

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 46, июнь 2020 г.

011000100010101001001111100100011001001000101001001001011

Школа программирования

Редакцию продолжает публикацию фрагментов книги Дмитрия Михайловича Златопольского «Программирование для школьников», которая готовится к изданию в издательстве «Ленанд».

2.2. Варианты действий в программах. Условные операторы

До этого были рассмотрены программы, в которых действия выполнялись последовательно одно за другим, в соответствии с записанными в них операторами. Как правило, в программах возможны ситуации, когда на каком-то этапе требуется выполнение не строго определенного действия, а одного из двух (или более) вариантов действия, при этом выбор того или иного варианта зависит от некоторого условия. Возможны также ситуации, когда некоторое действие должно выполняться не всегда, а только при определенном условии. Рассмотрим особенности каждого случая.

2.2.1. Два варианта действий

Пример задачи. Дано целое число. Определить, является ли оно четным.

Решение

Начальная часть программы решения задачи:

```
вывод нс, "Введите целое число "  
ввод n
```

Далее возможны два варианта действий (два варианта оператора вывода информации на экран):

```
1)  
вывод нс, "Это число четное"
```

```
2)  
вывод нс, "Это число нечетное"
```

В таких случаях (возможны два варианта действий) в программе необходимо использовать так называемый «условный оператор» в следующей форме:

– на школьном алгоритмическом языке:

```
если <условие>  
то  
    <Действия 1-го варианта (1-я серия операторов)>  
иначе
```

<Действия 2-го варианта (2-я серия операторов)>

все

– на языке Паскаль¹:

```
if <условие>
then
  begin
    <Действия 1-го варианта (1-я серия операторов)>
  end { обратим внимание на то, что символ ; не записывается! }
else
  begin
    <Действия 2-го варианта (2-я серия операторов)>
  end
```

– на языке Python:

```
if <условие>:
  <Действия 1-го варианта (1-я серия операторов)>
  # Записываются с отступом
else:
  <Действия 2-го варианта (2-я серия операторов)>
  # Записываются с отступом
```

Такую форму оператора называют «полный условный оператор».

В самом простом случае *<условие>* – это два арифметических выражения, между которыми записан знак операции сравнения².

Во всех языках программирования есть 6 операций сравнения:

Таблица 2.2

	Знак операции	Означает
1	<	Меньше
2	<=	Меньше либо равно
3	>	Больше
4	>=	Больше либо равно
5	Зависит от языка программирования	Равно
6	То же	Не равно

Примеры условий:

```
a < 0
x <= a + b
(a + b) / 2 > 0
2 * c >= 7 * d - 1
```

В языке Python:

```
N % 5 == 3
v != 100
```

Каждое условие может соблюдаться (быть истинным), например таким является условие $a < 0$ при значении переменной a , равном -12 , или не соблюдаться (быть ложным), и после выполнения в программе операции сравнения получается соответствующий результат

¹ Если в каком-то варианте действий выполняется один оператор (а не их не серия), то служебные слова **begin** и **end** не записываются. Заметим также, что конструкция в виде двух указанных служебных слов называется «составной оператор». В составном операторе может присутствовать любое число других операторов, в том числе составные.

² Операции сравнения называют также «операциями отношения», а знаки этих операций – «операторами сравнения» или «операторами отношения».

логического типа (True/Истина/да или False/Ложь/нет). В дальнейшем мы будем использовать результат операций сравнения в виде слов Истина или Ложь.

Схема работы полного условного варианта показана на рис. 2.5³:

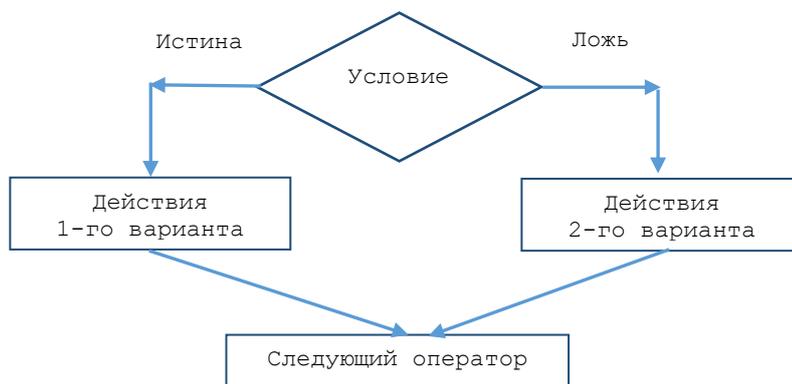


Рис. 2.5

Применительно к рассмотренному примеру оператор оформляется так:

```
если  $n \bmod 2 = 0^4$  | Если остаток от деления числа  $n$  на 2 равен нулю
  то
    вывод нс, "Это число четное"
  иначе
    вывод нс, "Это число нечетное"
все
```

Можно также использовать переменную ост, соответствующую остатку от деления числа n на 2, и предварительно рассчитать её значение:

```
ост :=  $n \bmod 2$ 
если ост = 0
  то
    вывод нс, "Это число четное"
  иначе
    вывод нс, "Это число нечетное"
все
```

Возникает вопрос: «А можем ли мы считать первым вариантом действий вывод на экран сообщения о том, что заданное число – нечетное?». – Конечно, можем:

```
если ?
  то
    вывод нс, "Это число нечетное"
  иначе
    вывод нс, "Это число четное"
все
```

Но при этом условие, используемое в операторе, должно быть уточнено – оно должно быть *истинным* для случая выполнения в программе *действий первого варианта* (см. рис. 2.5), то есть таким – $n \bmod 2 = 1$:

```
если  $n \bmod 2 = 1$ 
  то
    вывод нс, "Это число нечетное"
```

³ Приведенную схему называют «развилка» или «ветвление», а части оператора со служебными словами **если/if** и **иначе/else** – «ветви».

⁴ Напомним (см. раздел 2.1), что знак mod в программе на школьном алгоритмическом языке используется условно.

```
    иначе
        вывод нс, "Это число четное"
все
```

Самостоятельно убедитесь в том, что для обоих способов оформления оператора результат, согласно схеме на рис. 2.5, будет одним и тем же как для четных, так и для нечетных значений числа n .

Как видно из схемы на рис. 2.5 и согласно общей форме полного условного оператора (см. выше), как в «ветви **если**/**if**, так и в ветви **иначе**/**else** может быть указан не один оператор, а их серия.

Пример задачи. Даны два вещественных числа a и b . Если первое больше второго, то увеличить каждое число в 2 раза, иначе – уменьшить в два раза.

