

Занятие 1. Введение

Предлагаемая серия занятий предназначена для учащихся старших классов средней школы, интересующихся программированием. Предполагается, что целевая аудитория хотя бы немного знакома с основами алгоритмизации, имеет понятие о блок-схеме. В качестве языка программирования выбран Паскаль, системы программирования FreePascal и Delphi. Эти системы разрешены для использования на этапах Всероссийской олимпиады учащихся по информатике. От TurboPascal пришлось отказаться по следующим причинам:

1. Эта система программирования разработана для использования под управлением операционной системы MS-DOS и использует сегментированное адресное пространство, что не позволяет в полной мере использовать компьютерную память;
2. В ней отсутствуют 64-битные типы данных, использование которых на этапах всероссийской олимпиады школьников, по крайней мере, начиная с регионального, является очень и очень желательным.
3. Длина строковых переменных ограничена 255 символами.

Используя Delphi, мы не будем строить приложения с графическим интерфейсом, ограничиваясь исключительно консольными приложениями. Программы, которые мы будем разрабатывать, в равной степени можно будет компилировать и в среде Delphi, и в среде FreePascal. Это возможно потому, что FreePascal можно перевести в режим совместимости с Delphi.

Где взять FreePascal

FreePascal бесплатен. Его можно свободно скачать с сайта <http://www.freepascal.org>.

Вот ссылки на загрузку версии 2.4.0 для 32-битных операционных систем семейства Windows:

<ftp://ftp.chg.ru/pub/lang/pascal/fpc/dist/2.4.0/i386-win32/fpc-2.4.0.i386-win32.exe>

или

<ftp://ftp.freepascal.org/pub/fpc/dist/2.4.0/i386-win32/fpc-2.4.0.i386-win32.exe>

Размер дистрибутива около 35 Мбайт.

А здесь можно скачать систему помощи

<ftp://ftp.chg.ru/pub/lang/pascal/fpc/dist/2.4.0/docs/doc-html.zip>

или

<ftp://ftp.freepascal.org/pub/fpc/dist/2.4.0/docs/doc-html.zip>

Она скачивается и устанавливается отдельно. Размер файла около 10 Мбайт.

После установки FreePascal следует распаковать архив *doc-html.zip* в папку, в которую установлен FreePascal (например, c:\FPC\2.4.0 при установке с настройками по умолчанию). Потом запустите FreePascal IDE, в меню выберите Help — Files... — New, укажите файл *fptoc.html* из папки doc (c:\FPC\2.4.0\doc), нажмите ОК и **подождите**, пока файлы помощи будут проиндексированы.

Если у вас установлена операционная система Windows XP, то удобно настроить запуск FreePascal в полноэкранном режиме (для ОС Vista и Windows-7 это не получится). Для этого запустите FreePascal, нажмите Alt и пробел, в контекстном меню выберите «Свойства». На вкладке «Общие» настройте отображение во весь экран, а на вкладке «Расположение» установите размер буфера экрана по ширине 80, а по высоте 25. Размер окна должен установиться таким же автоматически. Нажмите кнопку «ОК», выберите «Изменить ярлык для запуска этого окна» и снова «ОК». На этом настройка завершена. Ещё мне кажется удобным использование менеджера файлов FarManager. Его можно скачать с сайта www.farmanager.com. Регистрация для жителей бывшего СССР свободная. Для это запускается из командной строки

far -r

и в появившемся окне в верхней строке пишется «xUSSR регистрация», а в нижнем — маленькими русскими буквами текущий день недели. Настройка этой программы для работы в полноэкранном режиме осуществляется так же.

Для удобства советую также в рабочей папке с вашими программами создать также 2 командных файла для запуска интегрированной среды разработки (IDE) и компилятора командной строки. В каждый файл впишите по одной строке. Если вы не изменяли настроек по умолчанию, то содержимое этих файлов должно быть следующее.

