3"l

Введение

Цель настоящего пособия - дать достаточно полное описание

существующего русского варианта языка программирования

LogoWriter, реализованного на компьютерах IBM PC. Поскольку

LogoWriter - язык, предназначенный, главным образом, для исполь-

зования в школьном обучении, то основными пользователями языка

являются преподаватели и дети. Данное пособие адресовано препо-

давателям, применяющим LogoWriter в процессе обучения. Пособие

не предназначено для систематического изучения программирования

на языке LogoWriter или для изучения методики преподавания этого

языка.

Мы предполагаем, что читатель имеет минимальные навыки ра-

боты с компьютером IBM PC и знаком с простейшими понятиями опе-

рационной системы MS DOS. Все необходимые сведения об операцион-

ной системе можно найти или в фирменном руководстве по MS DOS

или ,например, в книге Брябрин В.М. "Программное обеспечение

персональных ЭВМ, М., Наука", 1988. Описываемая реализация языка

LogoWriter может работать на компьютере IBM PC (и совместимых с

ним) при объеме оперативной памяти не менее 128 Кб под управле-

нием MS DOS версии 2.0 и выше. Размер графического изображения -

320х152 точек, текстовый режим - 40 символов в строке. В графи-

ческом режиме может быть использовано 4 цвета при построении

изображения и 16 цветов фона. В текстовом режиме для задания

цвета текста может быть использовано 4 цвета (список цветов при-

веден в приложении 5).

Сведения о языке LogoWriter излагаются, в основном, в соот-

ветствии с руководством "LogoWriter. Reference Guide", Logo

Computer Systems Inc, 1986.

Пособие состоит из трех частей.

Часть I. Структура системы LogoWriter.

Вводятся основные понятия языка LogoWriter, описываются ви-

ды используемых наборов данных на внешних носителях (файлов).

Описывается общая структура сеанса работы с системой: вызов

программы, структура главного меню (оглавления), структура обыч-

ного листа и структура листа форм, выход из программы. Описыва-

ются режимы работы программы.

Часть II. Примитивы языка LogoWriter.

Описывается синтаксис инструкций, команд и процедур языка

LogoWriter. Приводится алфавитный список зарезервированных тер-

минов (примитивов) языка, снабженных комментариями и примерами

использования примитивов.

Часть III. Приложения.

Приведенные приложения содержат дополнительные сведения о

работе системы, такие как использование специальных листов, спи-

сок сообщений системы и прочее. Здесь же содержится англо-русс-

кий словарь примитивов и глоссарий.

- 2 -

Часть I. Структура системы LogoWriter.

В первом разделе этой части приводятся основные понятия

системы LogoWriter. В разделе 2 описываются файлы, используемые

в системе LogoWriter: загрузочный модуль LOGOWR.COM, лист форм,

обычные листы, текстовые файлы и файлы графики. В разделе 3 схе-

матически описывается сеанс работы с системой. В разделе 4 опи-

сываются различные режимы работы системы LogoWriter.

1. Предварительные сведения.

Одно из основных понятий системы LogoWriter - лист. Листом

называется совокупность графической информации (картинка разме-

ром 320х152 точки) и текстовой информации объемом до 4061 симво-

ла. В системе LogoWriter имеются средства, позволяющие записы-

вать лист на диск в виде специального файла, считывать подобный

файл в оперативную память и редактировать лист. Обычно работа с

системой состоит в том, что Вы считываете лист с диска (или на-

чинаете работать с пустым листом, если Вы хотите создать новый

лист), вносите в него те или иные изменения и записываете его на

диск. Разработчики системы стремились к тому, чтобы работа с

листом в системе LogoWriter была похожа на работу с обычным лис-

том бумаги, поэтому часто удобно представлять себе графику и

текст, содержащиеся в листе LogoWriter'а в виде картинки и над-

писей на листе бумаги. Как и следовало ожидать, такая аналогия

не является абсолютно полной, поэтому читателю не следует удив-

ляться, что операции, которые можно выполнять с листом бумаги и

операции, которые можно выполнять с листом LogoWriter'а совпада-

ют лишь в общих чертах.

Лист состоит из двух частей (он имеет две стороны) - лице-

вой и изнаночной. Лицевая сторона предназначена для рисования

картинок и для написания произвольных текстов. Изнаночная сторо-

на предназначена для записи текстов процедур (программ и подп-

рограмм на языке LogoWriter), причем результатами работы проце-

дур обычно являются какие-либо изменения текстов и рисунков на

лицевой стороне листа. Общее количество текстовой информации на

обеих сторонах листа не должно превышать 4061 символ, это коли-

чество может распределяться произвольно - например, одна сторона

листа может содержать 4061 символ, если другая сторона не содер-

жит текста. При работе с листом в системе LogoWriter в любой мо-

мент Вам доступна какая-то одна сторона листа (обычно после заг-

рузки листа с диска лист повернут к Вам лицевой стороной). С по-

мощью клавиши "переверни" (см.I.2) Вы можете перевернуть лист,

то есть сделать доступной другую сторону листа.

Во время работы с листом экран дисплея выглядит следующим

образом: в верхней строке написано имя листа (или символы "???"

- если лист новый и ему не присвоено имя) и слово "Изнанка", ес-

ли Вы работаете с изнаночной стороной листа. Экран разделен го-

ризонтальной чертой на две неравные части. В верхней области эк-

рана находится часть текстовой информации (19 строк), записанной

на выбраной стороне листа. (Нижняя область экрана используется

при работе с полем команд, описанным ниже). С помощью специаль-

ных клавиш Вы можете просматривать в верхней части экрана имею-

щийся

- 3 -

на листе текст. Если Вы работаете с лицевой стороной, то в верх-

ней части экрана кроме текста изображается вся графическая ин-

формация, нарисованная на листе.

- 4 -

Вы можете непосредственно изменять информацию на доступной

стороне листа - например, Вы можете записать какой-нибудь текст.

Кроме того, Вы можете выполнять элементарные команды языка

LogoWriter (команды-примитивы, они описаны в части II) или вы-

полнять некоторые процедуры . Выполняемые примитивы и вызовы

процедур (иными словами - инструкции) пишутся в нижней части эк-

рана, под чертой. Написанные Вами инструкции хранятся в специ-

альном буфере - поле команд, часть содержимого поля команд видна

в нижней части экрана. С помощью специальных клавиш Вы можете

просматривать, редактировать и (повторно) выполнять инструкции,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) - Обычно текст выполняемой процедуры должен быть записан

на изнаночной стороне того листа, с которым Вы в данный момент

работаете, однако в системе LogoWriter имеются средства, позво-

ляющие выполнять процедуры, написанные на других листах.

- 5 -

записанные в поле команд. Поле команд не является частью листа,

поэтому его содержимое не записывается в файл, в котором хранит-

ся лист, и не изменяется при смене листов.

Так как на диске может храниться несколько листов, то сис-

тема обычно в начале работы предлагает Вам выбрать нужный лист в

оглавлении. Оглавление представляет собой меню системы, позволя-

ющее выбрать нужный Вам режим работы. Одним из таких режимов ра-

боты (кроме работы с обычным листом) является просмотр и редак-

тирование набора форм черепашки. В системе LogoWriter Вы можете

вместо традиционного изображения черепашки языка Logo использо-

вать одно из 90 изображений, и система предоставляет Вам возмож-

ность самому нарисовать нужные изображения. Набор форм представ-

ляет собой закодированную специальным образом информацию, однако

разработчики системы стремились к тому, чтобы работа с набором

форм была похожа на работу с обычным листом. По этой причине на-

бор форм черепашки называется в системе LogoWriter листом форм,

хотя он и не является листом в описанном выше смысле и работа с

ним отличается от работы с обычным листом.

Если Вы хотите сохранить лист на диске, то он должен иметь

имя, в качестве которого может быть использована любая последо-

вательность, содержащая не более 8 символов, допускаемая систе-

мой MS DOS в качестве имени файла. Листы с именами "startup" и

"help" играют особую роль. Если на диске имеется лист с именем

"startup", то система LogoWriter после загрузки сразу переходит

к работе с этим листом, а не к работе с оглавлением. Если на

диске имеется лист с именем "help", то содержимое этого листа

доступно для чтения (но не для изменения) при работе с любым

другим листом (подробнее о листах "startup" и "help" см. прило-

жение 2).

Кроме файлов, предназначенных для хранения обычных листов и

листа форм, система LogoWriter позволяет использовать обычные

текстовые файлы системы MS DOS и специальные графические файлы,

предназначенные для хранения графики. Вы можете перенести текст

с любой стороны листа в текстовый файл или записать текст из

текстового файла на нужной Вам стороне листа. Вы также можете

сохранить графику, имеющуюся на лицевой стороне произвольного

листа, в графическом файле, а также заменить графику, имеющуюся

на лицевой стороне любого листа, на картинку, хранящуюся в гра-

фическом файле.

2. Используемые файлы и специальные клавиши.

В системе LogoWriter используются следующие файлы:

LOGOWR.COM - загрузочный модуль, единственный файл, необхо-

димый для работы системы. Этот файл загружается с внешнего носи-

теля (диска) в начале сеанса. В процессе работы системы

LogoWriter данный загрузочный модуль находится в оперативной па-

мяти. Таким образом файл LOGOWR.COM нужен лишь в начале сеанса,

для загрузки в оперативную память, в процессе сеанса внешний но-

ситель, содержащий этот модуль может быть снят (см.I.3).

SHAPES.SHP - файл, содержащий лист форм, которые можно ис-

пользовать в качестве формы "черепашки". Если этот файл отсутст-

- 6 -

вует, то в качестве набора форм используется некоторый начальный

набор форм, создаваемый программой LOGOWR.COM. Файл SHAPES.SHP

обновляется (или создается) при завершении работы с листом форм

(см.I.4.2). Файл SHAPES.SHP может быть удален с помощью команды

"сотрифайл".

Файлы, предназначенные для хранения листов имеют имена ви-

да: имя\_листа.LWR. Такие файлы корректируются (и создаются) или

с помощью команды "помнилист" или при завершении работы с листом

(см.I.3), для создания файла листу должно быть присвоено имя ко-

мандой "ил". Лист загружается в оперативную память компьютера

или при работе с оглавлением (I.4.1) или командой "возьмилист".

Файл может быть удален при работе с оглавлением (I.4.1) или с

помощью команд "сотрилист" и "сотрифайл". Команда "защита" поз-

воляет защитить файл от изменений, команда "беззащиты" снимает

защиту.

Текстовые файлы. Они могут иметь произвольные имена, допус-

каемые системой MS DOS. Их структура совпадает с обычной струк-

турой текстовых файлов системы MS DOS, кодировка символов -

"альтернативная" (приложение 6). Файлы создаются командой "пом-

нитекст", содержимое файлов становится доступным с помощью ко-

манды "возьмитекст". Файлы могут быть удалены командой "сотри-

файл".

Графические файлы могут иметь произвольные имена, допускае-

мые системой MS DOS. Структура файла имеет специальный вид. Фай-

лы создаются командой "помнирис", содержимое файлов становится

доступно с помощью команды "возьмирис". Файлы могут быть удалены

командой "сотрифайл".

