



Кафедра конструирования  
и производства радиоаппаратуры

---

УТВЕРЖДАЮ

Заведующий кафедрой КИПР

\_\_\_\_\_ В.Н. ТАТАРИНОВ

“ \_\_\_ ” \_\_\_\_\_ 2012 г.

## Работа в интегрированной среде Borland Pascal

Лабораторная работа по дисциплинам «Информатика» для студентов специальностей  
211000.62 «Конструирование и технология электронных средств» (бакалавриат) и 162107.65  
«Информатика и информационные технологии» (специалитет)

Разработчик:

Доцент кафедры КИПР

\_\_\_\_\_ Ю.П. Кобрин

## СОДЕРЖАНИЕ

<b>1</b>	<b>ЦЕЛЬ РАБОТЫ</b> .....	<b>3</b>
<b>2</b>	<b>ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ</b> .....	<b>3</b>
<b>3</b>	<b>КОНТРОЛЬНЫЕ ВОПРОСЫ</b> .....	<b>3</b>
<b>4</b>	<b>ОТЧЕТНОСТЬ</b> .....	<b>3</b>
<b>5</b>	<b>ИНТЕГРИРОВАННАЯ СРЕДА BORLAND PASCAL</b> .....	<b>4</b>
5.1	Краткие сведения о работе в системе Borland Pascal .....	4
5.2	Редактирование программ .....	7
5.3	Работа со справочной системой .....	9
5.4	Компиляция, запуск и отладка программ .....	12
<b>6</b>	<b>ОСНОВНЫЕ ОБЪЕКТЫ АЛГОРИТМИЧЕСКОГО ЯЗЫКА PASCAL</b> .....	<b>13</b>
6.1	Набор символов языка Pascal .....	13
6.2	Лексемы (слова) языка Pascal .....	13
6.2.1	Специальные символы .....	13
6.2.2	Зарезервированные (ключевые) слова .....	14
6.2.3	Идентификаторы .....	14
6.2.4	Числа .....	16
6.2.5	Строки символов .....	16
6.2.6	Комментарии .....	17
6.2.7	Операторы языка .....	18
<b>7</b>	<b>ПРОГРАММЫ НА ПАСКАЛЕ</b> .....	<b>19</b>
7.1	Общие сведения о программе на Паскале .....	19
7.2	Пример заготовки программы на Паскале .....	22
<b>8</b>	<b>СПИСОК ЛИТЕРАТУРЫ</b> .....	<b>23</b>

## 1 ЦЕЛЬ РАБОТЫ

- ✓ Приобретение начальных навыков работы в интегрированной среде *Borland Pascal*.
- ✓ Знакомство с базовыми понятиями алгоритмического языка *Borland Pascal*.
- ✓ Знакомство со структурой программ на *Borland Pascal*.

## 2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Перед выполнением этой работы следует:

- ✓ изучить правила работы с интегрированной средой программирования *Borland Pascal*, а также ознакомиться со структурой программ на языке *Borland Pascal* [1] [2] [3] [4] [5] [6] [7]
- ✓ Ответить на контрольные вопросы.
- ✓ Войти в свой личный каталог и загрузить систему *Borland Pascal 7.0*.
- ✓ Настроить интегрированную среду для последующей работы. Записать файл конфигурации в личный каталог.
- ✓ Изучить возможности текстового редактора системы *Borland Pascal*.
- ✓ Научиться работать с блоками текста.
- ✓ Создать файл-заготовку для своих программ и сохранить его в своем каталоге. Имя файла - фамилия студента.
- ✓ Добиться, чтобы при компиляции файла-заготовки не было сообщений об ошибках.
- ✓ Оформить отчет и защитить его у преподавателя.

## 3 КОНТРОЛЬНЫЕ ВОПРОСЫ

Ответьте письменно на следующие контрольные вопросы:

- 1) Каковы назначение и возможности интегрированной среды *Borland Pascal*?
- 2) Какие подменю имеет главное меню интегрированной среды *Borland Pascal*?
- 3) Какие команды текстового редактора Вы знаете?
- 4) Что такое блок текста программы и как его выделить? Какие операции с блоками вы знаете?
- 5) Как осуществить контекстный поиск и замену?
- 6) Перечислите правила образования идентификаторов в Паскале. Приведите примеры идентификаторов.
- 7) Что представляет собой строка символов?
- 8) Что представляет собой строка символов?
- 9) Что представляет собой строка комментариев?
- 10) Как запустить программу?

## 4 ОТЧЕТНОСТЬ

Отчет должен быть выполнен в соответствии с [8] и состоять из следующих разделов:

- ✓ Тема и цель работы.
- ✓ Индивидуальное задание.
- ✓ Откомпилированный текст программы-заготовки (в электронном виде).
- ✓ Ответы на контрольные вопросы.

✓ Выводы.

При защите отчета по работе для получения зачета студент должен:

- ✓ уметь отвечать на контрольные вопросы;
- ✓ продемонстрировать навыки работы в среде **Borland Pascal**.

## 5 ИНТЕГРИРОВАННАЯ СРЕДА BORLAND PASCAL

### 5.1 Краткие сведения о работе в системе Borland Pascal

*Borland Pascal* - это эффективный компилятор языка Паскаль с интеллектуальной интегрированной средой (*the Integrated Development Environment - IDE*), простой для изучения и использования. Хороший дизайн, легкость в изучении и использовании выгодно отличают ее от подобных оболочек других известных компиляторов (рис. 5.1)<sup>1</sup>. Все, что нужно для создания, отладки и выполнения программ на Паскале (текстовый редактор, компилятор, редактор связей и отладчик) доступно из IDE *Borland Pascal*.



Рис. 5.1 - Интеллектуальная интегрированная среда Borland Pascal



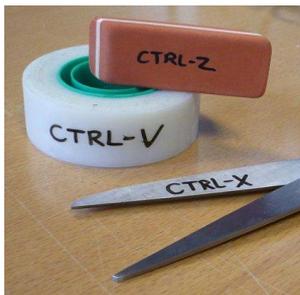
Для запуска IDE *Borland Pascal* нажмите соответствующий ярлык на рабочем столе.

<sup>1</sup> Чтобы из оконного режима перейти в полноэкранный, следует нажать клавиши **Alt+Enter**.

