



ЗАПИСЬ АЛГОРИТМОВ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ

ОСНОВНЫЕ СВЕДЕНИЯ ОБ АЛГОРИТМАХ

11 класс



ИЗДАТЕЛЬСТВО

БИНОМ

Ключевые слова

- языки программирования
- данные
- структура данных
- идентификаторы
- операторы
- трассировочные таблицы

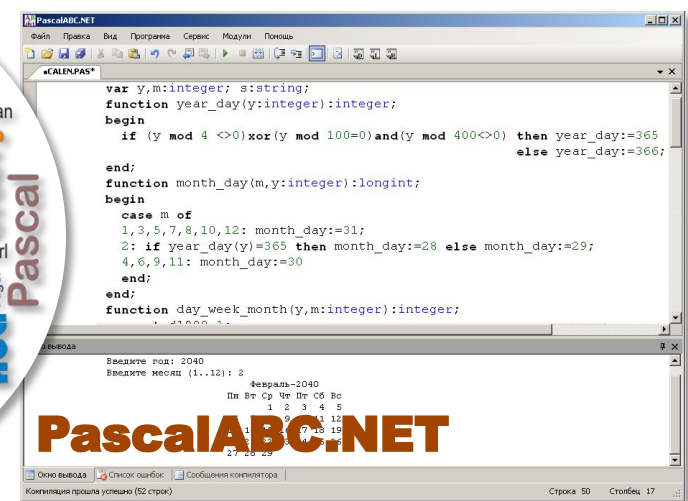
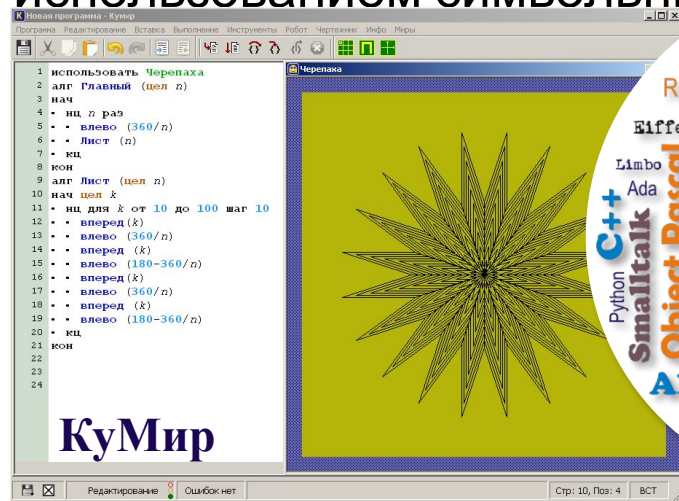


Язык программирования

Язык программирования – формальная знаковая система, предназначенная для записи компьютерных программ.

Компьютерную программу можно считать последовательностью строк символов некоторого алфавита. Современные системы програм-мирования допускают использование визуальных элементов (окон, иконок и др.) для построения программ, в частности, для создания интерфейса пользователя. Такое программирование называют *визуальным*. Тем не менее, основная, алгоритмическая часть любой программы строится с

использованием символьных строк.



Структурная организация данных

Информация, представленная в виде, пригодном для автоматизированной обработки, называется **данными**.

Компьютер оперирует только одним видом данных – отдельными битами, или двоичными цифрами.



Под **структурой данных** в общем случае понимают множество элементов данных и множество связей между ними.

Различают простые и сложные структуры данных.

Простые структуры данных не могут быть разделены на составные части больше, чем бит.

К ним относятся:

- числовые,
- символьные,
- логические и др.