Соответствующая программа:

```
вывод нс, "Введите число a"
ввод а
вывод нс, "Введите число b"
ввод b
если а > b
    то
а := а * 2
b := b * 2
иначе
а := а/2
b := b/2
все
| Новые значения чисел
вывод нс, "a = ", а
вывод нс, "b = ", b
```

Контрольные вопросы

1. В каких случаях в программе используется полный условный оператор? Как он оформляется? Как он «работает»? (Что происходит при его выполнении?) Нарисуйте графическую схему выполнения.
2. Что представляет из себя условие, записываемое в условном операторе в простейшем случае? Какие знаки операций сравнения (отношения) могут использоваться в нем?
3. Что является результатом операции сравнения?

Прежде чем предлагать задания для самостоятельной работы, заметим, что когда возможны два варианта вывода информации на экран, в которых имеются одинаковые фрагменты, например:

```
ост := n mod 2
если ост = 0
    то
        вывод нс, "Это число четное"
    иначе
        вывод нс, "Это число нечетное"
все
```

– можно общую часть выводить отдельно:

```
ост := n mod 2
вывод нс, "Это число "
если ост = 0
    то
        вывод "четное"
```

```

иначе
  вывод "нечетное"
все

```

Задачи для самостоятельной работы

1. Определите (не оформляя программу) значение переменной c после выполнения следующего фрагмента программы:

```

a := 40
b := 10
b := -a/2 * b
если a < b
  то
    c := b - a
  иначе
    c = a - 2 * b
все

```

2. Определите значение переменной a после выполнения следующего фрагмента алгоритма⁵:

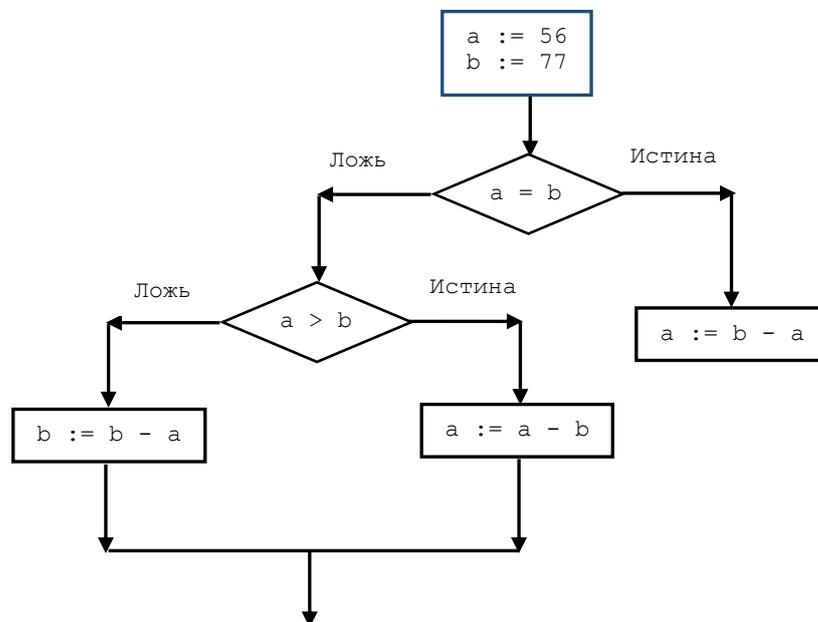


Рис. 2.6

3. Даны два различных вещественных числа. Напишите программу, которая определяет:

- какое из них больше;
- какое из них меньше.

4. Напишите программу, которая определяет, является ли число a делителем числа n .

5. Напишите программу, которая определяет, в какую из областей – **I** или **II** (рис. 2.7) – попадает точка с заданными координатами. Для простоты принять, что точка не попадает на границу областей.

⁵ Одним из способов записи алгоритма является оформление так называемых «блок-схем». На рис. 2.6 приведен фрагмент блок-схемы.

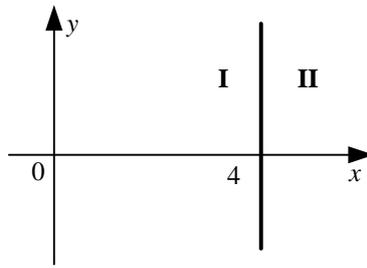


Рис. 2.7

6. Напишите программу, в которой рассчитывается значение y при заданном значении x :

$$y = \begin{cases} \sin^2 x & \text{при } x > 0, \\ 1 - 2\sin x^2 & \text{в противном случае.} \end{cases}$$

7. Даны радиус круга и сторона квадрата. Напишите программу, которая определяет, у какой фигуры площадь больше.

8. Напишите программу, которая решает задачу: «Дано целое число n . Вывести на экран следующее за ним четное число».

9. Даны коэффициенты a , b и c квадратного уравнения $ax^2 + bx + c = 0$ (a не равно 0). Напишите программу, которая определяет, имеет ли это уравнение корни или нет (сами корни, если они есть, вычислять не нужно).

2.2.2. Один, но не обязательный вариант действий

Пример задачи. Даны три целых числа, среди которых есть отрицательные. Вывести на экран отрицательные числа на одной строке с пробелом между ними.

Решение

Используем в программе три переменные – a , b и c .

Начальная часть программы решения задачи:

```
вывод нс, "Введите 3 целых числа (как минимум одно – отрицательное)"
```

```
ввод  $a$ ,  $b$ ,  $c$ 
```

```
вывод нс, "Среди них отрицательные: "
```

Далее – в каком случае выводить значение первого числа a ? Только если соблюдается условие $a < 0$. Аналогично и для двух других чисел. В таких случаях (когда какие-то действия в программе выполняются не всегда, а только при определенном условии) используется условный оператор в следующей форме:

```
если <условие>
то
    <Действия (серия операторов)>
все
```

Такой вариант условного оператора называют «неполный условный оператор». В программе на языке Паскаль неполный условный оператор в общем случае оформляется так⁶:

```
if <условие> then
    begin
        <Действия (серия операторов)>
    end
```

⁶ Если выполняется один оператор (а не их не серия), то служебные слова **begin** и **end** не записываются.

– на языке Python:

```
if <условие>:  
<Действия (серия операторов)> # Записываются с отступом
```

Применительно к рассмотренному примеру завершающая часть программы решения задачи оформляется так:

```
если a < 0  
  то  
    вывод a, " "  
все  
если b < 0  
  то  
    вывод b, " "  
все  
если c < 0  
  то  
    вывод c  
все
```

Схема работы неполного условного оператора показана на рис. 2.8⁷:

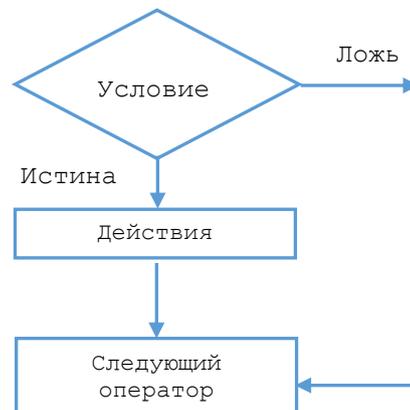


Рис. 2.8

Еще раз обратим внимание на то, что нет альтернативных вариантов действий, как в случае, рассмотренном в пункте 2.2.1, а есть единственно возможный, но не обязательный, вариант.