При работе с системой LogoWriter используются различные

специальные клавиши. В описании системы LogoWriter для обозначе-

ния таких клавиш обычно используются не символы, изображенные на

клавише, а специальные названия. Так, в дальнейшем мы будем

обычно говорить не о клавише F4, а о клавише "верни". Ниже при-

водится список специальных клавиш и их названия.

F1 - выделяй

F2 - вырежи

F3 - копируй

F4 - верни

F5 - шрифт

F6 - удаление строки

F8 - надпись

F9 - перемещение

F10 - помощь

Enter - ввод

Esc - выход

PgUp - следующий экран

PgDn - предыдущий экран

Home - начало листа

End - конец листа

В следующих 6 строчках "+" означает комбинацию двух клавиш:

при нажатой клавише "Ctrl" нажимается соответствующая вторая

клавиша.

- 7 -

Ctrl + U - вверх

Ctrl + D - вниз

Ctrl + F - переверни

Ctrl + Break - останов

Ctrl + <- - начало строки

Ctrl + -> - конец строки

3. Сеанс работы с системой LogoWriter.

Для вызова системы следует выполнить команду

>d:path\LOGOWR

здесь d - накопитель, path - путь доступа, описывающий файл

LOGOWR.COM . На экране дисплея появляется фирменный знак систе-

мы. К этому моменту модуль LOGOWR.COM загружен в оперативную па-

мять, поэтому можно вынуть дискету, содержащую этот файл и заме-

нить ее на нужную Вам рабочую дискету. Далее следует нажать кла-

вишу "ввод". Если существует лист с именем "startup", то система

загружает этот лист и Вы переходите к работе с этим листом (см.

приложение 2). В противном случае Вы переходите к работе с ог-

лавлением (меню) системы. Типичное оглавление приведено на рис.

3.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1)- Коллектив "Школа-1" распространяет дискеты, содержащие кроме

файла LOGOWR.COM ряд дополнительных файлов, необходимых для на-

чальной загрузки MS DOS и автоматического выполнения описанной

команды. При использовании такой дискеты для вызова системы дос-

таточно вставить дискету в накопитель и включить компьютер.

- 8 -

Экран при работе с оглавлением выглядит следующим образом:

в верхней строке написано название текущего каталога и слово

"оглавление" (навзвание каталога отсутствует, если Вы работаете

с корневым каталогом). Далее следует поясняющий текст и два вы-

деленых режима - "Новый лист" и "Формы". Ниже идет список лис-

тов, имеющихся в каталоге. После списка листов идет список под-

каталогов текущего каталога, перед названием каждого подкаталога

стоит символ "\". Строка, в которой находится курсор, выделяется

инверсным шрифтом (темные буквы на светлом фоне). При работе с

оглавлением Вы можете удалить лист (если он не был защищен ко-

мандой "защита"), перейти в подкаталог, выбрать или лист форм,

или уже имеющийся обычный лист, или новый лист (см.I.4.1).

При выборе имени подкаталога система высвечивает его оглав-

ление.

Если выбран режим "Формы", то система переходит в режим ре-

дактирования листа форм (см.I.4.2). Работа в этом режиме завер-

шается нажатием клавиши "выход", измененный набор форм записыва-

ется на диск и система возвращается в режим работы с оглавлени-

ем.

Если выбран режим "Новый лист" то система переходит в режим

создания нового листа (см. I.4.3). Если выбрано имя уже имеюще-

гося листа, то система берет содержимое листа, записанное в со-

ответствующем файле, и переходит в режим работы с этим листом

(см.I.4.3).

При загрузке листа с диска система проводит синтаксический

анализ текстов процедур, записанных на изнаночной стороне листа.

Если обнаружена синтаксическая ошибка, то система сообщает об

этом. Процедура, в которой обнаружена ошибка, и все процедуры,

написанные ниже ошибочной, при работе с листом игнорируются.

Работа с текущим листом завершается:

- при выполнении команды "оглавление" или при нажатии кла-

виши "выход": содержимое текущего листа записывается на диск,

система переходит в оглавление;

- при выполнении команды "возьмилист": содержимое текущего

листа записывается на диск, система переходит к работе с листом,

указанным в команде;

- при выполнении команды "уходи": информация на диске не

изменяется, система переходит в оглавление;

- при выполнении команды "дос": содержимое текущего листа

записывается на диск, сеанс работы с системой LogoWriter закан-

чивается.

4. Различные режимы работы.

4.1. Оглавление.

При работе с оглавлением Вы можете использовать следующие

клавиши:

стрелка вверх и стрелка вниз - перемещают курсор по меню.

"ввод" - выход из оглавления и переход к выбраному режиму

работы.

- 9 -

"выход" - на экране дисплея высвечивается оглавление. Эта

клавиша может быть использована в следующей ситуа-

ции: работая с оглавлением, Вы сменили диск. В

этом случае после нажатия клавиши "выход" появит-

ся оглавление, соответствующее новому диску.

"начало листа" ("конец листа") - переход в начало (конец)

оглавления.

"следующий экран" ("предыдущий экран") - высвечивание сле-

дующей (предыдущей) порции оглавления (если такая

существует).

"удаление строки" - с диска удаляется лист, на который ука-

зывает курсор. Удаления не происходит, если лист

защищен командой "защита". Будьте внимательны - в

системе LogoWriter нет средств, позволяющих восс-

тановить удаленный лист.

4.2. Лист форм.

Лист форм предназначен для хранения и редактирования набора

форм, которые можно использовать вместо стандартной формы чере-

пашки (форма черепашки меняется с помощью команды "новформа").

Для того, чтобы перейти к работе с листом форм, следует или выб-

рать режим "формы" в оглавлении или выполнить команду "формы"

при работе с обычным листом.

Лист форм имеет две стороны - лицевую и изнаночную. Лицевая

сторона предназначена для выбора нужной формы, изнаночная - для

редактирования форм. Переход с лицевой стороны на изнаночную и

обратно происходит с помощью клавиши "переверни".

В начале работы лист повернут лицевой стороной, выделена

форма номер 1. На лицевой стороне находится 90 пронумерованных

форм в натуральную величину, каждая форма имеет размер 16х16 то-

чек. Выделенная форма окрашена красным цветом. На одном экране

помещается изображение 30-ти форм. В верхней строке экрана ука-

зано имя листа форм - "shapes".

При работе с лицевой стороной можно использовать следующие

клавиши:

стрелки - перемещение по экрану для выделения нужной формы.

"выход" - завершение редактирования листа форм.

"следующий экран", ("предыдущий экран") - высвечивается

следующий (предыдущий) набор из 30-ти форм.

"переверни" - переход на изнаночную сторону листа.

На изнаночной стороне листа находится изображение выделен-

ной формы в натуральную величину, а также увеличенное изображе-

ние выделенной формы. В верхней строке экрана указано имя листа

- "shapes" и слово "Изнанка". Изменение формы происходит при ра-

боте с увеличенным изображением. Изменение изображения черепашки

в натуральную величину происходит автоматически.

Используются следующие клавиши:

"выход" - завершение редактирования листа форм.

"переверни" - переход на лицевую сторону листа.

стрелки - перемещение курсора по увеличенному изображению

- 10 -

формы.

пробел - если клетка, в которой находится курсор, не зак-

рашена, то она закрашивается, в противном слу-

чае - очищается.

"следующий экран" ("предыдущий экран") - переход к следую-

щей (предыдущей) форме.

"вырежи" - текущая форма удалаяется с листа форм и запоми-

нается в буфере.

"копируй" - текущая форма копируется в буфер.

"верни" - форма, находящаяся в буфере (после удаления или

копирования) заменяет текущую.

"останов" - редактирование формы прекращается, восстанавли-

вается первоначальная форма.

4.3. Обычный лист.

Работая с обычным листом, Вы имеете возможность выбрать

один из трех режимов: редактирование текстов, перемещение чере-

пашки и режим "надпись".

В первом режиме Вы можете:

- записывать, редактировать и выполнять инструкции, находя-

щиеся в поле команд.

- записывать и редактировать тексты процедур на изнаночной

стороне листа.

- записывать и редактировать произвольные тексты на лицевой

стороне листа.

В режиме перемещения черепашки Вы можете передвигать чере-

пашку по лицевой стороне листа.

В режиме "надпись" Вы можете создавать на лицевой стороне

листа произвольные надписи. Надпись является частью графического

изображения (а не частью текста). Поэтому, например, команда

"ст" (сотри текст) не удаляет надписи, не может изменять надписи

и текстовый редактор.

Обычно в начале работы с листом Вам доступна лицевая сторо-

на, включен режим редактирования текстов в поле команд (в нижней

части экрана мигает текстовый курсор). Клавиша "вверх" переводит

систему в режим редактирования текстов на лицевой стороне листа

(мигает текстовый курсор в верхней части экрана), клавиша "вниз"

возвращает Вас к работе с полем команд. Клавиша "переверни" поз-

воляет перейти от лицевой стороны листа к изнаночной и обратно.

Клавиши "вверх" и "вниз" играют на изнаночной стороне листа ту

же роль, что и на лицевой. Работая с листом, Вы можете, нажав

клавишу "надпись", перейти в ражим "надпись", нажав клавишу "пе-

ремещение" перейти в режим перемещения черепашки. И в том и в

другом случае нажатие клавиши "выход" возвращает Вас в режим ре-

дактирования текстов.

При работе с редактором текстов можно использовать следую-

щие клавиши:

стрелки - перемещение курсора по тексту.

"ввод" - действие клавиши ввод зависит от того, где Вы

ботаете. Если Вы редактируете текст в поле команд,

то нажатие клавиши "ввод" приводит к выполнению ин-

- 11 -

струкции, записанной в той строке, в которой нахо-

дится курсор. Если несколько строк в поле команд

вводились без нажатия клавиши "ввод", то все эти

строки образуют одну инструкцию. Если Вы редактиру-

ете текст на листе, то нажатие клавиши "ввод" при-

водит к записи в тексте в позиции курсора специаль-

ного символа - метки конца строки. Этот символ не

отображается на экране, символы записанные после

метки конца строки пишутся на экране с новой стро-

ки.

"выделяй" - включение режима выделения текста. Перемещая

курсор с помощью стрелок, Вы можете выделить фраг-

мент текста. Этот фрагмент обозначается на экране

дисплея инверсным шрифтом - темные буквы на светлом

фоне.

"вырежи" - выделенный фрагмент помещается в буфер и удаля-

ется из текста.

"копируй" - выделенный фрагмент копируется в буфер.

"верни" - фрагмент текста, находящийся в буфере, вставляет-

ся в текст в том месте, где расположен курсор.

"выход" - если включен режим выделения, то нажатие клавиши

"выход" отменяет его. Если режим выделения не вклю-

чен, то нажатие клавиши "выход" приводит к записи

листа на диск, система переходит к работе с оглав-

лением.

"удаление строки" - удаляется текст от позиции курсора до

конца строки.

"следующий экран" ("предыдущий экран") - на экран выводится

следующая (предыдущая) порция текста.

"начало листа" ("конец листа") - переход в начало (конец)

текста.

"начало строки" ("конец строки") - курсор переводится в на-

чало (конец) строки.

"шрифт" - переключение клавиатуры с русского алфавита на

английский и обратно.

В режиме перемещения черепашки Вы можете с помощью стрелок

передвигать черепашку по экрану. При включении данного режима

изображение черепашки начинает мигать.