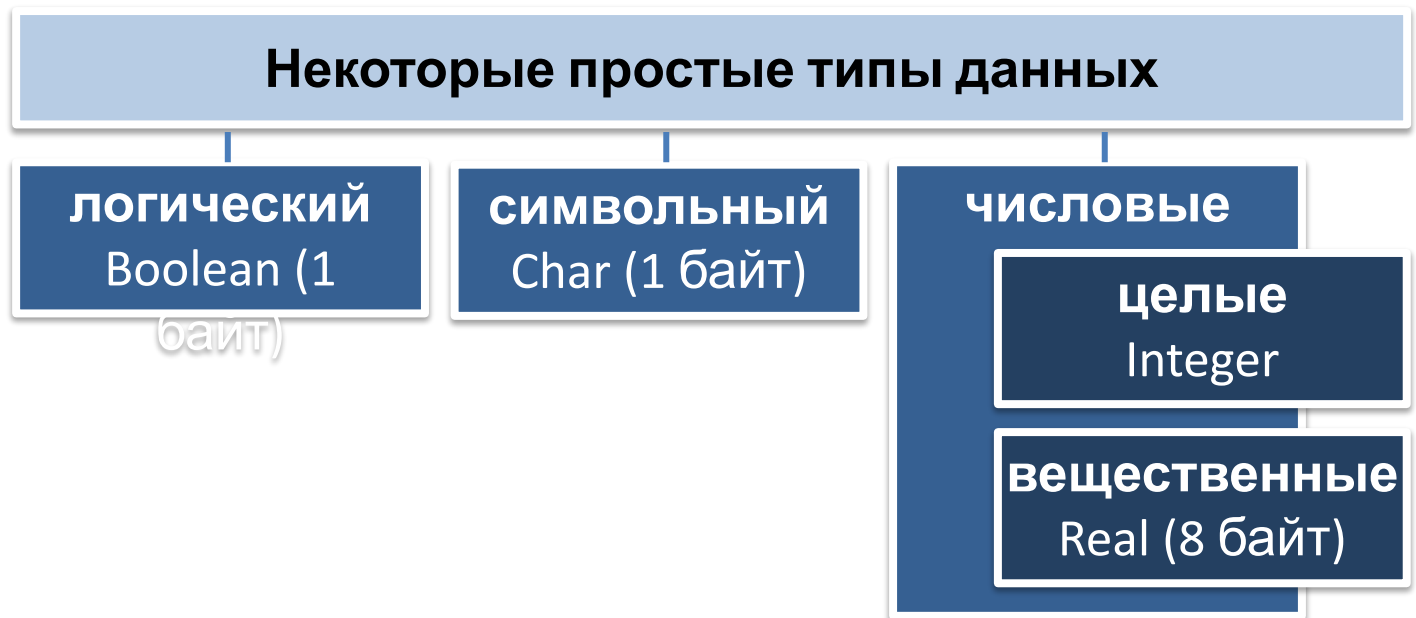
На основе простых структур строятся **сложные структуры данных**:

- массивы,
- списки,
- графы,
- деревья и др.

Некоторые простые типы данных



Pascal



Информация по каждому типу однозначно определяет:

- множество допустимых значений, которые может иметь тот или иной объект описываемого типа;
- множество допустимых операций, которые применимы к объекту описываемого типа;
- объём выделенной памяти для хранения данных указанного типа

Основные элементы языка Pascal



Pascal

- алфавит языка:
 - ✓ латинские буквы;
 - ✓ арабские цифры;
 - ✓ специальные символы;
- служебные слова, значение которых в языке программирования строго определено;
- постоянные и переменные величины;
- знаки операций;
- стандартные функции;
- выражения;
- операторы (языковые конструкции, с помощью которых в программах записываются действия, выполняемые над данными в процессе решения задачи)

Идентификаторы



Pascal

Все величины имеют имена (идентификаторы), формируемые по определённым правилам:



- имя может состоять из буквы или последовательности букв латинского алфавита, цифр и символа подчёркивания, но начинаться такая последовательность должна с буквы или символа подчёркивания;
- желательно, чтобы имя отражало смысл величины;
- имя не должно совпадать ни с одним из зарезервированных слов.

N12
Summa_X
Factorial
MyProgram



12N
Summa X
Факториал
Program



Операции в языке Pascal



Pascal

Арифметические операции

+	сложение
-	вычитание
*	умножение
/	деление
div	целочисленное деление
mod	остаток от целочисленного деления

Логические операции

not	логическое отрицание
and	логическое И
or	логическое ИЛИ
xor	исключающее ИЛИ

Операции отношений

=	равно
<>	не равно
<	меньше
>	больше
<=	меньше или равно
>=	больше или равно

Приоритет операций

1	not
2	*, /, div, mod, and
3	+, -, or, xor
4	=, <>, >, <, >=, <=

Структура программы



Pascal

program <имя программы>; **Заголовок программы**

var <переменные с указанием типов>; **Блок описания**
const <постоянные <с указанием типов>>; **данных**

begin **Блок описания действий**
 <последовательность команд>; **(программный блок)**
end.

Данные, обрабатываемые компьютером, хранятся в памяти. С точки зрения языка Pascal она разделена на секции, называемые *переменными*. Каждая переменная имеет имя, тип и значение; значения переменных могут меняться в ходе выполнения программы.

