('Odd');</pre>	<pre>if a mod 2 = 0 then writeln('Even') else writeln('Odd');</pre>

IV. Функции и процедуры

4.1. Каждая функция обязана возвращать значение.

Неправильно	Правильно
<pre>function getSum(a, b: integer): integer; begin write(a + b); end;</pre>	<pre>procedure writeSum(a, b: integer); begin write(a + b) end; function getSum(a, b: integer): integer; begin getSum := a + b end;</pre>

4.2. Каждая процедура и функция должна содержать перед и после себя ровно одну пустую строку.

Неправильно	Правильно
<pre> program checkEven; var n: integer; function isEven(n: integer): boolean; begin isEven := n mod 2 = 0; end; begin read(n); if isEven(n) then writeln('Even') else writeln('Odd'); end. </pre>	<pre> program checkEven; var n: integer; function isEven(n: integer): boolean; begin isEven := n mod 2 = 0; end; begin read(n); if isEven(n) then writeln('Even') else writeln('Odd'); end. </pre>

4.3. В случае, если в функцию или процедуру передается переменная сложного типа (массив, множество и т.п.), она обязательно должна выделяться одним из модификаторов `const` или `var`.

Неправильно
<pre> const array_length = 100000; type myarray = array [1..array_length] of integer; function getMax(d: myarray): integer; </pre>
Правильно
<pre> const array_length = 100000; type myarray = array [1..array_length] of integer; function getMax(const d: myarray): integer; </pre>

4.4. Между списком аргументов и именем функции/процедуры никогда не ставится пробел.

Неправильно	Правильно
<pre> function pow (d, p: integer): integer; begin if p = 0 then pow := 1 else pow := pow (d, p - 1) * d; end; begin writeln (pow (5, 3)); end. </pre>	<pre> function pow(d, p: integer): integer; begin if p = 0 then pow := 1 else pow := pow(d, p - 1) * d; end; begin writeln(pow(5, 3)); end. </pre>

V. Другие правила

5.1. Запрещается использование конструкции `goto`.

5.2. В программе запрещается использование нетривиальных констант без их явного описания в разделе `const`. Обычно нетривиальными считаются любые константы, кроме, быть может, -1, 0, 1, 2, 10.

Неправильно
<pre>var dice: integer; begin randomize; dice := random(6) + 1; writeln('Random dice throw gives you ', dice); end.</pre>
Правильно
<pre>const dice_size = 6; var dice: integer; begin randomize; dice := random(dice_size) + 1; writeln('Random dice throw gives you ', dice); end.</pre>

5.3. Каждая переменная перед использованием обязана быть явно инициализирована присваиванием. В случае массива допускается использование для инициализации `fillchar`.

Неправильно	Правильно
<pre>var c, d: integer; begin c := c + 10; // Undefined value d := d + 5; // Undefined value writeln(c, ' ', d); end.</pre>	<pre>var c, d: integer; begin c := 0; d := 0; c := c + 10; // 10 d := d + 5; // 5 writeln(c, ' ', d); end.</pre>

5.4. Запрещается сравнивать переменные типа `boolean` с `true` или `false`.

Неправильно	Правильно
<pre>if b = true then writeln(1); if b <> true then writeln(2); if b = false then writeln(3); if b <> false then writeln(4);</pre>	<pre>if b then writeln(1); if not b then writeln(2); if not b then writeln(3); if b then writeln(4);</pre>

5.5. Внутри цикла `for` запрещается явное изменение значения счетчика цикла.

Неправильно	Правильно
<pre>pairs := 0; i := 1; for i := 1 to n - 1 do if a[i] = a[i + 1] then begin pairs := pairs + 1; i := i + 1; end;</pre>	<pre>pairs := 0; i := 1; while i < n do begin if a[i] = a[i + 1] then begin pairs := pairs + 1; i := i + 1; end; i := i + 1; end;</pre>

5.6. Запрещено использовать строки длиннее 120 символов. Строки можно разбивать на части в длинных выражениях и списках аргументов, при этом если выражение разбивается на бинарной операции, то саму операцию лучше переносить на отдельную строку. Строки 2, 3, 4, ... одного оператора считается вложенными в первую строку и выделяются отступом.

Неправильно
<pre>if (a mod 3 <> b mod 3) and (b mod 3 <> c mod 3) and (c mod 3 <> d mod 3) and (b mod 3 ... writeln('One of the numbers a, b, c, d is negative :'));</pre>
Правильно
<pre>if (a mod 3 <> b mod 3) and (b mod 3 <> c mod 3) and (c mod 3 <> d mod 3) and (b mod 3 <> d mod 3) and (a mod 3 <> c mod 3) and (a mod 3 <> d mod 3) then writeln('One of the numbers a, b, c, d is negative :');</pre>

VI. Пример хорошей программы

```
1  program sample;
2
3  const
4      max_fact = 12;
5
6  var
7      k, n, i, j, ind, l, ans: longint;
8      a, b: array [0..max_fact] of longint;
9
10 function factorial(a: longint): int64;
11 var
12     i, j: longint;
13 begin
14     j := 1;
15     for i := a downto 2 do
16         j := j * i;
17     factorial := j;
18 end;
19
20 begin
21     readln(n, k);
22     ind := n;
23     for i := 1 to k do
24         read(a[i]);
25     for i := 1 to n do
26         b[i] := i - 1;
27     ans := 1;
28     for i:=1 to k do
29         begin
30             ans := ans + factorial(n - i) div factorial(n - k) * b[a[i]];
31             for j := a[i] + 1 to n do
32                 dec(b[j]);
33             end;
34         writeln(ans);
35     end.
```