В режиме "надпись" Вы можете писать надписи на экране. Над-

писи (т.е. тексты, сделанные в режиме "надпись") являются частью

графической информации, они не могут быть, в частности, изменены

с помощью текстового редактора. При включении режима "надпись"

на экране появляется изображение курсора надписей, Вы можете пе-

ремещать этот курсор с помощью стрелок.

При работе с листом Вы можете использовать две дополнитель-

ные клавиши:

"помощь" - на экране появляется содержимое листа "help"

(см. приложение 2). Вы можете просматривать на эк-

ране этот лист, но не можете его изменять или пере-

ворачивать. Поле команд в это время также недоступ-

но. Нажатие клавиши "выход" возвращает Вас на

- 12 -

ходный лист.

"останов" - нажатие клавиши "останов" прерывает выполнение

процедуры или команды.

Вы можете временно прервать работу с текущим листом и с по-

мощью команды "формы" перейти к редактированию листа форм. По

окончании работы с листом форм (нажатие клавиши "выход") система

возвращается к работе с исходным листом.

Завершение работы с текущим листом описано в разделе I.3.

Работая с листом, Вы могли с помощью команд или процедур

определить значения тех или иных переменных и с помощью команды

"контроль" запрограммировать некоторые клавиши. Информация об

этом не записывается на диск и, таким образом, не будет автома-

тически восстановлена при следующей загрузке листа. Но следует

помнить, что значения переменных и программирование клавиш не

уничтожаются автоматически при смене листов в течение одного се-

анса.

-13-

Часть II. Описание языка LogoWriter.

В данной части приводится неформальное описание языка

LogoWriter. Дается алфавитный список примитивов

(зарезервированных слов) языка с комментариями и примерами

использования.

1. Структурв языка.

1.1. Типы данных.

В языке LogoWriter используются три основных типа данных -

числа, слова и списки. Числа, допустимые в языке, содержатся в

интервале от -(10000000000000-0.0001) до 10000000000000-0.0001 и

имеют не более 4 десятичных знаков. Словом в языке LogoWriter

называется произвольная последовательность символов, содержащая

не более 32 000 символов. Списком называется произвольная

последовательность элементов списка, элементом списка может быть

или слово или список.

Числовая константа иммеет один из следующих видов:

<целая часть>.<дробная часть>

<целая часть>,<дробная часть>

-<целая часть>.<дробная часть>

-<целая часть>,<дробная часть>

здесь целая часть - последовательность из не более чем 12

цифр, дробная часть - последовательность цифр. При записи целого

числа дробная часть и десятичный знак могут отсутствовать.

Значением числовой константы является некоторое слово. При

вычислении значения числовой константы в дробной части

отбрасываются все цифры, начиная с пятой, отбрасываются все

незначащие нули, если имеется дробная часть, то десятичный знак

записывается в виде точки или запятой в зависимости от

установленного режима (см. примитивы "точка" и "запятая").

Пример числовых констант (мы предполагаем, что установлен

режим "точка"):

Запись Значение

-01,500 -1.5

12.33339 12.3333

Константа типа слово имеет следующий вид:

"<последовательность символов>

здесь <последовательность символов> должна содержать не более

32000 символов и не должна содержать символов пробел, [, ], (,

). Значением такой константы является слово, образованное

указанной последовательностью.

Пример констант типа слово:

Запись Значение

"abc abc

"abc" abc"

"-001.40009 -001.40009

Отметим, что не всякое слово может быть задано с помощью

константы (например, так нельзя задать слова, содержащие символ

пробел).

Костанты типа список записываются в виде:

[<элемент1>...<элементN>]

где <элементi> - или константа типа список или слово, не

- 14 -

содержащее символов пробел, [, ], (, ). Элементы списка

разделяются пробелами.

Пример константы типа список:

["abc [a b c] +001.100]

Значением этой константы будет список, первым элементом которого

будет четырехсимвольное слово "abc, вторым - список из трех

элементов, а третьим - восьмисимвольное слово +001.100.

В языке LogoWriter используется неявное преобразование

типов, а именно преобразование данных типа слово в числовой тип

и обратно. При преобразовании числового данного в данное типа

слово берется значение числового данного. При преобразовании

данного типа слова в данное типа число, преобразование осуществ-

ляется в точности также, как и при определении значения числово-

вой константы (см. выше).

Приведем некоторые примеры преобразования типов данных. В

примерах используется числовой датчик "+", дающий сумму двух чи-

словых аргументов и датчик "слово", соединяющий два аргумента

типа слово в одно слово.

Выражение Значение

"01.200 + 7,5 8.7

слово "01.200 -17.000 01.200-17

Если при выполнении операции сравнения один аргумент

является словом, а другой - числом, то производится попытка

преобразовать слово в число и далее сравниваются два числа.

Например:

Выражение Значение

01.200 = 1,2 истинна

"01.200 = 1,2 истинна

"01.200 = "1,2 ложь

1.2. Имена

В языке LogoWriter должны быть поименованны следующие

объекты: переменные, процедуры, листы и файлы на внеших

носителях. На имена переменных не накладывается никаких

ограничений - любое слово может быть именем переменной. Примеры

имен переменных:

а

аб

-1.75

\*\*[]

Хотя в языке можно использовать произвольные символы в

именах переменных, мы не советуем без необходимости использовать

в именах те сиволы, которые не могут быть использованы в

константах типа слово, поскольку доступ к значениям переменных с

такими "экзотическими" именами достаточно сложен.

Именем процедуры может быть любое слово, составленное из

букв, цифр, символов -, \_ , и отличное от записи числа. Пример

имен процедур:

квадрат

-1.75а

а\_б

Посколько имя листа используется в качестве имени файла на

- 15 -

диске, длина имени листа определяется операционной системой. В

нашем случае имя листа должно содержать не более 8 символов и

может состоять из букв латинского алфавита и цифр. Расширение

имени листа (символы LWR, см. I.2) в языке LogoWriter не

указываются.

Все прочие (отличные от листов) файлы задаются с помощью

имен с расширениями, список тех символов, которые можно

использовать в имени файла и в расширении имени определяются

операционной системой.

1.3. Датчики и команды.

Функции в языке LogoWriter называются датчиками. Выбор

названия объясняется тем, что в языке широко распостранены

0-местные функции, значение которых определяется текущим

состоянием вычислительной обстановки (например, датчик "цвет",

значением которого является номер текущего цвета черепашки, или

датчик "файлы", значение которого - список имен файлов в текущем

каталоге). Датчики могут быть встроенными, такие датчики

называются примитивами. Список всех примитивов приведен далее.

Кроме того Вы можете создать собственные датчики, так называемые

процедуры-датчики (см. описание процедур в разделе 1.4). Вызов

датчика имеет один из следующих видов:

<арг1> <имя датчика> <арг2>

(<имя датчика> <арг1> ... <аргN>)

здесь <арг1> ... <аргN> - аргументы датчика.

Первая запись используется для арифметических датчиков "+",

"-", "\*", "/" и логических датчиков "=", "<", ">". При вызове

всех остальных датчиков используется запись второго вида. Если

имя датчика однозначно определяет число аргументов (а это

выполнено для всех процедур-датчиков и для большинства

датчиков-примитивов), то при вызове датчика внешние круглые

скобки можно опустить. Для остальных датчиков указано, при каком

числе аргументов можно опустить внешние скобки.

Аргументы датчиков могут иметь один из следующих видов:

<константа>

:<имя переменной>

<вызов датчика>

Описание констант приведено в разделе 1.1, <имя переменной>

- это любое имя переменной, которое может быть задано с помощью

константы типа слово.

Типы аргументо в изначений датчиков-примитивов указанны при

описании примитивов, типы аргументов и значений

процедур-датчиков определяются операциями, применяемыми в теле

процедуры к значениям аргументов и типом того данного, которое

вырабатывается в качестве значения прорцедуры.

Приоритет датчиков:

1) \*,/

2) +,-

3) =,>,<

4) все остальные

Пример вызова датчика:

(слово "а :х элемент 2 "абв)

Здесь приведен вызов датчика "слово" с тремя аргументами. В

данном случае внешние скобки необходимы, так как без внешних

скобок можно вызывать датчик "слово" только с двумя аргументами.

- 16 -

Значение первого аргумента задано константой, значение второго

аргумента есть значение переменной с именем "х", значение

третьего аргумента вычисляется с помощью вызова датчика

"элемент", в этом вызове оба аргумента - константы.

При вызове датчика порядок вычисления определяется

приоритетом, таким образом вызов датчика

элемент 3 "абв = "в

даст значение "ж", поскольку сначала будет вычислено значение

датчика "=" с аргументами "абв" и "в" (значением этого датчика

будет слово "ложь"), далее будет вычислено значение датчика

"элемент" с аргументами 3 и "ложь". Значение этого датчика -

символ "ж". Если Вы хотите изменить порядок вычислений, то

следует пользоваться круглыми скобками. Таким образом вызов

датчика

(элемент 3 "абв) = "в

даст значение "истинна".

Операторы языка LogoWriter называются командами. Исходные

(встроенные) команды называются командами-примитивами (список

примитивов приведен далее). Кроме того, Вы можете создавать

процедуры-команды (обычно их называют просто процедурами, по

поводу процедур см. 1.4). Вызов команды имеет вид

(<имя команды> <арг1> ... <аргN>)

Аргументы команд имеют тот же вид, что и аргументы

датчиков. Соглашение о внешних скобках такое же, что и в случае

датчиков.

1.4. Процедуры и переменные.

Текст процедур записывается на изнаночной стороне листа.

Текст процедуры состоит из заголовка и тела процедуры. Заголовок

имеет вид:

это <имя процедуры> <список формальных параметров>

Имя процедуры должно быть уникальным, то есть оно должно

отличаться от имен остальных процедур и от имен примитивов.

Список формальных параметров представляет из себя

последовательность разделенных пробелами выражений вида

:<имя переменной>

Завершает заголовок метка конца строки.

Тело процедуры представляет собой произвольную

последовательность вызовов команд, завершаемую специальным

словом "конец". На команды, используемые в теле прорцедуры, не

накладывается никаких ограничений, в частности в теле процедуры

можно вызывать другие процедуры и саму себя (организовывать

рекурсивные вызовы). Имеется две команды - "стоп" и "выход",

которые можно использовать лишь внутри процедур. Первая из них

не имеет аргументов и просто прекращает выполнение процедуры.

Вторая команда - команда "выход" применяется при создании

процедур-датчиков. Она имеет один аргумент произвольного типа.

Эта команда прекращает выполнение процедуры-датчика, значением

процедуры становится значение аргумента команды "выход".

Приведем примеры процедур. Процедура-команда:

это фигура :х :у

повтори :х [вперед :у направо 360 / :х]

конец

Вызов команды

фигура 5 60

приведет к тому, что на лицевой стороне листа будет нарисован

пятиугольник с длинной стороны 60.

- 17 -

Процедура-датчик:

это куб :х

выход :х \* :х \* :х

конец

Вызов команды

покажи куб 3

приведет к тому, что в командной области будет напечатано число

27.