Блок описания действий начинается со слова *begin*, а заканчивается словом *end* и знаком точки. Действия представляются *операторами*. Операторы разделяются точкой с запятой.

Основные операторы языка Pascal



Pascal

Название	Общий вид
Присваивание	<i>Имя переменной := Значение</i>
Ввод с клавиатуры	readln (<i>список ввода</i>)
Вывод на экран	writeln (<i>список вывода</i>)
Условный	if <i>Условие</i> then <i>Оператор1</i> else <i>Оператор2</i>
Цикл с предусловием	while <i>Условие</i> do <i>Тело цикла</i>
Цикл с постусловием	repeat <i>Тело цикла</i> until <i>Условие</i>
Цикл с параметром с шагом +1	for <i>Переменная := Нач_знач</i> to <i>Кон_знач</i> do <i>Тело цикла</i>
Цикл с параметром с шагом -1	for <i>Переменная := Нач_знач</i> downto <i>Кон_знач</i> do <i>Тело цикла</i>

Анализ программ. Трассировочные таблицы

Для анализа свойств алгоритма и проверки его соответствия решаемой задаче используются **трассировочные таблицы**. В них фиксируется пошаговое исполнение алгоритма (программы), что позволяет наглядно представлять значения переменных, изменяющиеся при его выполнении. Поэтому трассировочные таблицы иначе называют **таблицами значений**. Используются трассировочные таблицы двух видов:

1 таблицы, каждая строка которых отражает результат *одного действия*

№	Команда или условие	Значение выражения	X	Y
1	readln (X)	356	356	
2	Y := 0	0		0
3	X > 0	да		
4	Y := Y*10 + X mod 10	6		6
5	X := X div 10	35	35	

2 таблицы, каждая строка которых отражает результат выполнения *группы действий*

Результат в КТ	k	x	S
Начальные значения	–	–	0
1	0	2	2
2	1	5	7
3	2	8	15
4	3	11	26

Трассировочная таблица первого вида



Pascal

```

program Number;
  var X, Y: longint;
begin
  readln(X);
  Y := 0;
  while X > 0 do
  begin
    Y := Y * 10 + X mod 10;
    X := X div 10;
  end;
  writeln (Y)
end.
    
```

Пример 1.

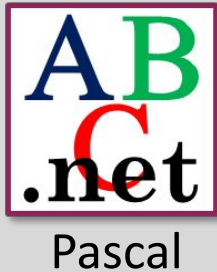
```

program Nu
  var X, Y: lo
begin
    
```

№	Команда или условие	Значение выражения	X	Y
1	readln (X)	356	356	
2	Y := 0	0		0
<p>В заголовке таблицы поместим имена всех переменных, используемых в программе. В отдельном столбце будем записывать команды и условия, имеющиеся в программе. Каждая строка таблицы соответствует <i>одному шагу</i> алгоритма.</p>				
10	Y := Y*10 + X mod 10	653		653
11	X := X div 10	0	0	
12	X > 0	НЕТ		
13	writeln (Y)			



Трассировочная таблица второго вида



```

program Summa;
  var k, x, S: integer;
begin
  S := 0;
  for k := 0 to 4 do
  begin
    x := k * 3 + 2;
    S := S + x
  end;
  writeln (S)
end.
    
```

Пример 2.

```

program Summa;
  var k, x, S: integer;
begin
    S := 0;
    k := 0;
  in
    x := k * 3 + 2;
    S := S + x;
  until (S = 40);
  writeln (S);
end.
    
```

Результат в КТ	k	x	S
Начальные значения	—	—	0
1	0	2	2
2	1	5	7
3	2	8	15
4	3	11	26
5	4	14	40

... (кТ). Выполнение алгоритма продолжается до контрольной точки и приостанавливается после выполнения отмеченной ею строки.

Будем считать, что контрольная точка поставлена на заголовке цикла. Укажите, что будет напечатано в результате выполнения программы.

Ответ: S = 40



Другие приёмы анализа программ



Pascal

```
var n, S: integer;
begin
  n := 1;
  S := 0;
  while n <= 625 do
  begin
    S := S + 30;
    n := n * 5
  end;
  write(s)
end.
```

Решение:

Выясним, какую функцию выполняет каждая из переменных, задействованных в программе.