Обычно система *Borland Pascal* размещается на жестком диске в каталоге с именем *BP* (или *BP7*) в нескольких папках (*BIN*, *UNITS*, *BGI* и др.). Для вызова *Borland Pascal* необходимо отыскать в древовидной структуре каталогов компьютера этот каталог и в его папке *BIN* файл *BP.EXE*. Тогда для вызова *Borland Pascal* следует, например, дать команду

**D:\BP\BIN\BP.EXE**

Для доступа к полосе меню **IDE** и выбора команд используют манипулятор «мышь» и клавиши со стрелками.



Многие элементы меню имеют соответствующие «горячие» клавиши<sup>2</sup> (англ. *hotkeys*, ключевые сокращения команд), которые немедленно активизируют некоторую команду или диалоговое окно. Можно также отмечать мышкой сокращения команд в строке статуса. Таблица 5.1 поясняет назначение общих «горячих» клавиш.

Таблица 5.1 - Общие «горячие» клавиши

Клавиши	Элемент меню	Функция
<i>F1</i>	<b>Help</b>	Показывает экран подсказки
<i>F2</i>	<b>File/Save</b>	Сохраняет файл, находящийся в активном окне редактора
<i>F3</i>	<b>File/Open</b>	Появляется диалоговое окно и возможность открыть файл
<i>F4</i>	<b>Run/Go to Cursor</b>	Запускает Вашу программу до строки, на которой стоит курсор
<i>F5</i>	<b>Window/Zoom</b>	Масштабирует активное окно
<i>F6</i>	<b>Window/Next</b>	Делает активным очередное открытое окно
<i>F7</i>	<b>Run/Trace Into</b>	Запускает Вашу программу в режиме отладки с заходом внутрь процедур
<i>F8</i>	<b>Run/Step Over</b>	Запускает Вашу программу в режиме отладки, минуя вызовы процедур
<i>F9</i>	<b>Compile/Make</b>	Компилирует программу текущего окна и подключает к ней подпрограммы из других модулей
<i>F10</i>		Делает полосу меню активной. Полезно, если не работает мышь
<i>Alt-X</i>	<b>File/Exit</b>	Завершает <i>Borland Pascal</i> с выходом в MS DOS

После загрузки система *Borland Pascal* автоматически переходит в режим набора и редактирования текстов программ (команда меню **EDIT**). Работа над исходными текстами программ осуществляется в многочисленных окнах, позволяющих иметь несколько представлений одного и того же файла или разных файлов. Окна редактора можно *передвигать*, *перекрывать* и *изменять их размеры*.

<sup>2</sup> **Горячие клавиши** — сочетания **клавиш** на клавиатуре компьютера, нажимая на которые можно выполнить действия не прибегая к меню и без помощи мыши более быстро и эффективно.

Перед началом работы первым делом **обязательно** установите путь к своему рабочему каталогу с помощью команды Change dir (изменение установленного по умолчанию диска или каталога) в меню File, иначе Вам сложно найти, куда Borland Pascal будет сохранять Ваши программы (Рис. 5.2).

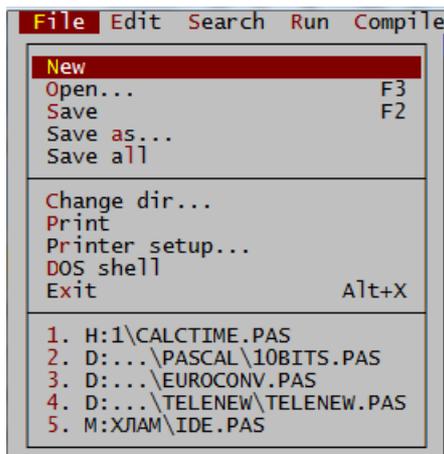


Рис. 5.2 - Меню File

Не рекомендуется работать с *Borland Pascal*, назначив в качестве каталога по умолчанию (т.е. текущего каталога) тот, в котором хранятся перечисленные выше файлы (этот каталог будем называть системным). Во-первых, в таком случае можно ошибочно стереть какой-либо из файлов системы программирования и тем самым нарушить ее работоспособность, а во-вторых, этот каталог очень скоро заполнится другими файлами, прямо не относящимися к *Borland Pascal*.

В большинстве окон Pascal можно заметить такие же элементы управления окнами, что и у Windows: *полоса заголовка, закрывающая кнопка, полосы скроллинга, уголок для изменения размеров окна, кнопка масштабирования, номер окна*. Только они выглядят и расположены иначе, чем в Windows.

Если позволяет память компьютера, Вы можете иметь в распоряжении любое количество открытых окон, но активным в любой момент времени может быть только одно окно. **Активное окно** - это окно, с которым Вы в настоящий момент времени работаете. Любая команда, которую Вы выбрали или текст, который Вы набрали, относится только к активному окну.

Активное окно легко отличить по *двойной рамочке* (см. рис. 5.1). Оно всегда имеет закрывающую кнопку, кнопку масштабирования, кнопки перемещения и уголок изменения размеров. Если Ваши окна перекрываются, то активное окно всегда находится *наверху* остальных. Окно редактора всегда показывает текущие номера строки и *столбца в нижнем левом углу*. Если вы изменили свой файл, то слева от номеров строки и столбца появится знак звездочки "\*".

Таблица 5.2 содержит краткую информацию о том, как можно управлять окнами в *Borland Pascal*. Заметим, что для выполнения этих действий мышка не нужна - клавиатура работает прекрасно.