В языке LogoWriter параметры процедур передаются по

значению, то есть при вызове процедуры создаются новые

переменные, соответствующие формальным параметрам процедуры,

начальное значение этих переменных совпадает со значением

аргументов в вызове процедуры, имена этих переменных совпадают с

именами, указанными в списке формальных параметров. По

завершению работы процедуры все эти новые переменные становятся

недоступны. Таким образом формальные параметры процедуры

соответствуют локальным переменным. Все остальные переменные

являются глобальными переменными. Глобальные переменные

создаются с помощью команд "пусть" и "назови". Если имя

переменной в списке формальных параметров какой-либо процедуры

совпадает с именем глобальной процедуры, то эта глобальная

переменная становится недоступна на время работы этой процедуры.

Во всех остальных случаях значение глобальной переменной может

быть прочитанно и изменено в любой момент, после создания

переменной. Более того, имена и значения глобальных переменных

относятся к общей вычислительной обстановке, то есть они не

изменяются при смене листов во время одного сеанса работы. К

общей вычислительной обстановке относятся также программирование

клавиш, текущий шрифт (кириллица или латинский), набор внешних

процедур и некоторые иные параметры.

Такие параметры, как цвет, номер формы и положение

черепашки образуют вычислительную обстановку листа, она

сохраняется на диске при смене листа и восстанавливается при

повторном вызове листа.

2. Примитивы языка LogoWriter.

Данный раздел содержит алфавитный список примитивов с ком-

ментариями и примерами.

Примитивы описываются по определенной схеме.

Приведем пример описания примитива "пиши":

-----------------------------------------------------------------

(1) пиши <слово/список> (5) 1 аргумент

(2) п <слово/список> (6) команда

(3) print

(4) Замечание: команда "пиши" допускает несколько аргументов; в

том случае когда аргументов два и более, команда вместе со

всеми аргументами заключается в круглые скобки:

(пиши <слово/список1> <слово/список2>...)

(7) Действие команды "пиши" заключает в себе две операции (кото-

рые можно выполнить и вручную): во-первых, на текущей стороне

листа печатается <слово/список> (при этом внешние скобки спи-

ска не печатаются) и, во-вторых, в представлении текста в

памяти машины (но не на экране!) появляется метка конца

строки, что равнозначно нажатию клавиши "ввод".

См. вставь, покажи, пишикоманды, надпись.

- 15 -

(8) Примеры:

ст (9) \*сотри текст\*

пиши "хорошо

п [хорошо живет]

(п [хорошо живет] [на свете] "Винни "Пух)

----------------------------------------------------------------

(1) синтаксис примитива со стандартным количеством аргуме-

нтов (в угловых скобках указываются допустимые типы аргу-

ментов).

(2) сокращенный русский вариант примитива

(3) английские варианты

(4) синтаксис примитива с нестандартным количеством аргу-

ментов (если таковое допускается примитивом)

(5) стандартное для данного примитива количество аргумен-

тов

(6) тип примитива

(7) комментарий

(8) примеры употребления

(9) звездочками отмечаются пояснения к отдельным инстру-

кциям или расшифровка сокращенных инструкций.

Типы примитивов LogoWriter'а:

1. Команда - это примитив, предписывающий LogoWriter'у выполне-

ние каких-либо действий.

2. Датчик - это примитив, передающий какую-либо информацию (на-

пример, слово или список) другому примитиву.

3. Логический датчик - это датчик, возвращающий значение "исти-

на" или "ложь".

4. Специальное слово ( конец, это).

Типы аргументов датчиков и команд (напомним, что некоторые

типы, такие как слово, список, число, имя переменной, имя проце-

дуры, имя листа и имя файла описанны в разделах 1.1 и 1.2):

буква буква английского алфавита.

время число от 0 до 255 (время в Logo-

Writer'е измеряется в тиках, 1 тик

приблизительно равняется 1/20 сек)

имя имя процедуры или имя переменной.

имя листа см. раздел 1.2.

имя файла см. раздел 1.2.

истина/ложь примитив или процедура, возвраща-

ющая значение "истина" или "ложь".

номер кнопки число 0, 1, 2 или 3.

символ буква английского или русского

- 16 -

алфавита, число или знак препина-

ния.

слово см. раздел 1.1.

слово/список слово или список.

список см. раздел 1.1.

список инструкций список, представляющий из себя пос-

ледовательность вызовов команд или

вызов датчика.

список черепашек список, элементами которого явля-

ются цифры 0, 1, 2 , 3 (никакая

цифра не может входить в список

дважды).

частота число от 37 до 9999.9999.

черепашка число 0, 1, 2 или 3.

число см. раздел 1.1.

[х у] список из двух чисел, задающих ко-

ординаты черепашки.

Арифметические операции.

-----------------------------------------------------------------

<число1> + <число2> 2 аргумента

датчик

Возвращает сумму чисел <число1> и <число2>.

Пример:

покажи 5 + 3

-----------------------------------------------------------------

<число1> - <число2> 2 аргумента

датчик

Возвращает разность чисел <число1> и <число2>.

Пример:

покажи 8 - 6

-----------------------------------------------------------------

<число1> \* <число2> 2 аргумента

датчик

Возвращает произведение чисел <число1> и <число2>.

Пример:

покажи 8 \* 6

- 17 -

-----------------------------------------------------------------

<число1> / <число2> 2 аргумента

датчик

Возвращает частное от деления <числа1> на <число2>.

Пример:

покажи 678 / 45

-----------------------------------------------------------------

<слово/список1> = <слово/список2> 2 аргумента

логический датчик

Возвращает значение "истина", если <слово/список1> и <слово/спи-

сок2> равны, и "ложь" - в противном случае. Проверка равенства

слов не различает прописных и строчных букв английского алфави-

та, однако, в случае русского алфавита, прописные и строчные

буквы различаются. Слово не равно списку, содержащему ровно это

слово.

Замечание: для компьютера русская и латинская буквы, имеющие

одинаковые начертания, - суть разные буквы.

См. равны?, одинаковы?

Примеры:

покажи 8 = 7

В поле команд Вы увидите: ложь

покажи "word = "Word

В поле команд Вы увидите: истина

покажи "муха = [муха]

В поле команд Вы увидите: ложь

------------------------------- ----------------------------------

<число1> > <число2> 2 аргумента

логический датчик

Возвращает значение "истина", если <число1> больше <числа2>, и

"ложь" - в противном случае.

Примеры:

покажи 6 > 7

если 5 > 4 [пиши [5 > 4]]

-----------------------------------------------------------------

<число1> < <число2> 2 аргумента

логический датчик

Возвращает значение "истина", если <число1> меньше <числа2>, и

"ложь" - в противном случае.

Примеры:

покажи 9 < 8

В поле команд Вы увидите: ложь

если 5 < 7 [пиши [решение верно]]

- 18 -

----------------------------------------------------------------

беззащиты без аргументов

unlock команда

Снимает с текущего листа защиту, установленную ранее командой

"защита".

См. защита

Пример:

новлист

ил "test \*имя листа\*

защита

беззащиты

-------------------------------------------------------------

буфер без аргументов

clipboard датчик

Возвращает содержимое текстового буфера в виде строки. Если

буфер пуст, возвращает пустую строку.

Пример:

ст \*сотри текст\*

пиши [чудесный вечер]

внт \*в начало текста\*

выделяй

вкст \*в конец строки\*

вырежи

покажи буфер

В поле команд Вы увидите: чудесный вечер

--------------------------------------------------------------

быстро без аргументов

fastturtle команда

Устанавливает быстрый режим движения черепашек. В исходном со-

стоянии (т.е. после загрузки системы) устанавливается быстрый

режим движения черепашек.

См. медленно.

Пример:

иг \*исходная графика\*

быстро

повтори 200[пр 3 вп 1] \*направо 3, вперед 1\*

медленно

повтори 200[пр 3 вп 1]

- 19 -

-------------------------------------------------------------

верни без аргументов

paste команда

Вставляет текст, находящийся в буфере, от текущего положения

курсора. Того же эффекта можно достичь нажатием клавиши "вер-

ни".

Пример: \*сотри текст\*

ст

пиши [Винни Пух]

пиши [хорошо живет на свете]

внт \*в начало текста\*

выделяй

повтори 9[квп] \*курсор вперед\*

вырежи

вкт \*в конец текста\*

верни

--------------------------------------------------------------

вксп <слово/список> <список> 2 аргумента

lput датчик

Сокращение от "В Конец СПиска". Возвращает новый список, по-

лученный при добавлении первого аргумента(слова или списка)

в конец второго аргумента(списка).

См. внсп (в начало списка).

Примеры:

ст \*сотри текст\*

покажи вксп "чудесный! [какой день]

В поле команд Вы увидите: [какой день чудесный!]

покажи вксп [день чудесный!] [какой сегодня]

В поле команд Вы увидите: [какой сегодня [день чудесный!]]

---------------------------------------------------------------

вкст без аргументов

eol команда

Сокращение от "В Конец СТроки". Перемещает курсор на листе в

конец строки. Того же эффекта можно достичь нажатием клавиши

"конец строки".

См. внст (в начало строки).

Пример:

ст \*сотри текст\*

пиши [хорошо живет на свете]

пиши [Винни Пух]

внт \*в начало текста\*

вкст

- 20 -

----------------------------------------------------------------

вкт без аргументов

bottom команда

Сокращение от "В Конец Текста". Перемещает курсор на листе в ко-

нец текста. Того же эффекта можно достичь нажатием клавиши "ко-

нец листа".

См. внт (в начало текста).

Пример:

ст \*сотри текст\*

пиши [хорошо живет на свете]

пиши [Винни Пух]

внт \*в начало текста\*

выделяй

вкт

----------------------------------------------------------------

внсп <слово/список> <список> 2 аргумента

fput датчик

Сокращение от "В Начало СПиска". Возвращает новый список, полу-

ченный при добавлении первого аргумента (слова или списка) в на-

чало второго аргумента (списка).

См. вксп (в конец списка).

Пример:

ст \*сотри текст\*

покажи внсп "какой [чудесный день!]

В поле команд Вы увидите: [какой чудесный день!]

покажи внсп [какой сегодня] [чудесный день!]

В поле команд Вы увидите: [[какой сегодня] чудесный день!]

----------------------------------------------------------------

внст без аргументов

sol команда

Сокращение от "В Начало СТроки". Перемещает курсор на листе в

начало строки. Того же эффекта можно достичь нажатием клавиши

"начало строки".

См. вкст (в конец строки).

Пример:

вставь [хорошо живет на свете]

внст

----------------------------------------------------------------

внт без аргументов

top команда

Сокращение от "В Начало Текста". Перемещает курсор на листе в

начало текста. Того же эффекта можно достичь нажатием клавиши

"начало листа".

См. вкт (в конец текста).

Пример:

ст \*сотри текст\*

пиши [оттого поет он эти]

вставь [песни вслух]

выделяй

внт

- 21 -

----------------------------------------------------------------

возьмилист <имя листа> 1 аргумент

вл <имя листа> команда

getpage

Текущий лист сохраняется на диске; загружается лист <имя листа>.

Пример:

возьмилист "picture

вл "story

----------------------------------------------------------------

возьмипроцедуры <имя листа> 1 аргумент

gettools команда

Процедуры, описанные на изнаночной стороне листа <имя листа>,

загружаются в память в качестве внешних процедур.

См. процедуры, сотрипроцедуры.

О внешних процедурах см. приложение 4.

