

## Задание для студентов гр. ИС 2 на период с 20.04.2020 – 26.04.2020 (4 часа – 2 пары)

Дисциплина «Математика»

Преподаватель Токарская М.С.

Почта для обратной связи: [maya\\_tok@mail.ru](mailto:maya_tok@mail.ru)

Тел. 89147174421 – WhatsApp – если есть вопросы.

**Все задания отправлять на почту!!!!**

### Задание:

1. Изучить лекцию «Первое знакомство с языком Паскаль», составить конспект. Алфавит языка записываем в том случае, если не писали его в конспекте по заданию №1.
2. Скачать среду программирования и установить ее на ПК:  
<https://www.softportal.com/get-10585-turbo-pascal-school-pak.html>
  - заранее создайте папку на своем ПК – ТПаскаль
  - скачайте туда архив
  - распакуйте архив в ту же папку – появится папка SchoolPak
  - двойным нажатием запустите файл turbo.exe
3. Выполнить пробное практическое задание. Оформить его следующим образом:
  - a. сам текст задания
  - b. блок-схема алгоритма решения (составить самостоятельно)
  - c. текст программы
  - d. скриншот экрана с набранной программой
  - e. скриншот окна выполнения программы

### Материалы и оборудование:

1. Конспекты предыдущих уроков
2. <http://book.kbsu.ru/> - Шауцукова Л.З. Информатика:  
[http://book.kbsu.ru/practice/2\\_1/2\\_1\\_0.html](http://book.kbsu.ru/practice/2_1/2_1_0.html) - пример 1.1., 1.2 (программа на ТПаскале)
3. [http://les-collegelik.ru/DistObuch2020/IS-2/OsnovAlgoritm/metodicheskie\\_ukazaniya\\_pascal.pdf](http://les-collegelik.ru/DistObuch2020/IS-2/OsnovAlgoritm/metodicheskie_ukazaniya_pascal.pdf) - Методическое руководство по Паскаль ABC (сайт колледжа)
4. <http://mif.vspu.ru/books/pascal/> - Turbo Pascal 7.0, Электронный учебник для студентов и школьников
- 5.
6. <http://bpascal.ru/doc/errors.php> - ошибки программы

### «Первое знакомство с языком Паскаль»

Первая версия языка Паскаль была разработана швейцарским ученым *Никлаусом Виртом* в 1968 году. Первоначально язык предназначался для целей обучения, поскольку он является достаточно детерминированным, т.е. все подчиняется определенным правилам, исключений из которых не так много. Основные характеристики: относительно небольшое количество базовых понятий, простой синтаксис, быстрый компилятор для перевода исходных текстов в машинный код.