Контрольные вопросы

В каких случаях в программе используется неполный условный оператор? Как он оформляется? Как он «работает»? (Что происходит при его выполнении?) Нарисуйте графическую схему выполнения.

Задачи для самостоятельной работы

10. Даны три целых числа. Вывести на экран те из них, которые являются четными.
11. Дано вещественное число. Вывести на экран его абсолютную величину (условно принимая, что соответствующей стандартной функции нет). Полный условный оператор не использовать.
12. Даны два различных вещественных числа. Определить наибольшее и наименьшее из них, не используя полный условный оператор, а применив:
 - а) два неполных условных оператора;
 - б) один неполный условный оператор.

⁷ Приведенную схему называют «обход».

Более сложные вопросы, связанные с вариантами действий в программах, будут рассмотрены в разделах 3.3 и 3.4.

«Высший пилотаж»

Когда в программе возможно присваивание переменной одного из двух значений, то вместо полного условного оператора:

```
если <условие>
  то
    а := <значение1>
  иначе
    а := <значение2>
все
```

можно использовать такой, более краткий, фрагмент:

```
а := <значение2> | Пока условно
если <условие>
  то | Меняем значение а
    а := <значение1>
все
```

2.3. Повторение действий в программе. Операторы цикла

Как правило, в программах приходится организовывать повторение одних и тех же действий⁸. Для этого используются так называемые «операторы цикла»⁹ – операторы, обеспечивающие повторения одних и тех же или аналогичных действий. Существует несколько их разновидностей. Рассмотрим особенности каждого вида.

2.3.1. Оператор цикла с параметром

Он применяется в тех случаях, когда в программе какие-то действия повторяются и при этом некоторая величина меняется с постоянным шагом¹⁰.

Пример задачи. Вывести «столбиком» все целые числа от 20 до 30 включительно.

Решение

Без использования оператора цикла в программе потребовалось бы записать:

```
вывод нс, 20
вывод нс, 21
...
вывод нс, 30
```

Видно, что есть повторяющиеся действия (вывод на экран числа) и что при этом выводимое число меняется с постоянным шагом (равным 1). Можно применить оператор цикла с параметром.

Общий вид этого оператора:

– в школьном алгоритмическом языке:

```
нц для <параметр цикла> от <начальное значение> до <конечное значение>
  <тело оператора цикла>
кц
```

⁸ Вспомните «некомпьютерную» задачу о переправе солдат в главе 1.

⁹ Кратко их называют «циклами». «Циклом» называют также повторение действий в программе.

¹⁰ В языке программирования Python этот оператор применяется и в других случаях.

– в языке Паскаль:

```
for <параметр цикла> := <начальное значение> to <конечное значение> do  
  <тело оператора цикла>
```

где <параметр цикла> – имя величины, являющейся параметром цикла (величины, меняющейся при повторении действий);

– <начальное значение> – начальное значение этой величины;

– <конечное значение> – то же, конечное; предполагается, что <конечное значение> больше <начальное значение> и что изменение от <конечное значение> до <начальное значение> происходит с шагом, равным 1;

– <тело оператора цикла> – операторы, повторяющиеся при работе оператора цикла¹¹.

В рассматриваемом примере оператор оформляется следующим образом:

```
нц для а от 20 до 30 | а – имя величины-параметра цикла  
  вывод нс, а  
кц
```

и (на языке Паскаль)

```
for а := 20 to 30 do writeln(a)
```

В языке программирования Python общий вид оператора цикла с параметром такой:

```
for <параметр цикла> in <набор значений>:  
  <тело оператора цикла> # Записывается с отступом
```

где <набор значений> – набор значений параметра цикла, для которых проводится повторение действий.

Так как в рассматриваемом примере выводимое число меняется от 20 до 30 с шагом 1, в качестве <набор значений> можно использовать функцию `range()`. Эта функция возвращает набор целых чисел, образующих арифметическую прогрессию. Например, если нужно получить числа 0, 1, 2, ..., 6, то следует так оформить вызов функции:

```
range(7)12
```

а в общем случае – для получения n последовательных целых чисел, начиная с 0:

```
range(n)
```

Часто нужно, чтобы набор из n последовательных целых чисел начинался с 1. В таких случаях функция `range` оформляется так:

```
range(1, n + 1)
```

Обратим внимание на то, что в последнем случае интервал чисел задается двумя значениями – начальным и конечным, причем конечное значение не используется.

В принципе, первый член прогрессии может быть любым целым числом (в том числе отрицательным). Если он равен a , а последний член прогрессии – b , то при $b > a$ функция `range()` должна быть оформлена так:

```
range(a, b + 1)
```

а если $b < a$, то так:

¹¹ В программе на языке Паскаль в случае, когда в теле цикла выполняются более одного оператора, следует использовать составной оператор (см. сноску в разделе 2.2).

¹² Обратите внимание – записывается число 7, хотя нужно получить максимальное число 6.

`range(a, b - 1, -1)`¹³

Примечание. В последнем случае указан шаг изменения значений – он равен -1 , а в первом по умолчанию шаг принят равным 1 .

Из сказанного следует, что для решения обсуждаемой задачи вывода чисел оператор цикла оформляется так:

```
for a in range(20, 31):  
    print(a)
```

С помощью функции `range` можно также получить набор чисел, шаг изменения которых отличается от 1 и -1 (но он тоже должен быть целым числом).

Схема, иллюстрирующая работу оператора цикла с параметром в школьном алгоритмическом языке и языке Паскаль, выглядит следующим образом:

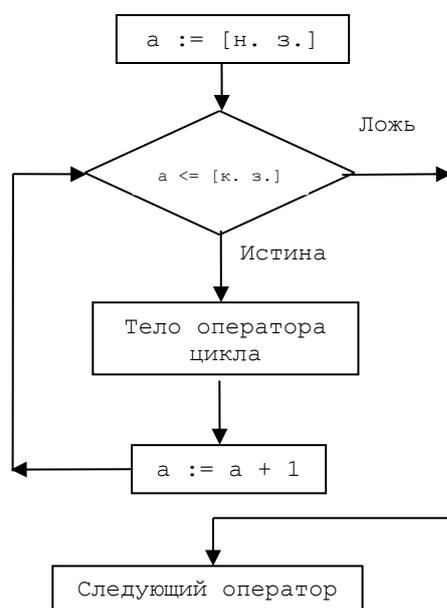


Рис. 2.9

На схеме буквой a обозначено имя величины-параметра цикла, $[н. з.]$ и $[к. з.]$ – соответственно начальное и конечное значения параметра.