Начальное значение переменной $S = 0$. При каждом выполнении тела цикла S увеличивается на 30 . Таким образом, искомое значение $S = 30 \cdot k$, где k — число выполнений тела цикла.

Начальное значение переменной $n = 1$. При выясним, при каком условии произойдёт выход из цикла. Цикл выполняется, пока $n \leq 625$. Следовательно, цикл завершится при достижении S значения, большего $625 = 5^4$, т.е. при $n = 5^5$.

Таким образом цикл выполнится 5 раз.

Следовательно, $S = 30 \cdot 5 = 150$.

Ответ: $S = 150$

Самое главное

Компьютерную программу можно считать последовательностью строк символов некоторого алфавита. Современные системы програм-мирования и языки допускают использование визуальных элементов (окон, иконок и др.) для построения программ и создания интерфейса пользователя. Тем не менее, основная, алгоритмическая часть любой программы строится с использованием символьных средств.

Компьютер оперирует только одним видом данных – отдельными битами, или двоичными цифрами. Задачи, решаемые с помощью компьютера, оперируют данными, имеющими форму чисел, символов, текстов и более сложных структур. Алгоритмы для обработки этих данных создаются с учётом их структуры – множества элементов данных и множества связей между ними.

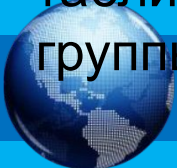


Самое главное

Различают простые и сложные структуры данных. Простые структуры данных не могут быть разделены на составные части больше, чем бит. К ним относятся числовые, символьные, логические и другие данные. Простые структуры данных служат основой для построения сложных структур данных – массивов, списков, графов, деревьев и др.

Для анализа свойств алгоритма и проверки его соответствия решаемой задаче используются трассировочные таблицы. В них фиксируется пошаговое исполнение алгоритма (программы), что позволяет наглядно представлять значения переменных, изменяющиеся при его выполнении. Используются трассировочные таблицы двух видов:

- таблицы, каждая строка которых отражает результат одного действия;
- таблицы, каждая строка которых отражает результат выполнения группы действий.



Вопросы и задания



Pascal

```
var x, d, R: longint;  
begin  
  readln(x);  
  R := 0;  
  while x > 0 do  
  begin  
    d := x mod 10;  
    R := 10*R + d;  
    x := x div 10  
  end;  
  writeln(R)  
end.
```

Задание 1. Ниже дана программа. Получив на вход натуральное число x , программа печатает число R . Укажи-те такое число x , при вводе которого будет напечатано двузначное число, сумма цифр которого равна 16. Если таких чисел несколько, укажите

це **Решение:**

Сложность этого задания состоит в том, чтобы разобраться, что делает программа.

Нетрудно заметить, что данная программа «переворачивает» исходное число x . Таким образом, надо найти двузначное число, сумма цифр которого равна 16:

$$16 = 7 + 9$$

$$16 = 8 + 8$$

$$16 = 9 + 7$$

Наименьшее число: 79

Ответ: 79

Ответ

Вопросы и задания



Pascal

Задание 2. Получив на вход натуральное число x ($x > 100$), программа печатает число M . Укажите наименьшее значение переменной x , при вводе которого алгоритм печатает 26.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 52;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M)
end.
```

Решение:

Данная программа реализует алгоритм Евклида для вычисления наибольшего общего делителя двух чисел – *НОД* (M , L).

Тогда, по условию задачи $\text{НОД}(52, x) = 26$.

Отсюда, $x = 104, 130, 156 \dots$

Наименьшее $x = 104$, но $\text{НОД}(52, 104) = 52$.

Следовательно, $x = 130$.

Ответ

Ответ: 130

Вопросы и задания



Pascal

```
var k, S: integer;  
begin  
  k := 10;  
  S := 0;  
  while k < 120 do  
  begin  
    S := S + k;  
    k := k + 5  
  end;  
  write (s)  
end.
```

Задание 3. Дана программа. Что будет напечатано после выполнения программы?

Решение:

Данная программа находит сумму арифметической прогрессии:

$$S = 10 + 15 + 20 + \dots + 115.$$

Формула для вычисления суммы первых n членов арифметической прогрессии:

$$S = \frac{a_1 + a_n}{2} \cdot n$$

В нашем случае:

$$n = (115 - 10) : 5 + 1 = 22.$$

Тогда:

$$S = (10 + 115) \cdot 22 / 2 = 1375.$$

Ответ: 1375

Ответ