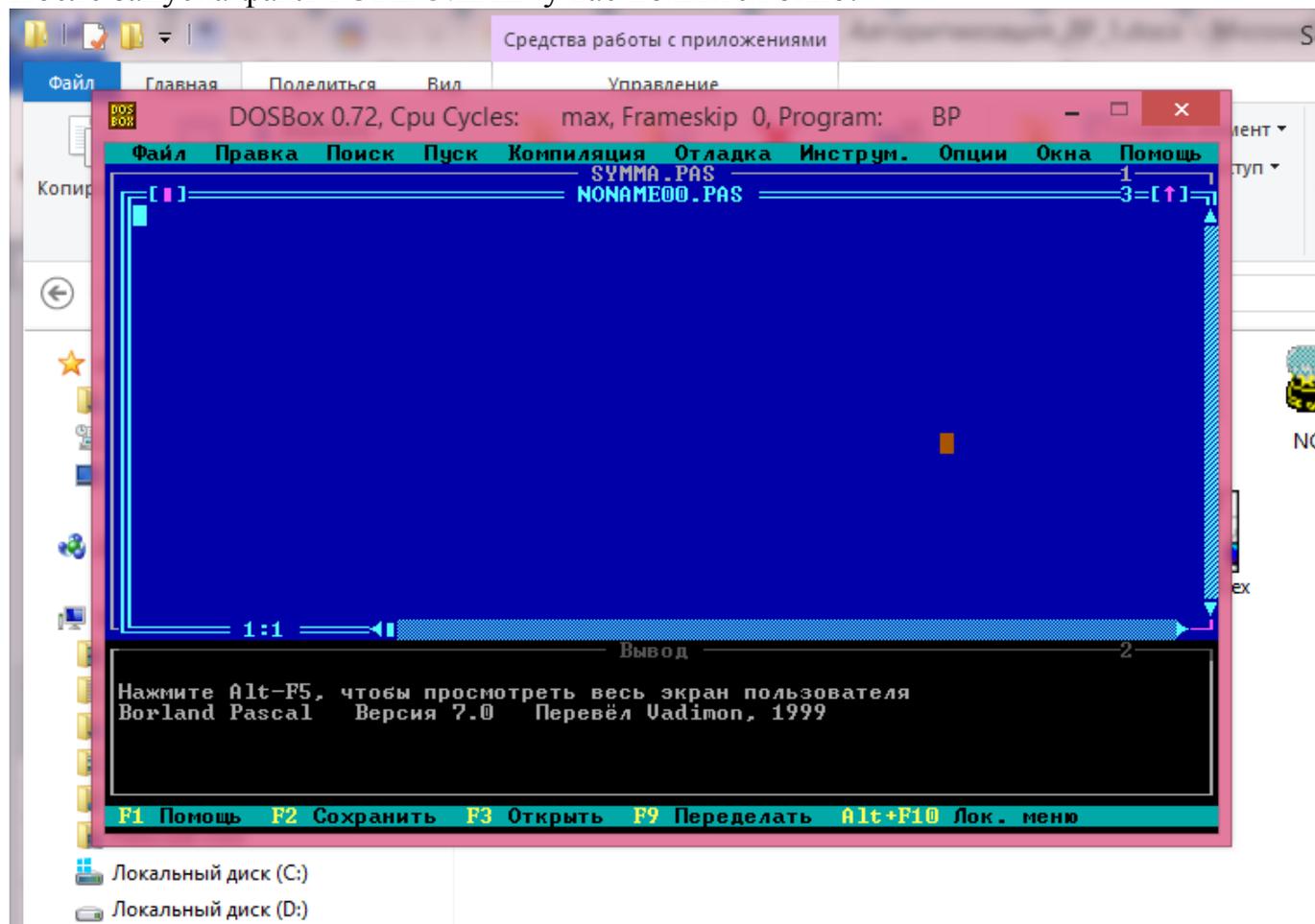
В 1992 г. фирма Borland International выпустила два пакета, основанных на языке Паскаль: Borland Pascal 7.0 и Turbo Pascal 7.0. Первый может работать в трех режимах - обычном и защищенном режимах MS DOS и в системе Windows. Для него необходимо порядка 30 Мбайт на жестком диске и около 2 Мбайт оперативной памяти. Турбо Паскаль 7.0 работает только в обычном режиме MS DOS и менее требователен к характеристикам компьютера. Поскольку основные компоненты, которые мы будем рассматривать в нашем курсе, совпадают в обоих продуктах, в дальнейшем будет использоваться название Турбо Паскаль.

Пакет включает в себя алгоритмический язык программирования высокого уровня, встроенный редактор и среду, предназначенную для отладки и запуска программ. Кроме того, пакет содержит большой объем справочной информации (англоязычной).

Турбо Паскаль относится к компиляторным языкам. Далее на примере вы поймете эту особенность.