В некоторых языках программирования предусмотрена возможность изменения значения величины-параметра цикла с шагом, отличным от 1 . Например, в языке Паскаль шаг может быть равен -1 , в языке Python – любому целому числу (см. чуть выше), в языке Бейсик – любому числу, в том числе вещественному. В таких случаях при оформлении оператора указывается значение шага¹⁴, а приведенная на рис. 2.9 схема несколько меняется.

Из схемы на рис. 2.9 можно определить, что количество повторений тела инструкции для рассматриваемого случая равно:

$$к. з. - н. з. + 1$$

¹³ Обратите внимание – записывается число $b - 1$, хотя нужно получить минимальное число, равное b .

¹⁴ В языке программирования Паскаль при шаге, равном -1 , в операторе вместо служебного слова `to` должно быть записано `downto`.

Поэтому говорят, что при использовании оператора цикла с параметром количество повторений тела цикла известно. С другой стороны, когда известно количество повторений каких-то действий, в программе удобно использовать именно этот оператор.

Контрольные вопросы

1. Что такое «цикл»?
2. В каких случаях используют оператор цикла с параметром?
3. Что такое «параметр цикла»? Что такое «тело оператора цикла»?

Задания

1. Нарисуйте схему работы оператора цикла с параметром, значение которого меняется:
 - а) от большего начального значения до меньшего конечного с шагом, равным -1 ;
 - б) от меньшего начального значения до большего конечного с шагом, равным 2 .
2. Ответьте, пожалуйста, на вопросы:
 - 1) может ли тело оператора цикла с параметром не выполниться ни разу?
 - 2) сколько раз будет выполняться тело оператора цикла с параметром при начальном (а) и конечном (б) значениях параметра цикла, равных:
 - $a = 1; b = 10$;
 - $a = 10; b = 20$;
 - $a = N; b = M$,если во всех случаях шаг изменения величины-параметра равен 1 ;
 - 3) сколько раз будет выполняться тело оператора цикла с параметром, если начальное значение параметра $a = N$, конечное значение $b = M$, а шаг изменения равен s ($s \neq 1$)? (конечно, в случае, если значение $s \neq 1$ допускается в языке программирования)

СОВЕТ № 2

Когда значения конечного или/и начального значения параметра цикла сразу не очевидны, удобно использовать прием, который мы использовали при обсуждении задачи вывода «столбиком» всех целых чисел от 20 до 30, – выписать (на бумаге) ряд операторов, решающих задачу без использования оператора цикла.

Пример задачи. «Составить программу для расчета факториала натурального числа n (факториал числа n , обозначаемый $n!$, равен $1 \times 2 \times 3 \times \dots \times n$)».

Решение

Прежде всего, заметим, что определить искомое значение так:

```
ввод n
факториал := 1 * 2 * 3 * ... * n
```

мы не можем.

Выпишем операторы, которые решают поставленную задачу:

```
произв := 1 * 2
произв := произв * 3
произв := произв * 4
...
произв := произв * n
```

где *произв* – переменная, накапливающая в себе произведение чисел (искомое значение).

Видно, что есть повторяющиеся действия – умножение текущего значения переменной *произв* на 2, 3, ..., n . Это меняющееся число должно быть принято в качестве параметра цикла (k), а его начальное и конечное значения равны, соответственно, 2 и n . Значит, можем использовать оператор цикла с параметром:

```
ввод n
прозв := 1
```

```
нц для к от 2 до n
  произв := произв * к
кц
вывод нс, произв
```

Внимание! Использованный прием помог нам также установить начальное значение переменной произв (оно равно 1, о чем часто забывают начинающие программисты).

Прежде чем предлагать задачи для самостоятельной работы, обсудим один важный вопрос. Допустим, что требуется вывести на экран следующую информацию:

```
10 11
11 12
...
29 30
```

Видно, что есть повторение действий (вывод на экран пар чисел). На первый взгляд кажется, что меняются две переменные величины, одна из которых — от 10 до 29, а другая — от 11 до 30. А в операторе цикла с параметром может быть только один параметр. Означает ли это, что мы не сможем применить такой оператор? Ответ — можем. Дело в том, одна из меняющихся величин зависит от другой. Если в качестве параметра цикла a принять меняющееся первое число каждой пары, то соответствующий оператор цикла можно оформить в виде:

```
нц для а от 10 до 29
  вывод нс, а, " ", а + 1
кц
```

Можно также в качестве параметра цикла a принять меняющееся второе число. Тогда оператор примет вид:

```
нц для а от 11 до 30
  вывод нс, а - 1, " ", а
кц
```

Задачи для самостоятельной работы

1. Напечатать «столбиком»:

а) все целые числа от 20 до 35;

б) квадраты всех целых чисел от a до 50 (значение a вводится с клавиатуры; $a \leq 50$);

в) кубы всех целых чисел от 10 до b (значение b вводится с клавиатуры; $b \geq 10$);

г) все целые числа от a до b (значения a и b вводятся с клавиатуры; $b \geq a$).

2. Одна штука некоторого товара стоит 20,4 руб. Напечатать таблицу стоимости 2, 3, ..., 20 штук этого товара.

3. Вывести на экран следующую информацию:

```
20 25
21 26
...
34 35
```

3. Напечатать таблицу умножения на 7:

```
1 x 7 = 7
2 x 7 = 14
...
9 x 7 = 63
```

4. Рассчитать значения z для значений a , равных 2, 3, ..., 17:

```
z = 4,3t2 - 8t + 13
t = 3a
```

5. Напечатать таблицу стоимости 50, 100, 150, ..., 1000 г сыра (стоимость 1 кг сыра вводится с клавиатуры).

6. Вывести «столбиком» следующие числа: 2,1, 2,2, 2,3, ..., 2,8.

Самостоятельно установите также, можно ли в языке программирования, который вы используете, изменять значение параметра в теле оператора цикла, а также определите значение параметра после окончания работы оператора. Это важные вопросы, ответы на которые помогут вам избежать ошибок при разработке программ. Заметим, что изменение параметра, а также его начального и конечного значений в теле оператора цикла может привести к непредсказуемым последствиям и считается плохим стилем программирования.

Частным случаем оператора цикла с параметром является вариант, при котором величина-параметр цикла в теле оператора не используется. Например, если требуется, например, повторить какие-то действия 10 раз, то соответствующий фрагмент может быть записан (с учетом схемы на рис. 2.9 и анализа результатов ответов на вопросы в пункте 2 только что приведенных заданий) как:

```
нц для с от 1 до 10
...
кц

или

нц для с от 2 до 11
...
кц
```

или другим подобным способом. Примером такой задачи является задача обработки набора чисел (см. главы 4 и 6). В них для ввода значений чисел из набора может быть использован следующий фрагмент программы:

```
нц для к от 1 до n | n – общее количество чисел в наборе
    вывод нс, "Задайте очередное число набора"
    ввод а | Очередное число
    ... | Обработка заданного значения числа а
кц
```

Видно, что величина *к* в теле оператора не используется – при выполнении оператора она играет роль счетчика числа повторений, поэтому обсуждаемый оператор цикла иногда называют (по мнению автора, ошибочно) — «оператор цикла со счетчиком».