Таблица 5.2 - - Управление окнами

Чтобы выполнить	Используйте один из этих способов
<b>Открыть Edit окно</b>	1) Выберите <b>File/Open</b> для открытия файла и покажите его в окне; 2) Нажмите <b>F3</b> .
<b>Открыть другие окна</b>	Выберите необходимое окно из меню <b>Window</b> .
<b>Закреть окно</b>	1) Выберите <b>Close</b> из <b>Window</b> меню; 2) Нажмите <b>Alt-F3</b> ; 3) отметьте закрывающую кнопку окна.
<b>Активизировать окно</b>	1) Отметьте в любом месте окна;

Чтобы выполнить	Используйте один из этих способов
	2) Нажмите <b>Alt</b> плюс номер окна (в верхнем правом углу окна); 3) выберите <b>Window/List</b> ; 4) нажмите <b>Alt-O</b> и выберите окно из списка; 5) выберите <b>Window/Next</b> или <b>F6</b> , чтобы сделать активным следующее окно (следующее в порядке, в каком они были открыты); 6) нажмите <b>Alt-F6</b> для активизации предыдущего окна.
<i>Передвинуть активное окно</i>	1) Тащите его заголовок; 2) Нажмите <b>Ctrl-F5 (Window/Size/Move)</b> и используйте клавиши со стрелками, чтобы поместить окно там, где Вам хочется, и нажмите <b>Enter</b> .
<i>Изменить размер активного окна</i>	1) Тащите уголок для изменения размера (или любой угол); 3) Выберите <b>Window/Size/Move</b> и нажимайте <b>Shift</b> одновременно с клавишами со стрелками, и нажмите <b>Enter</b> ; 2) нажмите <b>Ctrl-F5</b> и затем одновременным нажатием используйте <b>Shift</b> и клавиш со стрелками.
<i>Масштабировать активное окно</i>	1) Отметьте кнопку масштабирования в верхнем правом углу окна; 2) дважды отметьте заголовок окна; 3) выберите <b>Window/Zoom</b> ; 4) нажмите <b>F5</b> .

## 5.2 Редактирование программ

Для того чтобы перемещаться по тексту используйте клавиши управления курсором или "мышь". Нажатая предварительно клавиша **Ctrl** «усиливает» действие этих клавиш.

Выделенный текст может быть «вырезан» или скопирован в редактируемую буферную память (**Clipboard**), и неоднократно восстановлен затем в любом окне<sup>3</sup>.

**Работа с блоками текста.** Блок текста - это любой объем текста (от одного символа до сотен строк), который выделен на экране. В каждый момент времени в окне может быть выделен только один блок.

**Выделение блока.** Для выделения блока текста существует, по крайней мере, три способа:

- ✓ Буксируйте "мышь" по тексту, который вы хотите выделить.
- ✓ Переместите свой курсор в начало блока текста, нажмите клавишу **Shift** и, удерживая ее, переместите курсор к концу блока с помощью клавиш стрелок.
- ✓ Щелкните "мышью" в начале блока, переместите курсор к концу блока с помощью клавиш стрелок, затем, нажав **Shift**, снова щелкните кнопкой "мыши".

<sup>3</sup> Кстати, таким же образом в режиме **Help** можно выделить фрагмент текста помощи (это может быть, в частности, пример использования какой либо стандартной подпрограммы или функции), и перенести его затем в свою программу для дальнейшего использования.

**Вырезание, копирование и вставка блоков.** После выбора блока его можно скопировать, переместить или записать в файл.

Чтобы вырезать выделенный текст, нажмите клавиши **Shift+ Del** или выберите команды **Edit/Cut**. Выделенный блок удаляется из текста и помещается в буфер вырезанного изображения ("карман") - временную область памяти.

Для копирования выделенного текста нажмите клавиши **Ctrl+ Ins** или выберите команду **Edit/Copy**. Выделенный блок остается в тексте, а его копия помещается в буфер вырезанного изображения и готова для вставки в другом окне редактирования.

Чтобы вставить (скопировать) текст, содержащийся в буфере вырезанного изображения, в активное окно, нажмите клавиши **Shift+ Ins** или выберите команду **Edit/Paste**. Блок, содержащийся в буфере, вставляется в текущей позиции курсора.

Для стирания (удаления) выделенного текста нажмите клавиши **Ctrl + Del** или выберите команду **Edit/Clear**. Выделенный блок удаляется из текста, и копия не помещается в буфер вырезанного изображения. Единственный способ восстановления или удаления текста состоит в применении команды **Edit/Undo**.

Назначение некоторых наиболее употребляемых «горячих» клавиш текстового редактора можно понять из *таблицы 5.3*.

Таблица 5.3 - "Горячие" клавиши редактирования

Клавиши	Элемент меню	Функция
<i>Alt + Backspace</i>	<b>Edit/Undo</b>	Отказ от последней операции
<i>Ins</i>		Включение/выключение режима вставки
<i>Delete (Del)</i>		Удаляет символ, отмеченный курсором
<i>Backspace</i>		Удаляет символ слева от курсора
<i>Ctrl+Y</i>		Удаляет строку, отмеченную курсором
<i>Ctrl+ Q Y</i>		Удаляет от курсора до конца строки
<i>Ctrl+ T</i>		Удалить слово справа от курсора
<i>Ctrl+ Del</i>	<b>Edit/Clear</b>	Удаляет заранее помеченный блок из окна и не помещает его в карман (окно Clipboard)
<i>Ctrl+ Ins</i>	<b>Edit/Copy</b>	Копирует заранее помеченный блок в карман
<i>Shift+ Del</i>	<b>Edit/Cut</b>	Перемещает заранее помеченный блок в карман (с удалением из исходного текста)
<i>Shift+ Ins</i>	<b>Edit/Paste</b>	Вставляет отмеченный блок из кармана в текущую позицию курсора активного окна
<i>Ctrl+ K B</i>		Отметка начала блока
<i>Ctrl+ K K</i>		Отметка конца блока
<i>Ctrl+ K T</i>		Выделение слова
<i>Ctrl+ K C</i>		Копирование блока
<i>Ctrl+ K V</i>		Перемещение блока
<i>Ctrl+ K Y</i>		Уничтожение блока

### 5.3 Работа со справочной системой



Для работы со встроенной контекстно-ориентированной справочной информацией также используйте соответствующие "горячие" клавиши (Таблица 5.4).