*Турбо Паскаль предназначен для создания программ для решения математических задач, обработки текстов, получении графики и звука.*

После запуска файл TURBO.EXE у вас появится окно:



### Рассмотрим окно и основные сочетания клавиш при работе в Паскале:

- После запуска программы на экране появляется окно синего цвета или в режиме полноэкранного, или оконного. Для переключения между этими режимами используется сочетание клавиш **Alt+Enter**.
- Для создания нового окна используется команда **Файл→Новый**
- Для того, чтобы открыть файл команда **Файл→Открыть** или клавиша **F3**

– Для сохранения файла – **Файл**→**Сохранить** и **Файл**→**Сохранить как** или клавиша **F2**

– Для выхода из TP – **Файл**→**Выход** или сочетание клавиш **Alt+ X**

После написания программы необходимо протестировать и отладить программу, в TP программа **компилируется**, т.е. переводит операторы на язык машинных команд или проверяет правильность написания программы, соответствие правилам. Значит, после написания программы необходимо проверить ее правильность путем нажатия сочетания клавиш **Alt+F9** или через команду меню **Отладка**, и если имеются ошибки, он покажет ее.

– Итак, **Alt+F9** или меню **Отладка**

– Если ошибок нет, необходимо запустить программу путем нажатия сочетания клавиш **Ctrl+F9** или через команду меню **Запуск**.

– Для просмотра результатов выполнения программы используется сочетание клавиш **Alt+F5**.

– Еще одно важное сочетание клавиш: для переключения на русский язык применяется сочетание клавиш **правых Shift+Ctrl**, а для обратного перехода на английский – **левые Shift+Ctrl**.

Помните!!! Турбо Паскаль – программа MS DOS, поэтому раскладка языка Windows не имеет никакого отношения к нему.

– Alt+F11 - вернуть управление мышью в Windows.

– Alt+X – выход из Турбо Паскаля. Не рекомендуется выходить из программы, просто нажав кнопку **Закреть** окна программы.

## **СИНТАКСИС И СЕМАНТИКА ЯЗЫКА**

Описание каждого элемента языка задается его СИНТАКСИСОМ и СЕМАНТИКОЙ. Синтаксические определения устанавливают правила построения элементов языка. Семантика определяет смысл и правила использования тех элементов языка, для которых были даны синтаксические определения.

### **Правила пунктуации:**

1. Точка с запятой (;) ставится в конце заголовка программы, в конце раздела описания переменных, после каждого оператора. Перед словом End точку с запятой можно не ставить;

2. Запятая является разделителем элементов во всевозможных списках (список переменных в разделе описания, списке вводимых и выводимых величин)

### **Алфавит языка**

Алфавит - это совокупность допустимых в языке символов. Алфавит Турбо Паскаль включает следующий набор основных символов:

- строчные и прописные латинские буквы:
- **A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**
- **a b c d e f g h i j k l m n o p q r s t u v w x y z**

- пробел
- подчеркивание: \_ (расценивается как буква)
- арабские цифры: 0 1 2 3 4 5 6 7 8 9
- знаки операций: + - \* / = < > <= >= := @
- ограничители: . , ' ( ) [ ] ( . ) { } ( \* \* ) .. : ;
- спецификаторы: ^ # \$
- служебные (зарезервированные) слова (т.е. нельзя их использовать в своих программах в качестве переменных):

ABSOLUTE	EXPORTS	LIBRARY	SET
ASSEMBLER	EXTERNAL	MOD	SHL
AND	FAR	NAME	SHR
ARRAY	FILE	NIL	STRING
ASM	FOR	NEAR	THEN
ASSEMBLER	FORWARD	NOT	TO
BEGIN	FUNCTION	OBJECT	TYPE
CASE	GOTO	OF	UNIT
CONST	IF	OR	UNTIL
CONSTRUCTOR	IMPLEMENTATION	PACKED	USES
DESTRUCTOR	IN	PRIVATE	VAR
DIV	INDEX	PROCEDURE	VIRTUAL
DO	INHERITED	PROGRAM	WHILE
DOWNTO	INLINE	PUBLIC	WITH
ELSE	INTERFACE	RECORD	XOR
END	INTERRUPT	REPEAT	
EXPORT	LABEL	RESIDENT	

### Элементарные конструкции

Элементарные конструкции языка Паскаль включают в себя имена, числа и строки.

**Имена (идентификаторы)** называют элементы языка - константы, метки, типы, переменные, процедуры, функции, модули, объекты.