Контрольные вопросы

1. В чем особенность варианта оператора с параметром, который назван чуть выше «частным случаем»?

2. Почему, по вашему мнению, автор считает ошибочным использование названия «оператор цикла со счетчиком» вместо названия «оператор цикла с параметром»?

3. Сколько раз будет выполнено тело частного случая оператора цикла с параметром, заголовок которого:

- а) нц для к от 1 до 20;
- б) нц для а от 120 до 200;
- в) нц для х от а до в; (а и в – целые числа, $v > a$).

Задачи для самостоятельной работы

7. Вывести на экран 12 раз слово «Привет!» с пробелами между словами.

8. Напечатать ряд чисел 20 в виде:

20 20 20 20 20 20 20 20 20 20 .

9. Вывести на экран любое заданное число любое заданное число раз в виде, аналогичном показанному в условии предыдущей задачи.

Оператор цикла с параметром может быть использован в программе для обеспечения некоторой паузы, если в его теле не приводить никаких операторов (применить так называемый «пустой цикл»). Например:

```
нц для с от 1 до 20000
```

```
кц
```

Продолжительность паузы будет зависеть от быстродействия компьютера и от конечного значения параметра цикла. Заметим, что в ряде языков программирования имеются специальные процедуры, обеспечивающие паузу в выполнении программы.

Задание. Подберите на своем компьютере такое конечное значение параметра «пустого цикла», при котором в программе будет пауза длительностью примерно 5 сек.

2.3.2. Оператор цикла с предусловием

Этот оператор используется в тех случаях, когда в программе есть повторения действий, неизвестно, но неизвестно начальное и/или конечное значение меняющейся величины (то есть неизвестно количество повторений), но зато известно, при каком условии надо эти действия повторять.

Пример задачи. Дано натуральное число n . Вывести его цифры в «столбик», начиная с последней цифры.

Решение

Подумаем над вопросом: какую цифру целого числа (см. схему ниже) определить можно?

<i>первая</i>	<i>вторая</i>	<i>...</i>	<i>предпоследняя</i>	<i>последняя</i>
---------------	---------------	------------	----------------------	------------------

цифры числа

Ответ – последнюю (последняя цифра любого натурального числа равна остатку от деления этого числа на 10 – убедитесь в этом!):

```
посл := n mod 10 | посл – последняя цифра заданного числа n
```

(О знаке операции mod – см. раздел 2.1.)

А остальные цифры? Как, например, определить предпоследнюю цифру? Ее можно найти только так: получить число без последней цифры исходного числа (разделив исходное нацело на 10) и для «нового» числа определить последнюю цифру.

Аналогично можно получить и предпредпоследнюю, и остальные цифры.

Следовательно, для решения задачи в программе надо многократно выполнить следующие действия:

- 1) определить последнюю цифру заданного числа n ;
- 2) вывести ее на экран;
- 3) изменить число n , отбросив в нем уже «обработанную» последнюю цифру.

Но сколько раз надо повторить указанные действия? Неизвестно (неизвестно число разрядов в заданном числе n), то есть оператор цикла с параметром использовать нельзя. Надо применить оператор цикла с предусловием.

Его общий вид:

– в школьном алгоритмическом языке:

```
нц пока <условие>
```

```
<тело оператора цикла>
```

кц

– в языке Паскаль:

```
while <условие> do  
  <тело оператора цикла>
```

– в языке Python:

```
while <условие>:  
  <тело оператора цикла>
```

где <условие> – условие, при котором выполняется <тело оператора цикла>.

Схема, иллюстрирующая работу оператора цикла с предусловием, показана на рис. 2.10:

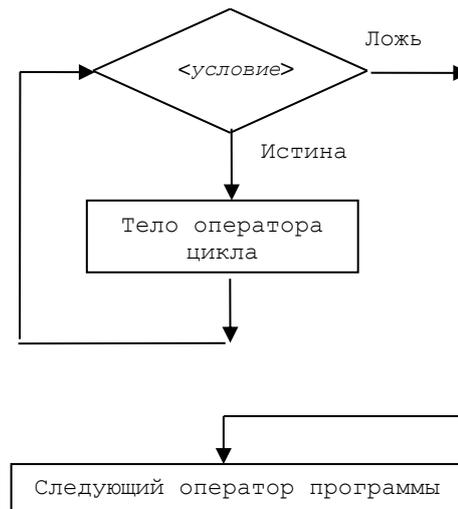


Рис. 2.10

Для рассматриваемого примера на школьном алгоритмическом языке оператор оформляется так:

```
нц пока <условие>  
  посл := n mod 10  
  вывод нс, посл  
  n := n div 10  
кц
```

(О знаке операции div – см. раздел 2.1.)

СОВЕТ № 3

Приведем полезное правило, с помощью которого можно установить условие, записываемое в операторе цикла с предусловием:

- 1) необходимо определить условие, при котором *нельзя* или *не нужно* выполнять повторяющиеся действия;
- 2) записать в операторе условие, противоположное найденному на предыдущем этапе.

В нашем случае мы должны прекратить действия, когда n станет равно нулю, значит, условие, записываемое после служебного слова **пока**, должно быть таким: $n > 0$ ¹⁵ (отрицательным n быть не может). Следовательно, можем оформить фрагмент так:

¹⁵ Конечно, в данном случае это условие можно было определить и не используя приведенное правило. В ряде случаев искомое условие не так очевидно.

```

нц пока n > 0
    посл := n mod 10
    вывод нс, посл
    n := n div 10
кц

```

Еще пример: «Найти наибольший общий делитель (НОД) двух заданных натуральных чисел, используя алгоритм Евклида. Суть алгоритма Евклида можно изобразить схемой:

$$\begin{aligned}
 &= \text{НОД}(a - b, b) \text{ при } a > b \\
 \text{НОД}(a, b) &= \text{НОД}(a, b - a) \text{ при } b > a \\
 &= a \text{ (или } b) \text{ при } a = b
 \end{aligned}$$

Например, согласно схеме:

$\text{НОД}(26, 18) = \text{НОД}(8, 18) = \text{НОД}(8, 10) = \text{НОД}(8, 2) = \text{НОД}(6, 2) = \text{НОД}(4, 2) = \text{НОД}(2, 2) = 2$ ».