Пример:

возьмипроцедуры "math

----------------------------------------------------------------

возьмирис <имя файла> 1 аргумент

вр <имя файла> команда

loadpic

Загружает на лицевую сторону текущего листа графическое изобра-

жение, сохраненное в файле <имя файла> командой "помнирис".

См. помнирис.

Пример:

возьмирис "park.pic

----------------------------------------------------------------

возьмитекст <имя файла> 1 аргумент

loadtext команда

Загружает текст, содержащийся в текстовом файле <имя файла>.

Текст может быть загружен как на изнанку, так и на лицевую сто-

рону текущего листа.

См. помнитекст.

Пример:

возьмитекст "story.txt

----------------------------------------------------------------

возьмиформы без аргументов

getshapes команда

Загружает файл с именем "shapes.shp" в качестве листа форм.

----------------------------------------------------------------

восстанови без аргументов

restore команда

Текущий лист заменяется тем его вариантом, который записан на

диске. При этом будут уничтожены все изменения, сделанные пос-

ле загрузки листа в память. Если лист новый, и на диске нет его

версии, то никаких изменений не происходит.

Пример:

возьмилист "game

иг \*исходная графика\*

повтори 4[вп 70 пр 90] \*вперед 70, направо 90\*

восстанови

- 22 -

----------------------------------------------------------------

вперед <число> 1 аргумент

вп <число> команда

forward, fd

Перемещает всех активных черепашек вперед на <число> шагов.

Замечание: в режиме перемещения, который включается нажатием

клавиши "перемещение", активную черепашку можно передвигать по

экрану "вручную" (с помощью клавиш со стрелками).

Примеры:

иг \*исходная графика\*

повтори 3 [вп 60 пр 120] \*вперед 60, направо 120\*

иг \*исходная графика\*

повтори 150 [вп 1 пр 3] \*вперед 1, направо 3\*

----------------------------------------------------------------

все без аргументов

всех

всем

all датчик

Возвращает полный список черепашек: [0 1 2 3].

Пример:

иг \*исходная графика\*

для всех \*активизирует всех черепашек\*

пч \*покажи черепашку\*

по \*перо опусти\*

вперед 100

покажи всех

В поле команд Вы увидите: [0 1 2 3]

----------------------------------------------------------------

вставь <слово/список> 1 аргумент

insert команда

Замечание: примитив допускает несколько аргументов; в том слу-

чае, когда аргументов более одного, все выражение заключается в

круглые скобки:

(вставь <слово/список1> <слово/список2>...)

Вставляет слово или список перед курсором, при этом курсор оста-

ется в конце вставленного текста.

См. пиши, пишикоманды, покажи, надпись.

Примеры:

ст \*сотри текст\*

пиши [хорошо живет на свете]

вставь [Винни Пух]

cт

(вставь [хорошо живет на свете][Винни Пух])

- 23 -

-----------------------------------------------------------------

входит? <слово/список1><слово/список2> 2 аргумента

member? логический датчик

Возвращает значение "истина", если <слово/список1> является эле-

ментом <слова/списка2>, и "ложь" - в противном случае.

Примеры:

покажи входит? "и "Винни

В поле команд вы увидите: истина

покажи входит? "Винни [Винни Пух]

В поле команд вы увидите: истина

покажи входит? "и [Винни Пух]

В поле команд вы увидите: ложь

-----------------------------------------------------------------

выделенное без аргументов

selected датчик

Возвращает в виде строки копию выделенного на листе текста.

Текст может быть выделен либо с помощью специальных клавиш, либо

с использованием примитивов "выделяй" и "ищи". Если на листе нет

выделенного текста, датчик возвращает пустое слово.

Пример:

ст \*сотри текст\*

пиши [Винни Пух]

выделяй

повтори 4[кнд] \*курсор назад\*

покажи выделенное

-----------------------------------------------------------------

выделяй без аргументов

select команда

Включает режим выделения текста. Для выделения фрагмента текста

используйте команды, перемещающие курсор: квп, кнд, кнз, квх,

вкст, внст, вкт, внт. Выделенным будет фрагмент от начального до

конечного положения курсора (исключая последнее). Команды, вызы-

вающие какие-либо изменения текста (ищи, замени, пиши), а также

команда "невыделяй" приводят к выключению режима выделения. Ре-

жим выделения включается также с помощью клавиши "выделяй".

См. невыделяй.

Пример:

ст \*сотри текст\*

пиши [Винни Пух]

выделяй

повтори 4[кнд] \*курсор назад\*

копируй \*копирует выделенный текст в буфер\*

вкт \*в конец текста\*

верни \*печатает на листе содержимое буфера\*

- 24 -

-----------------------------------------------------------------

вырежи без аргументов

cut команда

Помещает в текстовый буфер выделенный фрагмент текста и удаляет

его с листа. Того же эффекта можно достичь нажатием клавиши

"вырежи".

См. выделяй, буфер, верни.

Пример:

ст \*сотри текст\*

пиши [Винни Пух]

внт \*в начало текста\*

выделяй

повтори 5[квп] \*курсор вперед\*

вырежи

-----------------------------------------------------------------

выход <слово/список> 1 аргумент

вых <слово/список> команда

output, op

Прекращает выполнение процедуры и возвращает <слово/список>. Эта

команда может быть использована только внутри процедуры. С помо-

щью этой команды создаются процедуры-датчики.

Пример:

Опишите на изнанке следующую процедуру:

это квадрат :число

выход :число \* :число

конец

Перейдите в поле команд и напишите:

покажи квадрат 10

В поле команд Вы увидите: 100

-----------------------------------------------------------------

дваинтервала без аргументов

dspace команда

Устанавливает два интервала между строками при печати.

- 25 -

-----------------------------------------------------------------

делай <список инструкций> 1 аргумент

run команда или датчик

Выполняет <список инструкций>. Если <список инструкций> возвра-

щает некоторую информацию, "делай" возвращает точно такую же ин-

формацию.

Примеры:

иг \*исходная графика\*

назови [повтори 120[вперед 1 направо 3]] "круг

делай :круг

назови [2 + 3 + 4 + 5 + 6] "задача

покажи делай :задача

В поле команд Вы увидите результат: 20

-----------------------------------------------------------------

диск без аргументов

disk датчик

Возвращает имя текущего накопителя диска.

См. установидиск.

Пример:

покажи диск

-----------------------------------------------------------------

длина без аргументов

длину

textlen датчик

Возвращает длину текста, в символах, на текущей стороне листа.

Надписи (т.е. тексты, сделанные в режиме "надпись" или с помощью

команды "надпись") данным датчиком не учитываются.

Пример:

ст \*сотри текст\*

вставь [Винни Пух]

покажи длину

В поле команд Вы увидите: 9

-----------------------------------------------------------------

для <черепашка/список черепашек> 1 аргумент

tell команда

Задает множество активных черепашек. Именами черепашек являются

числа 0, 1, 2 и 3.

См. скажи, все, кто.

Примеры:

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

вп 30 пр 90 вп 30 \*вперед 30, направо 90, вперед 30\*

для [0 1]

cч \*спрячь черепашку\*

для 3

повтори 2[пр 90 вп 30] \*направо 90, вперед 30\*

- 26 -

-----------------------------------------------------------------

добавь <имя листа> 1 аргумент

load команда

Загружает изнанку листа <имя листа> на изнанку текущего листа.

Таким образом, к имеющимся на текущем листе процедурам добавля-

ются новые процедуры. При этом лицевая сторона текущего листа не

изменяется.

Пример:

добавь "lab

-----------------------------------------------------------------

домой без аргументов

home команда

Перемещает каждую активную черепашку в свою исходную позицию и

устанавливает ее в направлении 0 (т.е. на север).

Исходная позиция черепашки 0 - [0 0],

черепашки 1 - [0 -40],

черепашки 2 - [-40 -40],

черепашки 3 - [-40 0].

Пример:

иг \*исходная графика\*

вперед 20

скажи [1 2 3] [пч нд 20] \*покажи черепашку, назад 20\*

для [3 2] лв 90 вп 20 \*налево 90, вперед 20\*

домой

-----------------------------------------------------------------

дос без аргументов

dos команда

Сохраняет текущий лист и выходит из LogoWriter'а.

-----------------------------------------------------------------

доточки <[x y]> 1 аргумент

distance датчик

Возвращает расстояние от активной черепашки до точки <[x y]>.

Если активных черепашек несколько, то данный датчик применяет-

ся только к одной черепашке, а именно, к той, которая стоит

первой в списке, возвращаемом датчиком "кто".

Пример:

иг \*исходная графика\*

покажи доточки [20 20]

В поле команд Вы увидите результат: 28.2842

для 1

покажи доточки [20 20]

В поле команд Вы увидите результат: 63.2456

скажи 2 [покажи доточки [20 20]]

В поле команд Вы увидите результат: 84.8528

- 27 -

-----------------------------------------------------------------

если <истина/ложь> <список инструкций> 2 аргумента

if команда или датчик

Первый аргумент, <истина/ложь>, представляет собой условие, ко-

торое проверяется на истинность. Если условие истинно, <список

инструкций> выполняется, если ложно - игнорируется.

См. еслииначе.

Пример:

если 25 + 25 = 50 [пиши [25 + 25 = 50]]

-----------------------------------------------------------------

еслииначе <истина/ложь> <список инструкций1><список инструкций2>

ifelse 3 аргумента

команда или датчик

Первый аргумент - это условие, которое проверяется на истинность.

Если условие истинно, то выполняется <список инструкций1>, в

противном случае выполняется <список инструкций2>. Если <список

инструкций1> и <список инструкций2> являются датчиками, то и сам

примитив тоже будет датчиком и будет возвращать ту же информацию,

что и соответствующий список инструкций.

См. если.

Пример:

еслииначе 1 < 5 [пиши "меньше] [пиши "больше]

Напечатает на листе: меньше

-----------------------------------------------------------------

жди <число> 1 аргумент

wait команда

Приостанавливает выполнение инструкции на <число> тиков (один

тик равен приблизительно 1/20 секунды).

Пример:

иг \*исходная графика\*

повтори 22 [вперед 5]

повтори 22 [вперед 5 жди 3]

-----------------------------------------------------------------

ждикоманды без аргументов

readlistcc датчик

Приостанавливает выполнение инструкции до тех пор, пока в поле

команд не будет введена строка, и возвращает последнюю в виде

списка. Введенная строка должна сопровождаться нажатием клавиши

"ввод". Данный датчик считывает только ту строку, которая была

введена в порядке выполнения примитива "ждикоманды". Никакие

строки, введенные до начала выполнения примитива, считываться не

будут.

См. ждисписок, ждисимвол.

Пример:

Опишите такую процедуру на изнанке:

это диалог

ск \*сотри команды\*

пишикоманды [напиши свое имя: ]

назови ждикоманды "имя

конец

Перейдите в поле команд и дайте инструкцию:

диалог

Напечатайте Ваше имя, нажмите клавишу "ввод" и дайте команду:

покажи :имя

- 28 -

-----------------------------------------------------------------

ждисимвол без аргументов

жс датчик

readchar

Приостанавливает выполнение инструкции до тех пор, пока с клави-

атуры не будет введен символ, считывает и возвращает последний.

Считываемый символ не печатается на экране.

См. ждисписок, ждикоманды.