Файл fp.bat для запуска IDE

```
c:\FPC\2.4.0\bin\i386-win32\fp.exe %1 %2 %3 %4 %5
```

Файл fpc.bat для запуска компилятора командной строки

```
c:\FPC\2.4.0\bin\i386-win32\fpc.exe %1 %2 %3 %4 %5
```

Алфавит языка Паскаль

Алфавит языка Паскаль составляют буквы латинского алфавита, цифры и специальные знаки
+ - * / = < > [] . , () : ^ @ { } \$ #

При этом прописные и строчные буквы не различаются.

Все числа в языке Паскаль записываются в десятичной или шестнадцатеричной системе счисления. Перед записью шестнадцатеричного числа ставится знак доллара. Например, \$FF.¹ В записи вещественного числа целая и дробная части разделяются точкой. Например, 25.3 или -1.75. Вещественные числа могут быть записаны в экспоненциальном формате. При этом мантисса от порядка отделяется латинской буквой e или E. Например, запись 1.38e-23 означает $1,38 \cdot 10^{-23}$, а запись 5E12 означает $5 \cdot 10^{12}$.

Знаки арифметических операций в Паскале

Знак операции	Описание
+	Сложение
-	Вычитание
*	Умножение
/	Деление чисел. Результат — вещественное число независимо от типа операндов. Например, $3.8/2=1.9$ $7/2=3.5$
div	Частное от деления двух целых чисел (с отбрасыванием остатка). Округление до ближайшего целого в направлении нуля. Например, $17 \text{ div } 7 = 2$ $-17 \text{ div } 7 = -2$
mod	Остаток от деления двух целых чисел. Другими словами, $a \text{ mod } b = a - (a \text{ div } b) * b$ Например, $17 \text{ mod } 7 = 3$ $-17 \text{ mod } 7 = -3$

Операция возведения в степень в языке Паскаль не определена.

¹ FreePascal, кроме того, допускает запись чисел в восьмеричной и двоичной системах счисления. Запись восьмеричного числа начинается со знака &, а запись двоичного со знака %. Например, &25, %1010. Однако, эта возможность FreePascal отключается при работе в режиме совместимости с Delphi.

Приоритет операций такой же, как и в математике. Для его изменения используют круглые скобки.

Типы данных

Основная цель любой программы состоит в обработке данных. Данные различного типа хранятся в оперативной памяти и обрабатываются по-разному. В любом алгоритмическом языке каждая константа, переменная, результат вычисления выражения или функции должны иметь свой тип. Тип данных определяет

- внутреннее представление данных в памяти компьютера;
- множество значений, которые могут принимать величины этого типа;
- операции и функции, которые можно применять к величинам этого типа.

Все переменные в Паскале должны быть описаны. Это значит, что перед их использованием в программе должны быть объявлены их типы. Константы (постоянные) тоже имеют свой тип, но указывать явно в программе его обычно не нужно.

Ниже представлены целочисленные типы данных Delphi.

Таблица 1. Целочисленные типы данных Delphi

<i>Тип</i>	<i>Размер в байтах</i>	<i>Описание</i>	<i>Диапазон</i>
Integer	4	Целое число со знаком	-2147483648..2147483647
LongInt	То же, что и Integer		
SmallInt	2	Целое число со знаком	-32168..32167
ShortInt	1	Целое число со знаком	-128..127
Byte	1	Целое число без знака	0..255
Word	2	Целое число без знака	0..65535
Cardinal	4	Целое число без знака	0..4294967295
LongWord	То же, что и Cardinal		
Int64	8	Целое число со знаком	$-2^{63}..2^{63}-1$ (-9223372036854775808 .. 9223372036854775807)

Как уже указывалось, FreePascal может работать в режиме совместимости с Delphi. В этом случае всё, указанное в таблице, справедливо и для FreePascal. В других режимах тип Integer может оказаться эквивалентным не LongInt, а SmallInt. Кроме того, в FreePascal определён целочисленный беззнаковый тип данных Qword размером в 8 байт. Это соответствует диапазону значений от 0 до 18446744073709551615.