Анализ показывает, что для нахождения НОД нужно многократно вычитать из большего из двух чисел меньшее, причем количество повторений неизвестно, то есть следует применить оператор цикла с предусловием:

```

алг Определение НОД
нач цел a, b, НОД
    | Ввод чисел
вывод нс, "Задайте первое число "
ввод a
вывод нс, "Задайте второе число "
ввод b
    | Определение НОД
нц пока ?
    если a > b
        то
            a := a - b
        иначе
            b := b - a
    все
кц
    НОД := a      | или НОД := b
    | Вывод результата
вывод нс, "НОД этих чисел равен ", НОД

```

Определим условие, обозначенное символом «?», согласно приведенному выше правилу. Когда мы должны прекратить действия? Согласно алгоритму Евклида – когда $a = b$. Значит, вместо вопросительно знака должно быть записано противоположное условие: $a <> b$.

«Высший пилотаж»

В приведенной программе, когда a много больше b или когда b много больше a , например, при $a = 348$, $b = 6$, будут происходить многократные вычитания меньшего числа из большего, что нерационально. Этот недостаток можно устранить так: вместо многократных вычитаний можно определить остаток от деления большего числа на меньшее. При этом условие продолжения цикла должно быть таким: ни одно из меняющихся чисел не должно быть равно нулю:

```
...
| Определение НОД
нц пока  $a \neq 0$  и  $b \neq 0$ 
  если  $a > b$ 
    то
       $a := a \bmod b$ 
    иначе
       $b := b \bmod a$ 
  все
кц
```

После этого НОД может определен следующим образом:

```
если  $a = 0$ 
  то
    НОД :=  $b$ 
  иначе
    НОД :=  $a$ 
все
```

или, короче (и эффективно):

```
НОД :=  $a + b$ 
```

Возможен еще более краткий вариант программы нахождения НОД:

```
алг Определение_НОД
нач цел  $a, b, \text{остаток}, \text{НОД}$ 
цел  $a, b, \text{остаток}, \text{НОД}$ 
| Ввод чисел
...
нц пока  $a \bmod b \neq 0$ 
   $\text{остаток} := a \bmod b$ 
   $a := b$ 
   $b := \text{остаток}$ 
кц
НОД :=  $b$ 
вывод нс, "Наибольший общий делитель этих чисел равен", НОД
кон
```

В особенностях его работы разберитесь самостоятельно.

Контрольные вопросы

1. В каких случаях используется оператор цикла с предусловием?
2. Как определить условие, записываемое в таком операторе?
3. Может ли тело оператора цикла с предусловием:
– не выполниться ни разу?

– выполняться бесконечно (или до того момента, когда пользователь прервет выполнение программы¹⁶)?

Задачи для самостоятельной работы

10. Дано натуральное число. Определить:

а) сумму его цифр;

б) количество его цифр;

в) его первую цифру;

г) его k -ю (при счете справа налево) цифру;

д) число из его цифр, читаемых справа налево (то есть, например, из числа 2048 получить число 8402).

11. Дано натуральное число. Вывести на экран цифры его записи в двоичной системе счисления, начиная с последней двоичной цифры.

12. Даны натуральные числа a и b ($a > b$). Определить:

а) результат целочисленного деления a на b , не используя стандартную операцию целочисленного деления;

б) остаток от деления a на b , не используя стандартную операцию вычисления остатка.

11. Вывести на экран натуральные числа, не превышающие заданное число n . Оператор цикла с параметром не использовать.

13. Напечатать все нечетные числа из интервала $[10, 100]$. Оператор цикла с параметром не использовать.

Оператор цикла с предусловием также может быть использован в программе для обеспечения некоторой паузы в работе программы. Составьте соответствующий вариант самостоятельно.

Обращаем внимание на одну особенность работы оператор цикла с предусловием. Как следует из рис. 2.10, условие проверяется только *перед* выполнением тела оператора, но не проверяется в процессе выполнения. Неучет этого обстоятельства может привести к ошибкам. Очень наглядно это продемонстрировано в учебнике [3] на примере задачи с исполнителем Робот.

Робот, находящийся на поле из клеток и умеющий перемещаться на одну клетку влево, вправо, вверх или вниз, находится в клетке **А** (рис. 2.11) и выполняет фрагмент программы:

нц пока снизу свободно

вниз

вниз

кц

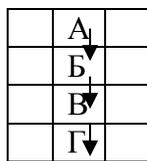


Рис. 2.11

Проследим работу оператора согласно схеме на рис. 2.10.

1. Исходное положение – Робот в клетке **А**. Проверяется условие **снизу свободно** – оно истинно – выполнится тело оператора, и Робот, сместившись дважды вниз, окажется в клетке **В**.

¹⁶ О том, как это сделать в используемой вами системе программирования, узнайте из её справочной системы. Мы же заметим, что такое состояние программы называют – «зацикливание» (см. раздел 3.6).

2. Опять проверяется условие снизу свободно – оно также истинно, и начнется выполнение тела оператора – Робот сначала сместится в клетку Г, но вторая команда (оператор) вниз выполниться не сможет (появится сообщение об ошибке).

Конечно, Робот – условный исполнитель. Приведем и «реальную» задачу. Пусть требуется вывести на экран числа в следующем порядке:

```
21 20
19 18
17 16
...
11
```

Казалось бы, можем записать в программе:

```
а := 21
нц пока а >= 11
    вывод нс, а, " ", а - 1
    а := а - 2
кц
```

Однако в результате выполнения этого фрагмента на экран будет выведено следующее (также согласно схеме на рис. 2.10):

```
21 20
19 18
17 16
...
11 10
```

что не соответствует условию.

Еще более «реальный» пример. В программах, моделирующих игру двух участников¹⁷, делающих ходы по очереди, должен быть фрагмент, блок-схема которого является следующей:

повторение пока не <условие окончания игры>

Ход первого участника

Вывод новой позиции

Ход второго участника

Вывод новой позиции

конец повторений

Казалось бы, все правильно – участники по очереди делают ходы до тех пор, когда игра не закончится. Однако при таком оформлении программы, согласно схеме, если после хода первого участника игра закончится, блок Ход второго участника все равно будет выполняться (на экран будут выводиться вопросы, связанные с ходом второго игрока).

Аналогичные ситуации могут иметь место в других компьютерных программах, моделирующих игры двух и более участников, и в ряде других случаев. Помните об этом!

Задание

Подумайте, как нужно изменить приведенную блок-схему для того, чтобы в описанной ситуации операторы блока Ход второго участника не выполнялись.

Продолжение — в следующем выпуске.

¹⁷ См., например, приложение 3.

Ответы, решения, разъяснения к заданиям, опубликованным в журнале «Мир информатики», выпуск № 42 (февраль 2020 г.)