Примеры:

еслииначе ждисимвол = "а [пиши [очень приятно]] [пиши

[ничего подобного]]

Введите какой-нибудь символ с клавиатуры

покажи код ждисимвол

Введите какой-нибудь символ с клавиатуры

-----------------------------------------------------------------

ждисписок без аргументов

readlist датчик

Приостанавливает выполнение инструкции до тех пор, пока на листе

не будет напечатана строка, считывает и возвращает последнюю в

виде списка. Напечатанная строка должна сопровождаться нажатием

клавиши "ввод". Данный датчик считывает только ту строку, которая

была введена в порядке выполнения примитива "ждисписок". Никакие

строки, введенные до начала выполнения примитива, считываться не

будут.

См. ждикоманды, ждисимвол.

Пример:

Опишите такую процедуру на изнанке:

это диалог

ст \*сотри текст\*

пиши [напиши свое имя:]

назови ждисписок "имя

вставь [я очень рада,]

пиши :имя

конец

Перейдите в поле команд и дайте инструкцию:

диалог

Теперь напечатайте Ваше имя и нажмите клавишу "ввод".

- 29 -

-----------------------------------------------------------------

замени <слово1> <слово2> 2 аргумента

replace команда

Вызывает поиск <слова1>, начиная от текущего положения курсора

до конца текста, и замену первого вхождения <слова1> на <слово2>.

Если <слово1> не найдено, положение курсора не меняется.

См. ищи, нашла.

Пример:

ст \*сотри текст\*

пиши [быть или не быть]

внт \*в начало текста\*

замени "быть "мыть

замени "быть "мыть

На листе будет написано: мыть или не мыть

-----------------------------------------------------------------

заполни без аргументов

shade команда

Заполняет лист или замкнутую область, в которой находится центр

черепашки, изображениями текущей формы черепашки. LogoWriter не

может заполнить область, которая по размеру меньше черепашки или

имеет слишком сложную конфигурацию.

Пример:

иг \*исходная графика\*

повтори 4 [вп 70 пр 90] \*вперед 70, направо 90\*

пп \*перо подними\*

направо 45

вперед 30

по \*перо опусти\*

заполни

-----------------------------------------------------------------

запятая без аргументов

setcomma команда

Устанавливает десятичную запятую для записи вещественных чисел.

См. точка.

-----------------------------------------------------------------

защита без аргументов

lock команда

Устанавливает защиту текущего листа. При этом в верхнем левом

углу появляется слово "защищено". Защищенный лист не может быть

удален с диска, и никакие изменения защищенного листа на диск

записаны не будут. Защита с листа может быть снята только с по-

мощью команды "беззащиты". Команды "защита" и "беззащиты" дейс-

твуют лишь по отношению к текущему листу.

См. беззащиты.

- 30 -

-----------------------------------------------------------------

звук <частота> <время> 2 аргумента

tone команда

Производит звук заданной частоты и длительности. Время в Logo-

Writer'е измеряется в тиках; один тик приблизительно равняется

1/20 сек.

Примеры:

ст \*сотри текст\*

повтори 3[звук 523 20 звук 659 10 звук 784 20 пиши

[я играю]]

Таблицу частот по октавам см. в приложении 5.

-----------------------------------------------------------------

значение <имя> 1 аргумент

thing датчик

Возвращает значение переменной <имя>. Переменные создаются кома-

ндами "назови" и "пусть".

См. назови, пусть.

Примеры:

назови [Аня Петя Дима] "дети

пиши значение "дети

На листе напечатает: Аня Петя Дима

назови [отличный парень] "Дима

пиши значение "Дима

На листе напечатает: отличный парень

ст \*сотри текст\*

пиши значение псл :дети \*последний\*

На листе напечатает: отличный парень

-----------------------------------------------------------------

и <истина/ложь1> <истина/ложь2> 2 аргумента

and логический датчик

Замечание: данный примитив допускает более двух аргументов; в

том случае, когда аргументов три и больше, примитив вместе со

всеми аргументами заключается в круглые скобки:

(и <истина/ложь1> <истина/ложь2> <истина/ложь3>...)

Возвращает значение "истина", если все аргументы истинны, и

"ложь", если хотя бы один из аргументов ложен.

Примеры:

ст \*сотри текст\*

пиши и "истина "ложь

На листе напечатает: ложь

пиши и 5 = 5 "а = "а

На листе напечатает: истина

- 31 -

-----------------------------------------------------------------

иг без аргументов

rg команда

Сокращение от "Исходная Графика". Приводит графическую систему в

исходное состояние : стирается все графическое изображение, фон

приобретает черный цвет, черепашка принимает стандартную форму,

опускает перо и окрашивается в белый цвет. Видимой и активной

становится лишь черепашка с номером 0.

Пример:

вп 50 \*вперед 50\*

нф 12 \*новая форма 12\*

нц 2 \*новый цвет 2\*

для всех

пч \*покажи черепашку\*

надпись [у-у-у]

новфон 3

иг

-----------------------------------------------------------------

ил <имя листа> 1 аргумент

namepage, np команда

Сокращение от "Имя Листа". Присваивает имя текущему листу.

Пример:

нл \*новый лист\*

ил "flower

-----------------------------------------------------------------

или <истина/ложь1> <истина/ложь2> 2 аргумента

or логический датчик

Замечание: данный примитив допускает более двух аргументов; в

том случае, когда аргументов три и больше, примитив вместе со

всеми аргументами заключается в круглые скобки:

(или <истина/ложь1> <истина/ложь2> <истина/ложь3>...)

Возвращает значение "истина", если хотя бы один из аргументов

истинен, и "ложь" - в противном случае.

Примеры:

ст \*сотри текст\*

пиши или "истина "ложь

На листе напишет: истина

пиши или 5 = 2 "а = "а

На листе напишет: истина

- 32 -

-----------------------------------------------------------------

имя? <слово> 1 аргумент

name? логический датчик

Возвращает значение "истина", если <слово> является именем суще-

ствующей переменной (т.е. переменной с именем <слово> присвоено

значение), и "ложь" - в противном случае. Создать переменную или

присвоить значение существующей переменной можно командами "на-

зови" и "пусть".

См. назови, пусть.

Примеры:

назови [Винни Пух] "зверь

пиши имя? "зверь

На листе напишет: истина

пусть "а 100

пиши имя? "а

На листе напишет: истина

-----------------------------------------------------------------

истина специальное слово

true

Используется в качестве аргумента примитивов: если, еслииначе,

и, или , не.

См. ложь.

Примеры:

ст \*сотри текст\*

если "истина [пиши "Даша]

На листе напишет: Даша

еслииначе и "истина "ложь [пиши "Даша] [пиши "Маша]

На листе напишет: Маша

-----------------------------------------------------------------

ищи <слово> 1 аргумент

search команда

Ищет первое вхождение <слова> как отдельного слова или подслова

в тексте листа, начиная от текущего положения курсора до конца

текста, и выделяет его (при этом включается режим выделения).

Для отмены выделения используется команда "невыделяй".

См. замени, нашла?, невыделяй.

Примеры:

ст \*сотри текст\*

пиши [Винни Пух]

внт \*в начало текста\*

ищи "Пух

невыделяй

внт

ищи "ин

вырежи

- 33 -

-----------------------------------------------------------------

каждая <список инструкций> 1 аргумент

each команда

Все активные черепашки по очереди выполняют инструкции, содер-

жащиеся в <списке инструкций>. Это означает, что до тех пор, по-

ка очередная черепашка не выполнила все инструкции, следующая не

делает ничего. Список активных черепашек задается командами

"скажи" и "для".

Примеры:

иг \*исходная графика\*

для [0 1 2 3]

пч \*покажи черепашку\*

каждая [повтори 4[направо 90 вперед 70]]

скажи [1 3] [каждая [вперед 60]]

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

каждая [новкурс 90 \* кто вперед 50]

-----------------------------------------------------------------

квп без аргументов

cf команда

Сокращение от "Курсор ВПеред". Перемещает курсор на текущей сто-

роне листа на один символ вперед. Эту же операцию можно выпол-

нить "вручную" с помощью стрелки, направленной вправо.

Пример:

ст \*сотри текст\*

пиши "Чебурашка

внт \*в начало текста\*

выделяй

повтори 9 [квп]

-----------------------------------------------------------------

квх без аргументов

cu команда

Сокращение от "Курсор ВверХ". Перемещает курсор на текущей сто-

роне листа на предыдущую строку. Эту же операцию можно выполнить

"вручную" с помощью стрелки, направленной вверх.

Пример:

ст \*сотри текст\*

пиши "Винни

пиши "Пух

выделяй

повтори 2 [квх]

-----------------------------------------------------------------

кк <число> 1 аргумент

sqrt датчик

Сокращение от "Квадратный Корень". Возвращает значение квадрат-

ного корня из <числа>.

Примеры:

покажи кк 81

В поле команд напишет: 9

пиши [квадратный корень из числа 300 есть ]

вставь кк 300

- 34 -

-----------------------------------------------------------------

клавиша? без аргументов

key? логический датчик

Возвращает значение "истина", если на клавиатуре была нажата ка-

кая-нибудь клавиша, и "ложь" - в противном случае. Клавиша не

должна считываться датчиками "ждисимвол", "ждисписок", "ждикома-

нды".

Примеры:

покажи клавиша?

В поле команд напишет: ложь

Опишите на изнанке листа такую процедуру:

это пример

еслииначе клавиша? [формы стоп] [пиши [нажмите

клавишу] жди 20]

пример

конец

Перейдите в поле команд и дайте инструкцию:

пример

Нажмите любую клавишу

-----------------------------------------------------------------

кнд без аргументов

cb команда

Сокращение от "Курсор НазаД". Перемещает курсор на текущей сто-

роне листа на один символ назад. Эту же операцию можно проде-

лать "вручную" с помощью стрелки, направленной влево.

Пример:

ст \*сотри текст\*

пиши [Винни Пух]

выделяй

повтори 3 [кнд]

вырежи

-----------------------------------------------------------------

кнз без аргументов

cd команда

Сокращение от "Курсор вНиЗ". Перемещает курсор на следующую

строку. Эту же операцию можно проделать "вручную" с помощью

стрелки, направленной вниз.

Пример:

ст \*сотри текст\*

пиши "Винни

пиши "Пух

внт \*в начало текста\*

выделяй

кнз

- 35 -

-----------------------------------------------------------------

кнопка? <номер кнопки> 1 аргумент

button? логический датчик

Возвращает значение "истина", если на джойстике нажата кнопка

<номер кнопки>, и "ложь" - в противном случае.

Пример:

покажи кнопка? 1

-----------------------------------------------------------------

код <символ> 1 аргумент

ascii датчик

Возвращает число, равное ASCII-коду <символа>. ASCII - это аме-

риканский стандартный код для обмена информацией.

Таблицу ASCII-кодов см. в приложении 6.

Пример:

покажи код "а

В поле команд напишет: 160 (если это буква русского алфавита)

-----------------------------------------------------------------

конец без аргументов

end специальное слово

Словом "конец" должно заканчиваться описание каждой процедуры на

изнанке листа. Открывает процедуру слово "это".