Идентификатор в Турбо Паскале может включать в себя:

- буквы латинского алфавита,
- цифры
- символ подчеркивания.

Строчные и прописные буквы не различаются (например, NAME, Name и name будет означать одно и то же).

Цифра не может стоять на первом месте в идентификаторе, а символ подчеркивания может находиться в любой позиции (например, name1 и name2item являются допустимыми идентификаторами, а 5name - нет; \_name, name\_, name\_item - тоже до-

пустимые названия). Длина идентификатора может быть любой, но значимыми являются только первые 63 символа. В качестве имен не допускается использовать служебные слова.

Для отделения друг от друга идентификаторов, чисел, зарезервированных слов используются разделители. В качестве них можно использовать:

- пробел и табуляцию;
- перевод строки;
- комментарий.

В любом месте программы, где можно поместить один разделитель, их можно поместить любое количество и в любом сочетании, что позволяет наглядно представить структуру программы.

Комментарии заключаются либо в скобки { ... }, либо в скобки вида (\* ... \*) и могут занимать любое число строк.

**Числа** в языке Паскаль обычно записываются в десятичной системе счисления. Они могут быть целыми и действительными. Положительный знак числа может быть опущен.

Целые числа записываются в форме без десятичной точки, например:

217 -45 8954 +483

Действительные числа записываются в форме с десятичной точкой:

28.6 0.65 -0.018 4.0

Возможна также запись с использованием десятичного порядка, который изображается буквой E: 5E12 -1.72E9 73.1E-16

В "перевод" такую запись следует понимать соответственно как:

$5 \times 10^{12}$   $-1.72 \times 10^9$   $73.1 \times 10^{-16}$

Паскаль допускает запись целых чисел и фрагментов действительных чисел в форме с порядком в шестнадцатичной системе счисления:

\$7F \$40 \$ABC0

**Строки** в языке Паскаль - это последовательность символов, записанная между апострофами. Если в строке в качестве содержательного символа необходимо употребить сам апостроф, то следует записать два апострофа.

Примеры строк:

'СТРОКА' 'STRING' 'ПРОГРАММА' 'АД"ЮТАНТ'

## ВВОД ДАННЫХ

Ввод данных – это передача информации от внешних устройств в оперативную память.

Вывод данных – передача данных из ОП на внешние носители.

Для ввода исходных данных используется **процедура:**

Название	Запись на языке Паскаль	Значение
<b>Read</b>	<b>Read (&lt;список ввода&gt;);</b> Например Read (A1,A2,...AK);	<список ввода> - последовательность имен переменных, разделенная запятыми; <b>Read</b> – считать; оператор обращения к стандартной процедуре ввода

При выполнении оператора работа ПК прерывается, после чего пользователь набирает на клавиатуре значения переменных (в соответствии с типом), отделяя из друг от друга **ПРОБЕЛАМИ**. В конце набора нажимают Enter.

Например:

```
Var T, K: real;  
    M, S: Integer;  
Begin  
    Read (T, M);  
    Read (K, S);
```

На клавиатуре следует ввести: **253.98 {пробел} 120 {Enter} 2.62E-02 {пробел} 34 {Enter}**

Если в программе имеется несколько операторов Read, то данные вводятся потоком, т.е. после считывания значений переменных для одного оператора Read данные для следующего оператора до окончания строки считываются из той же строки на экране, что и для предыдущего затем происходит переход на новую строку

Название	Запись на языке Паскаль	Значение
ReadLn	<b>ReadLn (&lt;список ввода&gt;);</b> <b>Например:</b> ReadLn (A1,A2,...AK);	<список ввода> - последовательность имен переменных, разделенная запятыми; <b>ReadLn</b> – считать строку; оператор обращения к стандартной процедуре ввода

Ввод данных; после считывания последнего в списке значений для одного оператора ReadLn данные для следующего ReadLn будут считываться с начала новой строки.

```
Var T, K: real;  
    M, S: Integer;  
Begin  
    Readln (T, M);  
    Readln (K, S);
```

На клавиатуре следует ввести: **253.98 {пробел} 120 {Enter} – первая строка**  
**2.62E-02 {пробел} 34 {Enter} – вторая строка**

При вводе исходных данных происходит преобразование из внешней формы представления во внутреннюю, определяемую типом переменных. Переменные, образующие список ввода, могут принадлежать либо к целому, либо к действительному, либо к символьному типам. Чтение исходных данных логического типа в языке Паскаль недопустимо.