Таблица 2. Некоторые вещественные типы данных Delphi

<i>Тип</i>	<i>Размер в байтах</i>	<i>Диапазон</i>	<i>Количество значащих цифр</i>
Single	4	$-1.5 \times 10^{45} .. 3.4 \times 10^{38}$	7–8
Double	8	$-5.0 \times 10^{324} .. 1.7 \times 10^{308}$	15–16
Extended	10	$-3.6 \times 10^{4951} .. 1.1 \times 10^{4932}$	10–20
Real	То же, что и Double		

В Паскале много других типов данных, однако, пока остановимся на уже упомянутых.

Получение значений переменными

Присваивание

Переменная — это именованная область памяти, в которой хранятся данные определённого типа. Переменная имеет имя и значение. Имя служит для обращения к области памяти, в которой хранится значение. Имена переменных представляют собой последовательности латинских букв и цифр, начинающиеся с буквы. Пробел внутри имени переменной недопустим. Как известно, переменные могут получать своё значение в результате присваивания. В Паскале оператор присваивания обозначается двоеточием с равно. Например, команда «переменной x присвоить значение 5» на Паскале запишется так:

$$x := 5$$

Вы, должно быть, знаете, что после выполнения этой команды значение переменной x станет равным пяти, а её старое значение безвозвратно исчезнет.

Если требуется увеличить значение x на 1, то можно написать

$$x := x + 1$$

Возможен и другой вариант:

$$\text{inc}(x)$$

Если значение переменной x было равно пяти, то после выполнения одной из указанных выше команд оно станет равно шести.

Уменьшение на 1 запишется

$$x := x - 1$$

или

$$\text{dec}(x)$$

Ввод

Для сообщения *пользователем* значений переменным в ходе выполнения программы применяют операторы ввода Read и ReadLn. Список имён переменных для ввода значений записывается в скобках через запятую. Например, команда «*Ввести x, y* » на языке Паскаль записывается так²

$$\text{Read}(x, y)$$

или так

$$\text{ReadLn}(x, y)$$

Между ними есть довольно существенная разница. Read считывает требуемые данные, и если их в буфере ввода оказывается больше, чем нужно для данной операции ввода, то остальные данные оставляются для обработки следующим Read или ReadLn. В отличие от Read, ReadLn считывает требуемые данные, а остальные из буфера ввода удаляет, и следующие команды ввода будут требовать от пользователя новых нажатий клавиш на клавиатуре. Символы конца строки, помещаемые в буфер ввода при нажатии Enter, оператор ReadLn тоже удаляет.

Поясним это на примере. Пусть в программе есть 3 целочисленные переменные x , y и z , а их ввод в программе организован следующим образом:

$$\begin{aligned} &\text{Read}(x, y); \\ &\text{Read}(z) \end{aligned}$$

² Пока будем иметь в виду только ввод с клавиатуры

Пусть также пользователь, вводя значения x и y , случайно указал не два, а три числа, и в строке ввода указал

```
10 20 30
```

и нажал Enter. Тогда x станет равной 10, y получит значение 20, а «лишнее» число 30 не исчезнет, а будет назначено переменной z . Причем в этом случае Read(z) не вызовет остановки программы в ожидании действий пользователя.

Если же при тех же предположениях ввод организовать следующим образом

```
ReadLn(x,y);  
Read(z)
```

то x и y так же получают значения 10 и 20, а вот значение 30 будет проигнорировано и команда Read(z) вызовет остановку программы в ожидании действий пользователя. Он должен будет задать с клавиатуры ещё одно значение для переменной z .