Таблица 5.4 - "Горячие" клавиши встроенной справочной информации

Клавиши	Элемент меню	Функция
F1	<b>Help/Contents</b>	Открывает контекстно-ориентированный экран справочной информации
F1 F1	<b>Help/Help on Help</b>	Вызывает справочную информацию по справочной информации (нужно нажать только <b>F1</b> , если Вы уже находитесь в системе справочной информации)
Shift-F1	<b>Help/Index</b>	Вызывает оглавление справочной информации
Alt-F1	<b>Help/Previous Topic</b>	Показывает предыдущий экран справочной информации
Ctrl-F1	<b>Help/Topic Search</b>	Вызывает информацию по конкретному элементу в программе, отмеченному курсором (только в редакторе)

**Перемещение в справочной системе.** Если вы посмотрите на справочный экран, то увидите текст, который выводится цветом, отличным от окружающего текста. Это ссылки. Вы можете использовать ссылки для вывода нового справочного экрана, содержащего новую информацию по соответствующей теме. Выберите один из следующих методов:

- ✓ Дважды щелкните на ссылке "мышью".
- ✓ Если справочный экран не содержит командных кнопок, нажимайте повторно клавишу **Tab**, пока не будет подсвечена ссылка, затем нажмите клавишу **Enter**.
- ✓ Если справочный экран представляет собой диалоговое окно с командными кнопками:
  - ✓ если вы используете "мышь", щелкните "мышью" на кнопке перекрестных ссылок **Cross-ref**;
  - ✓ если вы используете клавиатуру, нажмите **Enter**; при этом по умолчанию выбирается кнопка **Cross-ref**.
  - ✓ Чтобы выбрать другую кнопку, нажимайте повторно клавишу **Tab**, пока не будет подсвечена нужная кнопка, затем нажмите клавишу **Enter**.

Выводится новый справочный экран **Help** с информацией по выбранной теме. Вы перешли в новое место справочной системы. На этом экране вы можете видеть другие ссылки, которые можно выбирать для получения дальнейшей информации.

Чтобы вернуться к предыдущему экрану **Help**, выберите команду **Help/Previous Topic** или нажмите клавиши **Alt+F1**.

**Запрос помощи.** Доступ к справочнику *Help* можно получить несколькими способами:

- ✓ Выбрав команду *Help* в строке меню или нажав **Alt+ H** для вывода меню *Help*. В меню *Help* вы можете выбрать вывод экрана оглавления *Contents*, экрана тематического указателя *Index* по всей справочной системе, детальную информацию по теме, на которую указывает курсор в окне редактирования или справку по работе со справочной системой *Help*.

- ✓ Для вывода экрана тематического указателя *Borland Pascal Help Index* нажмите клавиши **Shift+F1**. Экран тематического указателя *Index* аналогичен тематическому указателю книги. Однако, вместо того, чтобы перевернуть страницы для получения информации по нужной теме здесь нужно дважды щелкнуть на ней кнопкой "мыши" или перейти к теме с помощью клавиши **Tab** и нажать **Enter**.

- ✓ Нажав клавишу **F1**, Вы получите контекстно-зависимую информацию, зависящую от того, что вы делаете в этот момент - редактируете, отлаживаете программу, выбираете параметры меню и т.д. Если вы находились в диалоговом окне, то увидите справочный экран по тому параметру, который выделен в момент нажатия **F1**.

- ✓ Выбрав командную кнопку *Help* в диалоговом окне. При выборе командной кнопки *Help* вы получите информацию о диалоговом окне.

- ✓ Поместив курсор на термин в окне редактирования, и выбрав *Topic Search*. Используйте любой из следующих методов:

- ❖ нажмите клавиши **Ctrl+F1**;
- ❖ выберите команду *Help/Topic Search*; удерживая клавишу **Ctrl**, щелкните правой кнопкой "мыши" (комбинация нажатий **Ctrl**+ правая кнопка "мыши" должна быть предварительно настроена – выберите команду *Options/Environment/Mouse* и выберите команду *Topic Search*).
- ❖ выберите команду *Topic Search* в локальном меню окна редактирования (нажмите клавиши **Alt+F10** для вывода на экран локального меню или щелкните правой кнопкой "мыши").

В справочном экране выводится информация о термине, на котором установлен курсор в активном окне.

**Копирование примеров исходного кода.** Справочная система *Help* содержит примеры кода для каждой процедуры и функции. Вы можете скопировать эти примеры из справочной системы в окно редактирования. Выполните следующие шаги:

- ✓ Выведите справочный экран по интересующей вас процедуре или функции.
- ✓ Прокрутите справочное окно, пока не увидите в нем пример исходного кода.
- ✓ Для вывода локального меню *Help* нажмите клавиши **Alt+F10** или щелкните правой кнопкой "мыши".

- ✓ Скопируйте пример:

- ❖ для копирования всего примера выберите команду *Copy Example*;
- ❖ для копирования части исходного кода подсветите то, что вы хотите скопировать, и выберите команду *Copy*.

✓ Вернитесь в окно редактирования и выберите команду **Edit/Paste**, нажмите клавишу **Shift+ Ins** или выберите команду **Paste** в локальном меню окна редактирования.

**Загрузка других справочных файлов.** Интегрированная среда позволяет вам подключать к справочной системе **Help** другие справочные файлы. Например, если вы хотите получать справочную информацию по **Turbo Vision**, то можете загрузить справочный файл **Turbo Vision. IDE** объединяет тематические указатели справочных файлов, что позволяет вам получить доступ в экране **Index** как к обычной справочной системе **Help**, так и к справочной системе **Turbo Vision**.

Чтобы загрузить новую справочную систему **Help**, выполните следующие шаги:

✓ Выберите команду **Help/Files** (Справочник/Файлы). Выводится диалоговое окно установки справочных файлов **Install Help Files**.

✓ Выберите команду **New** (Новый). Выводится диалоговое окно **Help Files** (Справочные файлы). Все справочные файлы в вашем каталоге **BIN** с расширением **.TPH** выводится в блоке списка. Если вы не видите справочных файлов, измените каталог **BP\BIN**.

✓ Дважды щелкните "мышью" на справочном файле, который вы хотите включить в справочную систему, или выделите его и нажмите клавишу **Enter**.

✓ Выберите командную кнопку **OK**.

✓ Выберите команду **Help/Index** или нажмите клавиши **Shift+F1**. В строке состояния вы можете увидеть краткое сообщение, показывающее индексирование и слияние. После завершения слияния вы можете прокрутить справочный экран и увидеть, что вам доступны все темы в выбранных справочных файлах. Тематические указатели остаются объединенными в течение текущего сеанса. Если в диалоговом окне **Preferences (Options/Environment/Preferences)** вы установили параметр **Environment Auto Save**, то тематические указатели будут оставаться объединенными и в следующих сеансах. Если вы не сохранили свою операционную среду, то при следующем запуске IDE тематический указатель **Help** возвращается в исходное состояние. Другими словами, объединение тематических указателей не будет сохраняться от одного сеанса к другому.