Задание «Шесть вопросов» (рубрика «Поиск информации»)

Ответы

1. Созвездие Рыбы находится между зодиакальными созвездиями Водолей и Овен.
2. Чемпионом СССР по футболу в 1961 году в классе “А” стала команда киевского “Динамо”.
3. Русалка изображена на гербе столицы Польши, Варшавы.
4. Лауреатом Нобелевской премии мира за 1976 год стали Бетти Уильямс (Betty Williams) и Мейрид Корриган (Mairead Corrigan), Ирландия.
5. Название своей столицы китайцы произносят как “Бэйдзин”.
6. Денежной единице Веймарской республики для того, чтобы отличить ее от полновесных довоенных золотых, имевших хождение до августа 1914 года, было дано название “Бумажная марка”.

Кроссворд

Ответы

По горизонтали: 3. ЛИНИЯ. 6. КОПИЯ. 8. ЧИСЛО. 10. ДЕЛЕНИЕ. 11. ЛАСТИК. 12. КОНЕЦ. 13. ПАПКА. 15. ОМЕГА. 18. ДВА. 20. ОТБОР. 21. БЛОК. 22. САЛАТ. 24. СОЛЯРИС. 25. ЛОГО. 26. ИГРА.

По вертикали: 1. ЛИНЕЙКА. 2. ЛИСТИНГ. 4. ЯЧЕЙКА. 5. ОСТАНОВ. 6. КОЛЬЦО. 7. ПЕРО. 9. СКАН. 13. ПАТ. 14. КУРСОР. 16. МАССИВ. 17. АДА. 18. ДОЛЯ. 19. АУРА. 21. БЛОК. 23. ЛИРА.

Задача “Чернявый, Кудрявый и Лысый”

Напомним, что необходимо было определить, кто из трех участников преступной группировки — Чернявый, Кудрявый и Лысый — совершил преступление, если из двух заявлений каждого:

Чернявый: “Я не делал этого. Это сделал Лысый”;

Кудрявый: “Лысый не виноват. Это сделал Чернявый”;

Лысый: “Я этого не делал. Кудрявый этого тоже не делал”

— суд установил, что один из них дважды солгал, другой дважды сказал правду, а третий — один раз солгал, а другой раз сказал правду.

Решение

Рассмотрим возможные варианты для двух правдивых заявлений.

1. Предположим, что их сделал Лысый. Из его слов следует (что ни он, ни Кудрявый не виноват) следует, что виноват Чернявый. Но тогда оба утверждения Кудрявого также верны, а этого быть не может. Следовательно, наше предположение неверно.

2. Допустим, что два раза сказал правду Кудрявый. В этом случае оба утверждения Лысого тоже верны, что также противоречит условию.

3. Из предположения, что правдивыми являются заявления Чернявого (то есть преступление совершил Лысый), следует, что Кудрявый дважды солгал, а Лысый в первый раз солгал, а во второй раз сказал правду. То есть такой вариант допустим.

Ответ: преступление совершил Лысый.

Удобно для решения использовать таблицу, в которой рассмотреть три возможных варианта и использовать алгебру логики¹⁸:

				Слова Чернявого		Слова Кудрявого		Слова Лысого	
№	Ч	К	Л						
1	1	0	0						
2	0	1	0						
3	0	0	1						

В левой части таблицы использованы условные обозначения для допущений:

Ч – преступление совершил Чернявый,

К – преступление совершил Кудрявый,

Л – преступление совершил Лысый,

1 – допущение истинно,

2 – допущение ложно.

После этого записывается результат проверки высказываний подозреваемых для каждого допущения:

			Слова Чернявого		Слова Кудрявого		Слова Лысого	
Ч	К	Л	Не Ч	Л	Не Л	Ч	Не Л	Не К
1	0	0	0	0	1	1	1	1
0	1	0	1	0	1	0	1	0
0	0	1	1	1	0	0	0	1

По условию один дважды солгал, второй – дважды сказал правду, третий – раз солгал, раз сказал правду. Строка с допущением, в которой две истины, две лжи, одна истина и одна ложь, дает решение задачи. Такой строкой является последняя строка, то есть преступление совершил Лысый.

Задача “Мамед и Самед”

Напомним условие: “Летом Мамед и Самед очень любили есть вишню и черешню. Однажды мама купила бидон вишни (208 штук) и ведро черешни (320 штук).

Известно, что Мамед ест одну вишенку за 5 секунд, а одну черешенку — за 8 секунд. Самед же одну вишенку ест за 6 секунд, а одну черешенку — за 4 секунды.

1. Какое минимальное время понадобится каждому брату по поеданию ягод, если они разделят поровну и вишни, и черешни?

2. За какое минимальное время мальчики могут съесть все эти ягоды?

Брать в рот и есть сразу несколько ягод — неприлично и опасно ☺”.

Решение

1. Мамед съест свою половину вишен (104 шт.) за $104 \times 5 = 520$ секунд и свою половину черешен (160 шт.) за $160 \cdot 8 = 1280$ секунд, что всего составит 1800 секунд, или 30 минут. Самед свою долю вишен съест за 624 секунды и свою половину черешен за 640 секунд, а всего за 1264 секунды.

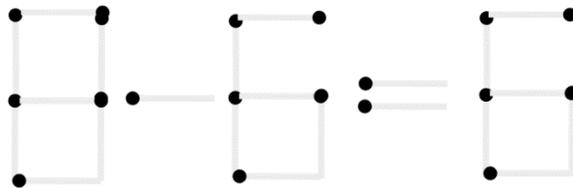
2. Чтобы съесть все за минимальное время, нужно, чтобы каждый начал есть то, что делает (☺) быстрее. Мамед съест все вишни за $208 \times 5 = 1040$ секунд. За это время Федя съест $1040 : 4 = 260$ черешен из 320. Останется 60 черешен.

¹⁸ Такое решение предложили Бабкова Наталья, Валенбахова Арина, Желтухина Анастасия и Клименко Дарья, г. Воронеж, лицей № 2 (учитель **Комбарова С.И.**).

Если за 8 секунд Вася ест одну черешенку, а Федя — две, то вместе за это время они съедят три черешенки. Значит, на оставшиеся 60 черешен им понадобится $60 : 3 \times 8 = 160$ секунд, а всего они потратят $1040 + 160 = 1200$ секунд, или 20 минут.

Головоломка «Получить верное равенство»

Напомним условие. Из спичек на столе выложили следующее:



1. Как получить верное равенство, переложив две спички?
2. Как получить верное равенство, убрав одну спичку и переложив другую?
3. Можно ли получить верное равенство, переложив только одну спичку?

Ответы (приведены равенства в виде цифр):

1. $6 + 0 = 6$

2.

$6 - 6 = 0$

$8 - 2 = 6$

$8 - 6 = 2$

3.

$0 + 6 = 6$

Задача «На новогоднем балу»

Напомним условие: «Как-то раз четыре товарища (Петя, Боря, Леша и Коля) пошли со своими сестрами (Светой, Наташей, Олей и Леной) на школьный новогодний бал.

Определите, кто чей брат и кто с кем танцевал во время первого танца, если:

а) каждый из ребят не танцевал со своей сестрой.