Примеры:

это квадрат

повтори 4 [вп 40 пр 90] \*вперед 40, направо 90\*

конец

это треугольник

повтори 3 [вп 40 пр 120] \*вперед 30, направо 120\*

конец

Перейдите в поле команд и дайте инструкции:

иг \*исходная графика\*

квадрат вперед 40 направо 30

треугольник

Черепашка нарисует дом

- 36 -

-----------------------------------------------------------------

контроль <буква> <список инструкций> 2 аргумента

when команда

Программирует контрольные клавиши. При нажатии запрограммирован-

ной контрольной клавиши вместе с клавишей "Ctrl" немедленно вы-

полняется <список инструкций>, независимо от того, выполняется

ли в данный момент какая-либо инструкция или нет. В качестве ко-

нтрольных могут использоваться 10 клавиш: N, O, P, Q, R, V, W,

X, Y, Z. Информация о контрольных клавишах по окончании сеанса

работы системы не сохраняется.

Замечание: если программа ждет ввода с клавиатуры символа/ спис-

ка в порядке выполнения примитивов "ждисимвол", "ждисписок",

"ждикоманды", то нажатие контрольной клавиши не даст ожидаемого

эффекта. См. сотриконтроль.

Примеры:

контроль "z [формы]

контроль "n [для всех

пч жди 15 сч] \*покажи черепашку...спрячь черепашку\*

Нажмите клавиши: Ctrl+N и через некоторое время Ctrl+Z

-----------------------------------------------------------------

копируй без аргументов

copy команда

Помещает копию предварительно выделенного текста в буфер. Того

же эффекта можно достичь нажатием клавиши "копируй".

См. выделяй, вырежи, буфер, верни.

Пример:

ст \*сотри текст\*

пиши [хорошо живет на свете]

выделяй

повтори 9 [кнд] \*курсор назад\*

копируй

вкт \*в конец текста\*

пиши буфер

-----------------------------------------------------------------

копируйфайл <имя файла1> <имя файла2> 2 аргумента

copyfile команда

Копирует файл <имя файла1> в файл <имя файла2>.

Пример:

копируйфайл "history "archive

-----------------------------------------------------------------

кпрв <слово/список> 1 аргумент

butfirst, bf датчик

Сокращение от "Кроме ПеРВого". Возвращает <слово/список> без

первого элемента.

Примеры:

ст \*сотри текст\*

пиши кпрв "Пух

На листе напишет: ух

пиши кпрв [хорошо живет на свете]

На листе напишет: живет на свете

- 37 -

-----------------------------------------------------------------

кпсл <слово/список> 1 аргумент

butlast, bl датчик

Сокращение от "Кроме ПоСЛеднего". Возвращает <слово/список> без

последнего элемента.

Примеры:

ст \*сотри текст\*

пиши кпсл "Пух

На листе напишет: Пу

пиши кпсл [хорошо живет на свете]

На листе напишет: хорошо живет на

-----------------------------------------------------------------

крась без аргументов

fill команда

Закрашивает замкнутую область или весь экран цветом черепашки.

Закрашивание начинается в том месте, где находится перо черепаш-

ки, т.е. в центре черепашки. Если перо черепашки находится на

черте, то окрашивается только черта.

Пример:

иг \*исходная графика\*

повтори 4 [вперед 50 направо 90]

пп \*перо подними\*

пр 45 \*направо 45\*

вп 20 \*вперед 20\*

по \*перо опусти\*

крась

-----------------------------------------------------------------

кто без аргументов

who датчик

Возвращает номер активной черепашки или список активных черепа-

шек. Если "кто" используется в списке инструкций для команды

"каждая", то "кто" всякий раз возвращает один номер.

См. для, скажи, каждая.

Примеры:

иг \*исходная графика\*

для [0 1 2 3]

пч \*покажи черепашку\*

вп 50 \*вперед 50\*

покажи кто

В поле команд напишет: [0 1 2 3]

- 38 -

-----------------------------------------------------------------

курс без аргументов

heading датчик

Возвращает значение направления активной черепашки в градусах,

от 0 до 359,9999. Если активными являются несколько черепашек,

то примитив применяется только к одной черепашке, а именно, к

той, которая стоит первой в списке, возвращаемом датчиком "кто".

Пример:

иг \*исходная графика\*

покажи курс

В поле команд напишет: 0

направо 90

покажи курс

В поле команд напишет: 90

-----------------------------------------------------------------

листы без аргументов

pagelist датчик

Возвращает список всех имеющихся на диске листов.

Пример:

покажи листы

-----------------------------------------------------------------

лицо? без аргументов

front? логический датчик

Возвращает значение "истина", если текущая сторона листа лицевая,

и "ложь" - в противном случае.

Примеры:

покажи лицо?

если лицо? [переверни]

-----------------------------------------------------------------

ложь без аргументов

false специальное слово

Используется в качестве аргумента примитивов "если", "еслииначе",

"и", "или", "не".

Пример:

покажи и "истина "ложь

-----------------------------------------------------------------

медленно без аргументов

slowturtle команда

Устанавливает медленный режим движения черепашки.

См. быстро.

Пример:

иг \*исходная графика\*

медленно

повтори 200[ направо 3 вперед 1]

быстро

повтори 200[направо 3 вперед 1]

- 39 -

-----------------------------------------------------------------

место без аргументов

pos датчик

Возвращает значение текущих координат черепашки в виде списка:

[x y]. Если активных черепашек несколько, то примитив применяет-

ся только к одной черепашке, а именно, к той, которая стоит

первой в списке, возвращаемом датчиком "кто". Координаты чере-

пашки могут изменяться в пределах от -159 до 160 (по оси х) и

от -94 до 95 (по оси у).

В состоянии исходной графики черепашка 0 находится в точке [0 0],

черепашка 1 - в точке [0 -40], черепашка 2 - в точке [-40 -40],

черепашка 3 - в точке [-40 0].

См. новместо.

Пример:

иг \*исходная графика\*

покажи место

В поле команд напишет: [0 0]

скажи 1 [покажи место]

В поле команд напишет: [0 -40]

-----------------------------------------------------------------

минус <число> 1 аргумент

minus датчик

Унарный минус. Используется в тех случаях, когда численное зна-

чение выражается не цифрами, а словом.

Примеры:

иг \*исходная графика\*

для [0 1 2 3]

пч \*покажи черепашку\*

по \*перо опусти\*

каждая [новкурс 90 \* минус кто вперед 50]

-----------------------------------------------------------------

мк без аргументов

cursorpos датчик

Сокращение от "Место Курсора". Возвращает значение текущего по-

ложения курсора в системе координат, используемых для определе-

ния положения черепашки.

См. место, поз.

Примеры:

ст \*сотри текст\*

повтори 5[пиши [где я?]]

кнд \*курсор назад\*

пп \*перо подними\*

новместо мк

повтори 3 [квх] \*курсор вверх\*

новместо мк

- 40 -

------------------------------ -----------------------------------

мусор без аргументов

recycle команда

Выполняет операцию "сборки мусора" в памяти машины, освобождая

неиспользованное пространство. Использование этой команды в сло-

жных программах ускоряет выполнение программ, предупреждает за-

медление работы программ в тех местах, где это нежелательно.

Рекомендуется использовать команду "мусор" перед применением

датчика "память".

См. память.

Пример:

мусор

покажи память

-----------------------------------------------------------------

надпись <слово/список> 1 аргумент

lable команда

Печатает <слово/список>, начиная от положения всех активных че-

репашек, в качестве так называемой надписи. Надписи являются

частью графического изображения, к ним не применимы команды тек-

стового редактора, а также соответствующие им специальные клави-

ши. Надпись можно стереть, либо сделав точно такую же надпись в

том же самом месте, либо командами "иг", "сг" и "сотри".

Нажатием клавиши "надпись" включается режим "надпись", в котором

надписи можно делать "вручную" (т.е. просто нажимая на нужные

клавиши).

См. пиши, вставь.

Примеры:

иг \*исходная графика\*

пиши [Винни Пух]

надпись [Винни Пух]

для [0 1]

пч \*покажи черепашку\*

надпись [Винни Пух]

ст \*сотри текст\*

сг \*сотри графику\*

-----------------------------------------------------------------

назад <число> 1 аргумент

нд <число> команда

back, bk

Перемещает всех активных черепашек назад на <число> шагов.

Замечание: в режиме перемещения (который включается с помощью

клавиши "перемещение") активную черепашку можно перемещать по

экрану "вручную" (с помощью стрелок, управляющих курсором).

Пример:

иг \*исходная графика\*

нд 50

пр 90 \*направо 90\*

нд 50

- 41 -

-----------------------------------------------------------------

назови <слово/список> <имя> 2 аргумента

name команда

Переменной <имя> присваивается значение <слово/список>. Если

перменной с таким именем ранее не существовало, то команда соз-

дает новую переменную. Если имя переменной было указано в заго-

ловке работающей процедуры (см. примитив "это"), то переменная

является локальной, то есть после завершения работы процедуры в

переменной будет восстановлено значение, имевшееся в ней до на-

чала работы процедуры. Все остальные переменные являются гло-

бальными, то есть значение таких переменных можно использовать и

изменять в любом месте программы, в котором отсутствует локаль-

ная переменная с тем же именем.

Аналогом данной команды является команда "пусть", последняя, од-

нако, требует противоположного порядка следования аргументов.

Значения переменных в памяти не сохраняются после окончания се-

анса работы системы. Для обращения к значению переменной исполь-

зуется следующая запись: ":<имя\_переменной>". Кроме того, значе-

ние переменной может быть получено с помощью датчика "значение".

См. пусть, это, сотриимя, сотриимена, значение.

Пример:

назови 0 "черный

новфон 2

новфон :черный

-----------------------------------------------------------------

налево <число> 1 аргумент

лв <число> команда

left, lt

Поворачивает всех активных черепашек налево на <число> градусов.

Пример:

иг \*исходная графика\*

повтори 4[налево 90 вперед 50]

-----------------------------------------------------------------

направо <число> 1 аргумент

пр <число> команда

right, rt

Поворачивает всех активных черепашек направо на <число> градусов.

Пример:

иг \*исходная графика\*

повтори 4[пр 90 вперед 50]

- 42 -

-----------------------------------------------------------------

нашла? без аргументов

found? логический датчик

Используется после выполнения команд "ищи" или "замени". Возвра-

щает значение "истина", если предшествующий поиск слова закончи-

лся успешно, и "ложь" - в противном случае.

Пример:

ст \*сотри текст\*

пиши [Винни Пух]

внт \*в начало текста\*

ищи "у

если нашла? [вставь "уууу]

-----------------------------------------------------------------

не <истина/ложь> 1 аргумент

not логический датчик

Возвращает значение, противоположное значению аргумента. Таким

образом, если <истина/ложь> истинно, "не" возвращает значение

"ложь", и наоборот.

Примеры:

пиши не "истина

На листе напишет: ложь

пиши 1 = 2

На листе напишет: ложь

пиши не 1 = 2

На листе напишет: истина

-----------------------------------------------------------------

невыделяй без аргументов

unselect команда

Осуществляет выход из режима выделения. Символы, ранее выделен-

ные в режиме выделения (т.е. окрашенные в реверсивный цвет),

перестают выделяться. Команда используется также для отмены вы-

деления после выполнения команды "ищи".

См. выделяй, ищи.