Если вы не хотите видеть на экране тематического указателя **Index** конкретные записи **Help**, то можете "удалить" справочный файл:

✓ Выберите команду **Help/Files**.

✓ Выберите имя справочного файла, который вы больше не хотите просматривать.

✓ Выберите команду **Delete**.

✓ Выберите командную кнопку **OK**.

**Выход из справочника Help.** Чтобы закрыть окно **Help** и вернуться в свою прикладную программу вы можете выбрать один из следующих методов:

✓ Нажать клавишу **Esc**.

✓ Щелкнуть "мышью" на элементе закрытия окна **Help**.

✓ Щелкнуть "мышью" вне окна **Help**.

Если вы хотите вновь вывести предыдущий справочный экран, нажмите клавиши **Alt+F1**.

#### 5.4 Компиляция, запуск и отладка программ

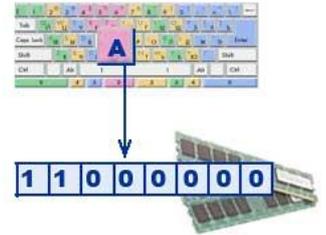
Для компиляции, запуска и отладки программ используйте меню или соответствующие "горячие" клавиши (Таблица 5.5).

**Таблица 5.5 – «Горячие» клавиши отладки/запуска**

Клавиши	Элемент меню	Функция
<i>Alt-F9</i>	<b>Compile/Compile</b>	Компилирует последний файл в редакторе
<i>Ctrl-F2</i>	<b>Run/Program Reset</b>	Переустанавливает выполняемую программу
<i>Ctrl-F4</i>	<b>Debug/Evaluate/Modify</b>	Позволяет запросить значение любой переменной или выражения при отладке программы «по шагам»
<i>Ctrl-F7</i>	<b>Debug/Add Watch</b>	Добавляет выражение или переменную в окно <b>Wath</b> для просмотра изменения их значений при отладке программы «по шагам»
<i>Ctrl-F8</i>	<b>Debug/Toggle Breakpoint</b>	Устанавливает или очищает условные точки прерывания (остановки в программе, после которых намереваются выяснить значение некоторых переменных и перейти к отладке «по шагам»)
<i>Ctrl-F9</i>	<b>Run/Run</b>	Компилирует и запускает программу
<i>F4</i>	<b>Run/Go To Cursor</b>	Компилирует и запускает программу до позиции курсора
<i>F7</i>	<b>Run/Trace Into</b>	Осуществляет прослеживание программы «по шагам» и внутри процедур
<i>F8</i>	<b>Run/Step Over</b>	При прослеживании работы программы «по шагам» перескакивает через вызовы процедур
<i>F9</i>	<b>Compile/Make</b>	Компилирует и редактирует связи программы

## 6 ОСНОВНЫЕ ОБЪЕКТЫ АЛГОРИТМИЧЕСКОГО ЯЗЫКА PASCAL

### 6.1 Набор символов языка Pascal



Программа на языке Pascal формируется из *символов*:

прописных букв латинского алфавита	коды ASCII от <b>65</b> до <b>90</b>	<b>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</b>
строчных букв латинского алфавита	коды ASCII от <b>97</b> до <b>122</b>	<b>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</b>
символа подчеркивания	код ASCII <b>95</b>	<b>-</b>
арабских десятичных цифр	коды ASCII от <b>48</b> до <b>57</b>	<b>0 1 2 3 4 5 6 7 8 9</b>
специальных символов	в скобках указан код ASCII	<b># (35) \$ (36) ' (39) ( (40) ) (41) * (42) + (43) , (44) - (45) . (46) / (47) : (58) ; (59) &lt; (60) = (61) &gt; (62) @ (64) [ (91) ] (93) ^ (94) { (123) } (125)</b>
символ пробела	код ASCII <b>32</b>	на экране не отображается
управляющие символы	коды ASCII от <b>0</b> до <b>31</b>	на экране не отображаются

Остальные символы расширенного набора кодов **ASCII** могут использоваться в символьных и строковых константах, а также в комментариях, что позволяет организовать диалог пользователь-компьютер *на русском языке*.

Русские буквы в программе обязательно должны заключаться в апострофы, например **'Текст на русском языке в Паскаль заключается в апострофы'**

### 6.2 Лексемы (слова) языка Pascal



**Лексемы** – минимальные значимые единицы текста программы. Рассмотрим основные лексемы языка Pascal.

#### 6.2.1 Специальные символы

<b>Одиночные символы</b>	<b>+ - * / &lt; = &gt; [ ] ( ) { } . , : ; ^ @ # \$</b>	
<b>Составные символы (нельзя разделять пробелами)</b>	<b>:=</b>	<b>присвоить</b>
	<b>..</b>	<b>диапазон</b>
	<b>&lt;=</b>	<b>меньше или равно</b>
	<b>&gt;=</b>	<b>больше или равно</b>
	<b>&lt;&gt;</b>	<b>не равно</b>
	<b>{ } (* *) ( . ) [ . ]</b>	<b>символы начала и конца комментариев</b>

### 6.2.2 Зарезервированные (ключевые) слова

**Зарезервированные слова** имеют строго определенное назначение, которое **не может быть изменено** в программе<sup>4</sup>:



**and asm array begin case const constructor destructor div do downto else end exports file for function goto if implementation in inherited inline interface label library mod nil not object of or paced procedure program record repeat set shl shr string then to type unit until uses var while with xor**

### 6.2.3 Идентификаторы

Чтобы программа решения задачи обладала свойством массовости, следует вместо конкретных значений величин использовать их обозначения, чтобы иметь возможность изменять их значения по ходу выполнения программы. Для обозначения переменных и постоянных величин, различных процедур, функций, объектов используются имена – **идентификаторы**, устанавливающие соответствие между объектом и некоторым набором символов.



**Идентификаторы** – последовательности латинских букв, цифр и знаков подчеркивания, используемые в качестве имен констант, типов, переменных, процедур, функций, модулей программ, меток и полей в записях. Идентификаторы могут иметь произвольную длину, однако значимыми являются только *первые 63 символа*. **Первый символ – обязательно не цифра!**

**Правильные идентификаторы:**

Summa  
Lab\_8\_  
Stop  
\_1\_2\_3\_  
MyConst  
My\_Variable

**Неправильные идентификаторы:**

**Сумма** – ошибка, идентификатор содержит буквы русского алфавита  
**MyConst#** - содержит недопустимый символ #  
**My-variable** - содержит недопустимый символ –  
**1\_2\_3** – начинается с цифры  
**Lab 3** – пробел между частями идентификатора

<sup>4</sup> Не допускается описание пользовательских идентификаторов, идентичных зарезервированным словам!