Значения исходных данных могут отделяться друг от друга пробелами и нажатием клавиш табуляции и Enter.

Не допускается разделение вводимых чисел запятыми!

## ВЫВОД ДАННЫХ

Для вывода результатов работы программы на экран используются процедуры:

Название	Запись на языке Паскаль	Значение
<b>Write</b>	<b>Write (&lt;список вывода&gt;);</b> Например <b>Write (A1,A2,...AK);</b>	<список вывода> - последовательность имен переменных, разделенная запятыми; <b>Write</b> – записать; оператор обращения к стандартной процедуре ввода
<p>Элементами списка вывода могут быть выражения различных типов, например:  <b>Write (234);</b> – выводится целая константа  <b>Write (A+B-C);</b> - выводится значение выражения  <b>Write (X, Summa, Arg1, Arg2);</b> - выводится значение переменных</p> <p><i>При выводе на экран нескольких чисел в троюку они <b>НЕ отделяются пробелами</b>. Об этом программист должен позаботиться самостоятельно, добавляя пробел, как буквенный символ:</i>  <b>Write (A, ' ', B, ' ', L)</b> – выведет переменные <b>A B L</b> через пробел</p>		
Название	Запись на языке Паскаль	Значение
<b>WriteLn</b>	<b>WriteLn (&lt;список вывода&gt;);</b> Например: <b>ReadLn (A1,A2,...AK);</b>	<список вывода> - последовательность имен переменных, разделенная запятыми; <b>ReadLn</b> – записать строку; оператор обращения к стандартной процедуре ввода
<p><b>Отличие от первого оператора в том, что после вывода последнего в списке значения происходит перевод курсора в начало следующей строки</b></p>		

Если процедура **WriteLn** используется без параметров, то она просто производит пропуск строки и переход к началу следующей строки.

Переменные, составляющие список вывода, могут относиться к целому, действительному, символьному или булевскому типам. В качестве элемента списка вывода кроме имен переменных могут использоваться выражения и строки.

Форма представления значений в поле вывода соответствует типу переменных и выражений: величины целого типа выводятся как целые десятичные числа, действительного типа - как действительные десятичные числа с десятичным порядком, символьного типа и строки - в виде символов, логического типа - в виде логических констант TRUE и FALSE.

**Оператор вывода позволяет задать ширину поля вывода для каждого элемента списка вывода.**

В этом случае элемент списка вывода имеет вид **A:K**, где **A** - выражение или строка, **K** - выражение либо константа целого типа:

1. Если выводимое значение занимает в поле вывода меньше позиций, чем **K**, то перед этим значением располагаются пробелы.
2. Если выводимое значение не помещается в ширину поля **K**, то для этого значения будет отведено необходимое количество позиций.

**Для величин действительного типа элемент списка вывода может иметь вид:**

**A:K:M**, где **A** - переменная или выражение действительного типа, **K** - ширина поля вывода, **M** - число цифр дробной части выводимого значения. **K** и **M** - выражения или константы целого типа.

В этом случае действительные значения выводятся в форме десятичного числа с фиксированной точкой.