б) Лена танцевала с Петей, а Светлана — с братом Наташи.

в) Оля танцевала с братом Светланы, Боря — с сестрой Алеша, а Алеша — с сестрой Пети».

Ответ

Лена танцевала с Петей, Света с Борей, Оля с Лешей, Наташа с Колей. Петя — брат Оли, Боря — брат Наташи, Алеша — брат Светланы, Коля — брат Лены

Задача «Спорт и учеба»

Напомним условие задачи: «В классе 25 учеников. Из них 17 умеют играть в баскетбол, 13 умеют плавать, а 8 — ходить на лыжах. Ни один из учеников не владеет теми тремя видами спорта, но как баскетболисты, так и пловцы, и лыжники имеют хорошие или удовлетворительные оценки по математике, что тем более знаменательно, так как шесть учеников имеют неудовлетворительные оценки по этому предмету. Сколько учеников имеют отличные оценки по математике? Сколько пловцов умеют ходить на лыжах?»

Решение

Так как из 25 учеников класса 6 имеют неудовлетворительные оценки по математике, то учеников с оценками, по крайней мере удовлетворительными, всего 19. Это означает, что спортсменов в классе не более 19.

Если считать за единицу один вид спорта, которым занимается один ученик, то таких единиц в классе всего $17 + 13 + 8 = 38$, а так как ни один ученик не занимается сразу всеми тремя видами спорта, то, следовательно, учеников, занимающихся спортом, — ввиду

предыдущего замечания — всего 19, и каждый из них занимается одновременно двумя видами спорта.

Теперь уже легко ответить на вопросы, поставленные в задаче.

1. Ни один ученик в классе не имеет отличной оценки по математике.
2. Из 19 спортсменов 17 умеют играть в баскетбол; поэтому только два ученика умеют одновременно плавать и ходить на лыжах, следовательно, два пловца умеют ходить на лыжах.

Числовые ребусы в пятеричной системе счисления

Ответы

Ребус № 10	$\begin{array}{r} + 33 \\ 33 \\ \hline 121 \end{array}$	Ребус № 11	$\begin{array}{r} + 44 \\ 44 \\ \hline 143 \end{array}$	Ребус № 12	$\begin{array}{r} + 11 \\ 12 \\ \hline 23 \end{array}$
------------	---	------------	---	------------	--

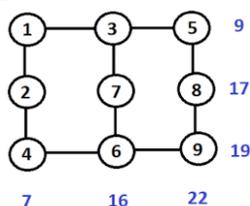
Пять ребусов

Ответы

Ребус № 1 – СОЦИАЛЬНАЯ СЕТЬ. Ребус № 2 – АККАУНТ. Ребус № 3 – НИКНЕЙМ.
Ребус № 4 – АВАТАР. Ребус № 5 – МЕССЕНДЖЕР.

Японская головоломка «сан-го-ку»

Решение



Ответы и решения представили:

— Абдувахидова Алина, Абдувахидова Софья, Камышанская Анастасия, Лефтерев Владислав, Смирнова Полина и Хозин Марат, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Абдуразакова Самира, Живаева Злата, Серякова Елена, Фаткуллина Аделина и Фомин Никита, Республика Башкортостан, г. Стерлитамак, гимназия № 4, учитель **Орлова Е.В.**;

— Бабкова Наталья, Валенбахова Арина, Желтухина Анастасия и Клименко Дарья, г. Воронеж, лицей №2, учитель **Комбарова С.И.**;

— Балабанов Константин, Владимирская обл., г. Александров, гимназия № 2, учитель **Волков А.Ю.**;

— Боровиков Иван, Костюк Максим, Кравец Сергей и Фунтиков Роман, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Великородных Светлана и Королёва Марина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Торопов Александр, Филимонова Надежда и Юмашев Сергей, средняя школа деревни Муравьево, Вологодская обл., учитель **Муравьева О.В.**

Наше пожелание 11-классникам:



Спасибо вам, уважаемые коллеги!

За активную работу с учащимися по материалам интернет-журнала “Мир информатики” редакция журнала “Информатика в школе” выражает благодарность:

- Артамоновой Виктории Владимировне, средняя школа поселка Новопетровский Московской обл.;
- Багровой Ольге Алексеевне, средняя школа г. Пионерский Калининградской обл.;
- Брунову Александру Сергеевичу, средняя школа села Ириновка, Новобурасский р-н Саратовской обл.;
- Букиной Елене Юрьевне, г. Зеленокумск Ставропольского края, школа № 1;
- Волкову Андрею Юрьевичу, Владимирская обл., г. Александров, гимназия № 2;
- Волкову Юрию Павловичу, Владимирская обл., г. Струнино, школа № 11;
- Демяхиной Оксане Вячеславовне, Республика Коми, г. Усинск, Центр дополнительного образования детей;
- Дикову Андрею Валентиновичу, г. Пенза, школа № 73;
- Долговой Галине Александровне, средняя школа села Восточное Нижегородской обл.;
- Евдокимовой Алевтине Ивановне, средняя школа поселка Осиновка, Алтайский край;
- Задубровскому Владимиру Федоровичу, Рязанская обл., Сапожковская средняя школа имени Героя России А.И. Тучина;
- Знаменской Ольге Владимировне, Владимирская обл., г. Александров, школа № 1;
- Зудину Василию Павловичу, Ардатовский областной многопрофильный техникум, поселок Ардатов Нижегородской обл.;
- Каликиной Татьяне Викторовне, Республика Карелия, Сегежский р-н, поселок Надвоицы, школа № 1;
- Комбаровской Светлане Ивановне, г. Воронеж, лицей № 2;
- Логинову Анатолию Владимировичу, г. Ровеньки Луганской обл., гимназия № 1, Малая академия наук, г. Ровеньки Луганской обл.;
- Марцинкевич Елене Евгеньевне, г. Рязань, школа № 61;
- Муравьевой Ольге Викторовне, средняя школа деревни Муравьево, Вологодская обл.;
- Орловой Елене Владимировне, Республика Башкортостан, г. Стерлитамак, гимназия № 4;
- Поповой Ирине Владимировне, Свердловская обл., г. Артёмовский, лицей № 21, Центр образования и профессиональной ориентации, посёлок Буланаш Артемовского р-на Свердловской обл.;
- Порываева Нафисе Сабитовне, Республика Татарстан, Мамадышский р-н, совхоз “Мамадышский”, политехнический колледж;
- Солодовой Елене Николаевне, Владимирская обл., г. Александров, школа № 14;
- Черновой Ларисе Ивановне, средняя школа села Сердар, Республика Марий Эл;
- Шитовой Любове Алексеевне, средняя школа села Горелово Тамбовской обл.;
- Юртаевой Галине Юрьевне, г. Челябинск, школа № 124;
- Ярцевой Ольге Владимировне, г. Ярославль, школа № 33;