**В языке *Pascal* не делается различие между символами на верхнем и нижнем регистрах (прописными и строчными)!**

Например, следующие идентификаторы будут обозначать одно и то же:

**Summa summa SUMMA sUMMA SuMmA**

<b>Стандартные идентификаторы</b>	<p>Служат для обозначения заранее определенных разработчиками языка типов данных, констант, процедур и функций, например: <i>integer, Sin, Cos, Ln, Sqr, Sqrt, Read, Readln, Write, Writeln</i>.</p> <p>В этом примере стандартный идентификатор <i>Sin</i> вызывает функцию, вычисляющую синус заданного угла. <i>Read, Readln</i> вызывают процедуру, организующую ввод данных, <i>Write, Writeln</i> вызывают процедуру, организующую вывод данных.</p> <p>Любой из стандартных идентификаторов, в отличие от зарезервированных слов, можно переопределить, но это чаще всего приводит к ошибкам. <b>Не переопределяйте стандартные идентификаторы</b>, чтобы не запутать себя и компьютер!</p>
<b>Пользовательские идентификаторы</b>	<p>Применяются для обозначения меток, констант, переменных, процедур и функций, определенных самим программистом. При этом идентификаторы в программе должны быть уникальными, т. е. в каждом блоке программы один идентификатор не может использоваться для обозначения более чем одной переменной или постоянной величины, и т. д.</p>

Иногда в разных модулях (например, **Unit1, Unit2, Unit3**) описаны *одинаковые* имена объектов (например, **Error**).

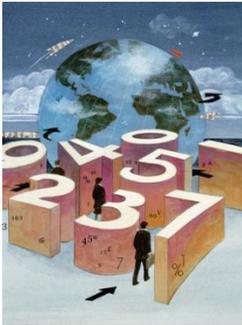
Для уточнения, из какого именно модуля используется объект, используются **квалифицируемые (уточненные) идентификаторы**, в которых перед именем переменной ставится модуль, содержащего этот идентификатор. При этом оба идентификатора разделяются точкой:

**Unit1.Error Unit2.Error Unit3.Error**

Уточняться могут: идентификаторы **меток, констант, переменных, типов данных, а также имена процедур и функций**.

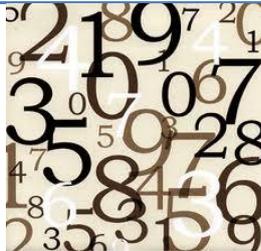
Уточненные идентификаторы используются также при работе **с записями и объектами**.

## 6.2.4 Числа



В Pascal используются **числа**:

- ✓ **целые десятичные,**
- ✓ **вещественные десятичные,**
- ✓ **целые шестнадцатеричные.**



1	2	3	C
4	5	6	D
7	8	9	E
A	0	B	F

**Целые десятичные числа** записываются стандартным образом и должны находиться в диапазоне от **-2147483648** до **2147483648**:

27  
+66  
-49  
9  
-6435

**Вещественные десятичные числа** записываются либо в виде обычной десятичной дроби, либо в показательной (инженерной) форме с основанием 10. Вместо основания 10 ставится строчная **e** или прописная **E** буква, за которой непосредственно указывается показатель степени.

6.28  
-234.567  
+33.5  
2.83e14  
-1E+27  
0.4569e-10

**Разделительный символ – обязательно точка!**

Перед **шестнадцатеричными целыми числами** ставится **символ \$**. Знак шестнадцатеричного числа (плюс или минус) определяется самой формой записи и *зависит от значения старшего разряда* (бита) двоичного представления числа.

\$0  
\$4E  
\$FFFF  
\$A10  
\$CBADF9

## 6.2.5 Строки символов

Для представления **текстовой информации** в Pascal используются **строки символов**, которые представляют собой последовательности символов из расширенного набора символов **ASCII**, заключенные в одинарные кавычки, например:

**'Delphi – очередная версия Borland Pascal'**

**'Проектирование РЭС'**

**'Автор языка Pascal - Н. Вурт'.**

Вся последовательность символов обязательно должна располагаться **в одной строке!**

Если между кавычками нет ни одного символа, то такая строка считается пустой (нулевой длины):

**" - пустая строка**

Кроме того, любой символ в строке **ASCII** (в том числе и те символы, которых нет на клавиатуре) может быть представлен в виде целого десятичного числа (от 0 до 255), перед которым ставится символ #:

СОДЕРЖАНИЕ СПРАВОЧНОЙ СИСТЕМЫ	
Как пользоваться справочной системой Меню и "горячие клавиши" Зарезервированные слова	
Функции и процедуры	О переводе
Встроенный ассемблер Командная строка Отладчик Директивы Сообщения об ошибках Object Browser Object Windows	Команды редактора Примеры программ Опции загрузки Turbo Vision Модули Словарь Windows API

**#182** символ псевдографики « || » - одинарное ответвление справа в таблице с двойной рамкой, выполненной в текстовом режиме



**#7** символ «короткий гудок» - позволяет при выводе чего-либо на экран дать сигнал

Перевод  
| строки

**#10** символ «перевод строки» - перевод курсора на строку ниже



**#13** символ «возврат каретки» - перевод курсора в начало строки

### 6.2.6 Комментарии

**Комментарии** в *Pascal* представляют собой фрагменты текста программы, ограниченные слева символом { или составным символом (\*, а справа - } или составным символом \*). В программе комментарии выполняют чисто информационную функцию и служат для описания работы алгоритма программы, отдельных подпрограмм, назначения типов, переменных и т.п.

{Комментарием может быть любая последовательность символов, ограниченная с двух сторон фигурными скобками, независимо от того, сколько строк она будет занимать!}

(\* Последовательность символов в круглых скобках со звездочками также является комментарием \*)

Если непосредственно после открывающего символа комментария следует знак \$, то комментарий воспринимается как директива компилятору:

{\$I+}

(\*\$F-\*)

{\$I My\_file.pas}

### 6.2.7 Операторы языка



Группа слов, имеющая определенный смысл, называется **словосочетанием**

В языке программирования словосочетание, состоящее из слов и символов и задающее правило вычисления некоторого значения, называется выражением. Ниже приведены (Таблица 6.1) примеры записи некоторых математических выражений на Pascal.