### Пример записи операторов вывода:

```
var rA, rB: Real;  
    iP, iQ: Integer;  
    bR, bS: Boolean;  
    chT, chV, chU, chW: Char;  
begin  
    ...  
    WriteLn(rA, rB:10:2);  
    WriteLn(iP, iQ:8);  
    WriteLn(bR, bS:8);  
    WriteLn(chT, chV, chU, chW);  
end.
```

## СТРУКТУРА ПРОГРАММЫ НА ТУРБО ПАСКАЛЬ

Программа состоит из заголовка и тела (блока), в конце которого следует точка – признак конца программы. Блок содержит разделы описаний и раздел операторов:

```
Program имя программы; } {заголовок программы}  
uses } {раздел подключения модулей}  
Label ...; {раздел описания меток}  
Const ...; {раздел описания констант}  
Type ...; {раздел определения типов}  
Var ...; {раздел описания переменных}  
Function ...; Procedure ...; {раздел описания функций и процедур}  
BEGIN ... } {раздел операторов}  
END.
```

блок описаний

Раздел операторов есть в любой программе и является основным. Разделы описаний могут не все содержаться в каждой программе.

В ТПаскале, в отличие от стандартного языка, возможно отсутствие заголовка программы, следование разделов Const, Type, Var, Label друг за другом в любом порядке в разделе описаний и сколько угодно раз.

**Раздел описания модулей определяется служебным словом USES** и содержит имена подключаемых модулей (библиотек) как входящих в состав системы Turbo Pascal, так и написанных пользователем. Раздел описания модулей должен быть первым среди разделов описаний.

Имена модулей отделяются друг от друга запятыми:

```
uses CRT, Graph;
```

Любой оператор в программе может быть помечен меткой. Имя метки задается по правилам образования идентификаторов Турбо Паскаль. В качестве метки также могут использоваться произвольные целые числа без знака, содержащие не более четырех цифр. Метка ставится перед оператором и отделяется от него двоеточием. Все метки, используемые в программе, должны быть перечислены в **разделе описания меток**, например:

```
label 3, 471, 29, Quit;
```

Описание констант позволяет использовать имена как синонимы констант, их необходимо определить в **разделе описания констант**:

```
const K= 1024; MAX= 16384;
```

В **разделе описания переменных** необходимо указать все переменные, используемые в программе, и определить их тип:

```
var P,Q,R: Integer;
```

```
    A,B: Char;
```

```
    F1,F2: Boolean;
```

Описание типов, процедур и функций будет рассмотрено ниже. Отдельные разделы описаний могут отсутствовать, но следует помнить, что в Паскаль - программе должны быть обязательно описаны все компоненты программы.

**Раздел операторов представляет собой составной оператор, который содержит между служебными словами**

```
Begin
```

```
.....
```

```
end
```

последовательность операторов.

Операторы отделяются друг от друга символом (;).

Текст программы заканчивается символом «точка».

### **Практическое задание:**

**Составить к каждому примеру словесное описание и блок-схему.**

**Выполнить задание (см.начало занятия)**

#### **Пример 1**

```
program Primer; {вычисление суммы двух чисел}
uses Crt; {подключение модуля, необходимого для процедур
           очистки экрана и задержки}
var
  x,y,s: integer;
begin
  ClrScr; {очистка экрана}
  WriteLn('Введите через пробел два числа ');
  ReadLn(x,y);
  s := x + y;
  WriteLn('Сумма чисел равна ',s);
  ReadKey; {ожидание нажатия клавиши}
end.
```

### ***Пример 2.***

```
program Znakomstvo;
uses Crt;
Var Name: String;
    Age: Integer;
begin
ClrScr;
WriteLn(' Здравствуй, я компьютер, а тебя как звать?');
Write(' Меня зовут ');
ReadLn(Name);
WriteLn(' Очень приятно ', Name, ' Сколько тебе лет? ');
Write(' Мне ');
ReadLn(Age);
WriteLn(' лет ');
WriteLn(' Ого! Целых ', Age, ' лет. Да ты совсем взрослый!');
ReadLn
end.
```

### ***Пример 3.***

```
program Obem;
uses Crt;
Var s,v,r,h: real;
begin
ClrScr;
WriteLn(' Введите радиус R и высоту h');
ReadLn(r,h);
S:=pi*sqr(r);
V:=s*h;
WriteLn(' V = ', v:6:2);
WriteLn(' S = ', s:8:3);
end.
```