Вывод

Для вывода в Паскале применяют операторы Write и WriteLn. Список выводимых значений, как и в случае ввода, записывается в скобках через запятую. Отличие WriteLn от Write состоит в том, что после вывода данных оператором Write курсор остаётся на той же строке, что и выведенные значения, а WriteLn перемещает курсор в начало следующей строки. Если требуется вывести строку символов, её заключают в апострофы:

```
WriteLn('Здравствуй, мир!')
```

Ещё пример. Пусть целочисленные переменные $x=10$, $y=20$, $z=30$. Тогда после выполнения фрагмента программы

```
WriteLn(x,' ',y);  
WriteLn(z)
```

на первой строке мы увидим числа 10 и 20, разделённые пробелом, а на следующей строке число 30:

```
10 20  
30
```

Вещественные числа по умолчанию выводятся в экспоненциальной форме. Если необходимо вывести их в формате с десятичной точкой, то в операторе вывода после имени переменной надо указать *минимальное* количество знаковых знаков для вывода числа и количество десятичных знаков в дробной части числа, разделённые двоеточием. Например, запись

```
Write(x:7:3)
```

означает, что для вывода значения вещественной переменной x будет использовано не менее 7 знаков, включая десятичную точку и пробелы перед записью числа, а само число будет округлено с точностью до трёх знаков после десятичной точки. Например, если значение переменной x равно 5.123567, то на экране мы увидим 5.124 с двумя пробелами перед цифрой 5. Если бы семи знаков для вывода значения переменной оказалось мало, то при выводе использовалось бы необходимое количество знаков, однако округление в любом случае велось бы до трёх знаков после запятой. Например, при значении x , равном 12345.6, вывелось бы 12345.600

Структура программы на языке Паскаль

Программа на языке Паскаль имеет следующую структуру. Пока что представим её с некоторыми упрощениями.

```
Program <имя_программы>;
```

```

Const    <имя_константы1> = <значение1>;
         <имя_константы2> = <значение2>;

...
Var      <список_1 имён переменных через запятую> : <тип>;
         <список_2 имён переменных через запятую> : <тип>;

...
Begin
    <оператор_1>;
    <оператор_2>;
    ...
    <оператор_N>
End.

```

Все идентификаторы языка Паскаль — это последовательности букв латинского алфавита и цифр, начинающиеся с буквы. Имена переменных, констант, программ являются идентификаторами. Операторы языка Паскаль разделяются между собой точкой с запятой. Она является именно разделителем между операторами, а не частью самого оператора, поэтому, например, перед End её ставить не нужно. Разрешается записывать несколько операторов в одной строке, разделяя их точкой с запятой. После End, обозначающего конец программы, обязательно ставится точка.

Вот пример простейшей программы на языке Паскаль. В ней вводятся два целых числа, вычисляются их сумма и разность и выводятся в одну строку. После выполнения этой программы курсор переводится в начало следующей строки.

```

Program SimplePr;
Var a,b,c : Integer;
Begin
    WriteLn('Введите 2 целых числа');
    c := a + b;
    Write('Их сумма равна ', c);
    c := a - b;
    WriteLn(' ', a разность ', c)
End.

```

Контрольные вопросы

- Вычислите значение переменных b и k
 - $b := 98 \text{ div } 10$
 - $b := 98 \text{ mod } 10$
 - $b := 98 \text{ div } 100$
 - $k := 98 / 100$
- Какой тип данных вы бы выбрали для переменной Delphi, которая
 - принимает неотрицательные целочисленные значения, не превышающие двухсот;
 - принимает целочисленные значения в пределах от -32000 до +32000;
 - принимает целочисленные значения от 1 до 60000;
 - принимает целочисленные значения от -10^9 до $+10^9$;
 - принимает целочисленные значения от -100 до +100;
 - хранит значение гипотенузы прямоугольного треугольника.
- Выберите правильно записанные имена переменных
 - w1
 - 1w
 - Summa
 - k25
 - юзй
 - u
- Какие из предложенных в предыдущем вопросе вариантов ответа могут являться также именами программ?
- Значение переменной s равно 1.41421356237

- a. Какой тип данных вы бы использовали для хранения значения этой переменной с указанной точностью?
- b. Какое значение будет выведено на экран после выполнения оператора `Write(s:5:3)`?
- c. Будет ли выведен пробел перед этим значением?
- d. Будет ли переведён курсор на новую строку после вывода переменной `s`?