Таблица 6.1 – Примеры записи арифметических выражений на Pascal

Математическая запись	Запись выражений на Pascal
$\frac{xy}{z}$	<code>x*y/z</code>
$\frac{x}{yz}$	<code>x/(y*z)</code> или <code>x/y/z</code>
$\frac{a^2 + \sqrt[3]{b}}{bc}$	<code>(sqr(a) + exp(ln(b) / 3)) / (b*c)</code>
$\frac{a_{i+1} + b_{i-1}}{2\pi y}$	<code>(a[i+1] + b[i-1]) / (2*pi*y)</code>
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	<code>(-b + sqrt(sqr(b) - 4*a*c)) / (2*a)</code>
$a^b$	<code>exp(b*ln(a))</code>

Минимальная конструкция языка, представляющая собой законченную мысль, называется **предложением**.

В языках программирования предложение, задающее полное описание некоторого выполняемого действия, называется **оператором**.

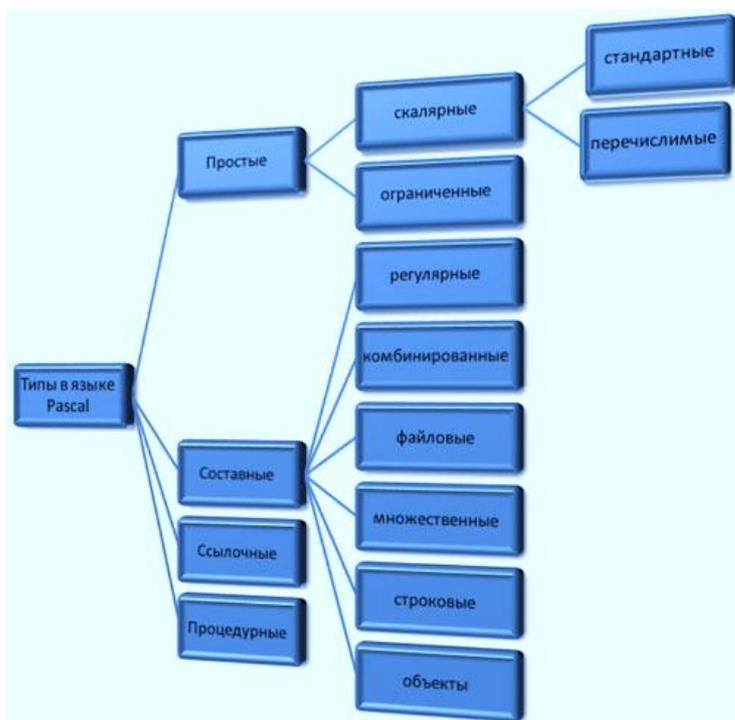
**Оператор** - это составная часть программы, фраза алгоритмического языка, предписывающая определённый порядок преобразования информации.

Любая программа содержит операторы. Операторы образуют программу так же, как обычные предложения образуют текст книги.

В **простых операторах** можно присваивать значение, активизировать процедуру или функцию или *передать управление* на любой выполняемый оператор.

**Структурные операторы** могут быть *составными* и содержать несколько операторов, *операторы цикла* или *условные операторы*, управляющие логикой программы.





Предложение, описывающее структуру и организацию данных – объектов языка, над которыми производятся различные действия, называется **описанием**.

В языке Pascal любые объекты (константы, переменные и т.п.), а также процедуры и функции, используемые программой, предварительно должным образом необходимо определить в разделах описания данных перед использующим их программным блоком.

Для всех данных нужно указать их тип, т.е. определить множество допустимых значений, которые может тот или иной объект, а также множество допустимых операций, которые применимы к нему. Кроме того, тип определяет формат внутреннего представления данных в памяти компьютера.

Чтобы научиться писать программы для компьютера правильно, необходимо изучить синтаксис языка программирования (правила записи его конструкций) и его семантику (смысл и правила использования этих конструкций).

## 7 ПРОГРАММЫ НА ПАСКАЛЕ

### 7.1 Общие сведения о программе на Паскале



Программа реализует алгоритм решения задачи. В ней программист записывает последовательность действий, выполняемых над определенными данными с помощью определенных операций для реализации заданной цели.

Основными характеристиками программы являются: точность полученного результата, время выполнения и объем требуемой памяти. О соответствии этих показателей решаемой задаче и возможностям компьютера должен позаботиться сам программист. В большинстве случаев определяющим требованием является точность. Ограничения по объему памяти и времени выполнения носят менее жесткий характер.



Программа на языке Pascal состоит из строк. Набор текста программы осуществляется с помощью встроенного редактора текстов системы программирования Pascal или любого другого редактора. В первом случае программа может после выхода из редактора (при нажатии клавиши F10) в главное меню компилироваться и выполняться. Во втором случае программу следует записать в файл с расширением *\*.pas* и затем вызвать для компиляции и выполнения в интегрированной среде программирования Pascal.

Набирая текст программы, программист может произвольно располагать строки на экране. Строка может начинаться с любой колонки, и это позволяет самим программистом устанавливать величину отступа от левой границы экрана для каждой строки, чтобы получить наиболее удобный для чтения текст программы. Количество операторов в строке произвольно, но если в строке записывается один оператор, то такая программа легче читается и проще отлаживается.

Существуют многообразные схемы написания программ на языке Pascal. Все они различаются количеством отступов слева в каждой строке и различным использованием прописных букв.

Наиболее продуктивной выглядит следующая схема:

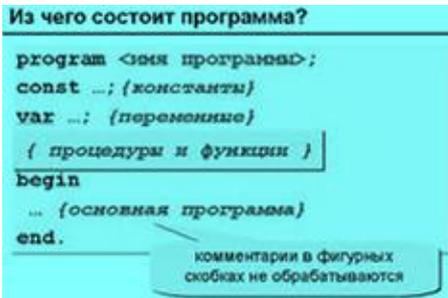
- зарезервированные слова *program*, *procedure*, *function* пишутся строчными буквами и если они записаны правильно, то на экране будут выделены белым цветом;
- имена констант, переменных, процедур, функций начинаются с прописных букв;
- операторы записываются только строчными буквами;
- логически подчиненные структуры записываются на одну строку ниже и на две позиции правее по отношению к более старшим.

Такая схема записи создает условия для лучшего понимания программы и существенно более быстрого обнаружения в ее тексте ошибок.

Заметим, что максимальный размер программы на Pascal ограничен. Компилятор позволяет обрабатывать программы и библиотечные модули, в которых объем данных и генерируемый машинный код не превышают 64 Кбайт каждый. Если программа требует большего количества памяти, следует использовать библиотечные модули (.TPU-файлы) или оверлейные структуры<sup>5</sup>.

---

<sup>5</sup> **Оверлеи** – части исполняемой программы, которые используют одну и ту же область оперативной памяти. В каждый момент времени в памяти может находиться только один оверлей, в зависимости от выполняемой функции. В процессе выполнения программы эти части могут замещать друг друга в памяти.



Обязательной частью является лишь тело программы, которое начинается словом **begin**, а заканчивается словом **end** с точкой.

Операторы в Паскале разделяются точкой запятой. Заголовок программы является хотя и необязательным, но желательным элементом и состоит из зарезервированного слова **program** и идентификатора - имени программы, за которым следует точка с запятой.

Порядок объявлений и описаний не регламентируется.

Областью действия объектов является блок, в котором они описаны, и все вложенные в него блоки.

Блок может содержать в себе другие блоки. Блок состоит из двух частей: **описательной** и **исполнительной**.

Первая часть может отсутствовать, если нет используемых в программе данных. Без второй блок не имеет смысла.

Блок, который не входит ни в какой другой блок, называется **глобальным**.

Если глобальный блок содержит другие блоки, то они называются **локальными**.

**Глобальный блок** — это основная программа, он должен присутствовать в любом случае. Локальные блоки — это процедуры и функции, их присутствие необязательно.

Объекты программы (типы, переменные, константы и т. д.) тоже называются глобальными и локальными.

Все **объекты**, используемые в программе на Pascal должны быть описаны в *разделе описаний*, который состоит из подразделов описания констант (**const**), типов (**type**), переменных (**var**), процедур (**procedure**) и функций (**function**), которые могут повторяться *многократно* в рамках раздела описаний *в любом порядке*.

Блочная структура обеспечивает структуризацию программ на уровне исходных текстов. В идеальном случае программа на языке Pascal состоит из процедур и функций, которые вызываются для выполнения из раздела операторов основной программы.

**Если в описании какого-либо элемента В (константы, типа, переменной, процедуры или функции) используется элемент А (константа, тип, переменная и т.д.), то элемент А должен быть описан перед элементом В!**



## 7.2 Пример заготовки программы на Паскале

```

* Program Имя_Программы; {Заголовок программы}
* {Назначение программы – Заготовка для создания новых программ}
* {Автор – Иванов И.И., студент группы 232-1}
* {Вариант задания - № 77}
* {Условия задания - ...}
* {Раздел указания используемых модулей – библиотек подпрограмм и описаний данных}
* Uses
    Имя1, Имя2, Имя3; {список имен используемых модулей}

    {Разделы описаний данных}
* Label ... ; {список используемых меток}
* Const ... ; {список используемых констант}
* Type ...; {список используемых меток}
* Var ... ; {список используемых меток}

    {Описания разработанных подпрограмм: процедур и функций}
* Procedure Имя(Список формальных параметров); {Заголовок процедуры}
* begin
    операторы процедуры }      тело процедуры
* end;

* Function Имя(Список формальных параметров): Тип результата; {Заголовок функции}
* begin
    операторы функции }      тело функции
* end;

begin {Начало основного блока программы}
    операторы программы, описывающие выполняемые программой действия
end. {Конец основного блока программы}

```

**Примечание.** \* - необязательный оператор

Программа на Паскале может использовать описания данных, а также подпрограммы и функции из программных *модулей*. Модуль (*unit*) можно рассматривать как библиотеку подпрограмм, которую может использовать Ваша прикладная программа. Как и программа, он имеет заголовок (который называется заголовком модуля) и основной блок, ограниченный *begin* и *end*. Программный код, который вы помещаете в модуль, может использоваться любой программой (в том числе и другим модулем).

Основной блок любой программы на Паскале может включать в себя строку *uses*, позволяющую программе использовать один или более модулей. К примеру, если Вы хотите изменять цвета выводимого на экран текста, то в Вашей программе необходимо использовать *стандартный* модуль *Crt* с библиотекой подпрограмм работы с экраном, являющийся *частью библиотеки Паскаля*. Для этого в нем имеются необходимые подпрограммы. Строка *uses Crt* подключит модуль *Crt* в Вашу программу.

Чтобы разделить большие программы на логически связанные фрагменты, Вы можете также написать и свои *собственные модули*.

## 8 СПИСОК ЛИТЕРАТУРЫ

1. **Кобрин, Ю.П.** Интегрированная среда Borland Pascal / Приложение к лабораторной работе "Работа в интегрированной среде Borland Pascal". - Томск : ТУСУР, кафедра КИПР, 2012. - 36 с.
2. **Рапаков Г.Г., Ржеуцкая С.Ю.** *Программирование на языке Pascal*. — СПб. : БХВ-Петербург, 2004. - 480 с.
3. **Фаронов, В.В.** Программирование на персональных ЭВМ в среде Турбо-Паскаль. ; . - М. : Изд-во МГТУ, 1990.-580 с.
4. **Павловская, Т.А.** Паскаль. Программирование на языке высокого уровня: Учебник для вузов. — СПб. : Питер, 2007. — 393 с.
5. **Фаронов, В.В.** Турбо Паскаль 7.0. Начальный курс. Учебное пособие. -М. : ОМД Групп, 2003. - 616 с.
6. **Рютген Т., Франкен Г.** *Турбо Паскаль 7.0*. — К. : Торгово-издательское бюро ВНУ, 1996-448 с.
7. **Попов, В.Б.** *Паскаль и Дельфи. Самоучитель*. — СПб. : Питер, 2004. — 544 с.
8. **Епанешников А., Епанешников В.** . *Программирование в среде Turbo Pascal 7.0*. - М. : ДИАЛОГ-МИФИ, 1993. - 288 с.