Пример:

ст \*сотри текст\*

вставь [Винни Пух]

выделяй

внст \*в начало строки\*

невыделяй

ищи "Пух

невыделяй

- 43 -

-----------------------------------------------------------------

новкурс <число> 1 аргумент

нк <число> команда

seth

Cокращение от "Новый Курс". Устанавливает всех активных черепа-

шек в направлении, соответствующем <числу> градусов.

0 соответствует направлению на север, 90 - на восток, 180 - на

юг, 270 - на запад.

Cм. курс.

Примеры:

покажи курс

новкурс 45

вперед 50

нк 180

вперед 50

-----------------------------------------------------------------

новлист без аргументов

нл команда

newpage

Сокращение от Новый Лист". Сохраняет текущий лист (если он име-

ет имя) и переходит к новому листу. Новый лист не имеет имени

(вместо имени стоит ???). Лист не может быть сохранен на диске,

если ему не присвоено имя командой "ил".

Пример:

новлист

ил "games \*имя листа\*

-----------------------------------------------------------------

новместо [x y] 1 аргумент

нм [x y] команда

setpos

Сокращение от "Новое Место". Перемещает всех активных черепашек

в точку [x y].

См. место.

Пример:

иг \*исходная графика\*

покажи место

В поле команд напишет: [0 0]

новместо [50 0]

нм [50 50]

-----------------------------------------------------------------

новпоз <число> 1 аргумент

settextpos команда

Сокращение от "НОВая ПОЗиция". Устанавливает курсор в позицию,

отстоящую от начала текста на <число> символов.

См. поз.

Примеры:

ст \*сотри текст\*

пиши [скорее открой]

новпоз 2

выделяй

новпоз поз + 8

вырежи

- 44 -

-----------------------------------------------------------------

новфон <число> 1 аргумент

setbg команда

Сокращение от "НОВый ФОН". Устанавливает цвет фона экрана, соот-

ветствующий <числу>.

См. фон.

Таблицу цветов см. в приложении 5.

Примеры:

иг \*исходная графика\*

новфон 4

покажи фон

В поле команд напишет: 4

иг

покажи фон

В поле команд напишет: 0

-----------------------------------------------------------------

новформа <число> 1 аргумент

нф <число> команда

setsh

Cокращение от "Новая Форма". Все активные черепашки принимают

форму с номером <число>.

См. форма.

Примеры:

иг \*исходная графика\*

нф 11

покажи форму

В поле команд напишет: 11

нф форма + 1

-----------------------------------------------------------------

новцвет <число> 1 аргумент

нц <число> команда

setc

Cокращение от "Новый Цвет". Перекрашивает всех активных черепа-

шек в новый цвет, соответствующий <числу>. Перо у черепашки тако-

го же цвета, что и сама черепашка.

См. цвет.

Таблицу цветов см. в приложении 5.

Пример:

иг \*исходная графика\*

нц 2

вперед 40

вперед 40

покажи цвет

В поле команд напишет: 2

нц 3

- 45 -

-----------------------------------------------------------------

новцт <число> 1 аргумент

нцт <число>

settc команда

Сокращение от "НОВый Цвет Текста". Текст, вводимый на лист после

выполнения команды "нцт", будет окрашен в соответствующий <чис-

лу> цвет. Цвет текста, введенного до команды, не изменится.

См. цт (цвет текста).

Таблицу цветов см. в приложении 5.

Примеры:

ст \*сотри текст\*

повтори 5 [пиши [Винни Пух]]

нцт 2

пиши [хорошо живет на свете]

покажи цт \*цвет текста\*

В поле команд напишет: 2

-----------------------------------------------------------------

оглавление без аргументов

contents команда

Сохраняет текущий лист (если у него есть имя) и переходит к

оглавлению. Оглавление представляет собой меню системы, находясь

в котором Вы можете выбрать для работы любой из имеющихся на

диске листов (в том числе специальный лист форм) или новый (т.е.

пустой) лист.

-----------------------------------------------------------------

одинаковы? <слово/список1> <слово/список2> 2 аргумента

identical? логический датчик

Возвращает значение "истина", если аргументы представляют собой

равные числа, одинаковые слова или одинаковые списки. Проверка

идентичности различает прописные и строчные буквы. Слово и спи-

сок, содержащий ровно это слово, не являются одинаковыми.

См. равны?, = .

Замечание: русская и латинская буквы, имеющие сходные начерта-

ния, - суть разные буквы.

Примеры:

ст \*сотри текст\*

пиши одинаковы? "а "А

На листе напишет: ложь

пиши одинаковы? "один [один]

На листе напишет: ложь

- 46 -

-----------------------------------------------------------------

одининтервал без аргументов

sspace команда

Устанавливает один интервал между строками при печати.

См. дваинтервала.

-----------------------------------------------------------------

ол без аргументов

cp команда

Сокращение от "Очисть Лист". Уничтожает графику и текст на обеих

сторонах текущего листа.

Замечание: эта команда уничтожает в том числе и процедуры, опи-

санные на изнанке листа. Если содержимое листа было уничтожено

по ошибке, командой "восстанови" можно восстановить имеющуюся

на диске версию текущего листа.

См. восстанови.

-----------------------------------------------------------------

ос <слово/список1> <слово/список2> 2 аргумента

sentence, se датчик

Замечание: "ос" допускает более двух аргументов; в том случае,

когда аргументов три и более, датчик вместе со всеми аргументами

заключается в круглые скобки:

(ос <слово/список1> <слово/список2> <слово/список3>...)

Сокращение от "Объединенный Список". Возвращает список, элемен-

тами которого являются <слово/список1> и <слово/список2>. Если

аргументами "ос" являются списки, то их внешние скобки раскрыва-

ются.

См. список.

Примеры:

покажи ос "Винни "Пух

В поле команд Вы увидите: [Винни Пух]

покажи ос [Винни] [Пух]

В поле команд Вы увидите: [Винни Пух]

покажи ос "Винни [Пух]

В поле команд Вы увидите: [Винни Пух]

-----------------------------------------------------------------

останов без аргументов

stopall команда

Прекращает выполнение всех процедур и возвращает управление в

поле команд. Того же эффекта можно достичь нажатием клавиши

"останов".

Пример:

иг \*исходная графика\*

контроль "N [останов]

повтори 10000 [вперед 1 направо 1]

Нажмите контрольную клавишу "Ctrl+N"

- 47 -

-----------------------------------------------------------------

остаток <число1> <число2> 2 аргумента

remainder датчик

Возвращает значение остатка от деления <числа1> на <число2>.

Примеры:

ст \*сотри текст\*

пиши остаток 12 5

На листе напишет: 2

пиши остаток 10 5

На листе напишет: 0

-----------------------------------------------------------------

оч <число> 1 аргумент

round датчик

Сокращение от "Округленное Число". Возвращает значение округлен-

ного до целого <числа>.

Примеры:

иг \*исходная графика\*

вперед оч 18.6

покажи оч доточки [70 70]

В поле команд напишет: 87

-----------------------------------------------------------------

память без аргументов

space датчик

Возвращает число, выражающее, в байтах, размер свободной памяти

компьютера, доступной для Вашего использования при работе с те-

кущим листом.

Для увеличения объема свободной памяти используйте команду "му-

сор".

Пример:

мусор

покажи память

-----------------------------------------------------------------

первый <слово/список> 1 аргумент

прв <слово/список> датчик

first

Возвращает первый элемент <слова/списка>. Первый элемент слова -

символ, первый элемент списка может быть словом или списком.

Примеры:

ст \*сотри текст\*

пиши прв "элемент

На листе напишет: э

пиши прв [хорошо живет на свете]

На листе напишет: хорошо

пиши прв 88888

На листе напишет: 8

- 48 -

-----------------------------------------------------------------

переверни без аргументов

flip команда

Переворачивает лист с изнанки на лицо и наоборот. Того же эффек-

та можно достичь нажатием клавиши "переверни".

Пример:

переверни

если лицо? [переверни]

-----------------------------------------------------------------

переименуй <имя файла1> <имя файла2> 2 аргумента

rename команда

Файлу <имя файла1> присваивает имя <имя файла2>.

Пример:

переименуй "story "history

-----------------------------------------------------------------

печатай без аргументов

printtext команда

Подает на печатающее устройство (принтер) всю текстовую информа-

цию (как видимую, так и не видимую на экране) с текущей стороны

листа. Надписи (т.е. тексты, написанные с помощью команды "над-

пись" или в режиме "надпись") являются частью графической инфор-

мации и на принтер не подаются.

См. печатай80, экран.

Пример:

повтори 8 [пиши [на печать]]

печатай

-----------------------------------------------------------------

печатай80 без аргументов

printtext80 команда

Подает на печатающее устройство (принтер) в 80-тиколонковом ре-

жиме всю текстовую информацию (как видимую, так и не видимую на

экране) с текущей стороны листа. Надписи (т.е. тексты, написан-

ные с помощью команды "надпись" или в режиме "надпись") являются

частью графической информации и на принтер не подаются.

См. печатай, экран.

Замечание: т.к. на экране текстовая информация располагается в

40 колонках, следует следить за тем, чтобы при вводе текстов на

лист клавиша "ввод" нажималась только в конце абзаца и непремен-

но в конце каждого абзаца.

Примеры:

повтори 40 [вставь [абракадабра]]

повтори 4 [пиши [проверка печати]]

печатай80

- 49 -

-----------------------------------------------------------------

пиши <слово/список> 1 аргумент

п <слово/список> команда

print

Замечание: команда "пиши" допускает несколько аргументов; в том

случае, когда аргументов два и более, команда вместе со всеми

аргументами заключается в круглые скобки:

(пиши <слово/список1> <слово/список2>...)

Действие команды "пиши" заключает в себе две операции (которые

можно выполнить и вручную): во-первых, на текущей стороне листа

печатается <слово/список> (при этом внешние скобки списка не

печатаются) и, во-вторых, в представлении текста в памяти маши-

ны (но не на экране!) появляется метка конца строки, что рав-

нозначно нажатию клавиши "ввод".

См. вставь, покажи, пишикоманды, надпись.

Примеры:

ст \*сотри текст\*

пиши "хорошо

п [хорошо живет]

(п [хорошо живет] [на свете] "Винни "Пух)

-----------------------------------------------------------------

пишиимена без аргументов

printnames команда

Печатает на текущей стороне листа все имеющиеся в памяти пере-

менные и их значения. Переменные создаются командами "пусть" и

"назови".

См. пусть, назови, покажиимена,имя?.

Примеры:

пусть "дети [Петя Маша Коля]

пусть "Петя "друг

назови 8 "число

пишиимена

-----------------------------------------------------------------

пишикоманды <слово/список> 1 аргумент

пк <слово/список> команда

type

Замечание: "пишикоманды" допускает несколько аргументов; в том

случае, когда аргументов два и более, примитив вместе со всеми

аргументами заключается в круглые скобки:

(пишикоманды <слово/список1> <слово/список2>...)

Печатает <слово/список> в поле команд. Внешние скобки списка не

печатаются.

См. покажи, пиши, вставь.

Пример:

пишикоманды "переверни

пк "иг \*исходная графика\*

- 50 -

-----------------------------------------------------------------

по без аргументов

pd команда

Сокращение от "Перо Опусти". Опускает перо у всех активных чере-

пашек. При движении черепашки опущенное перо рисует. В состоянии

исходной графики перо опущено у всех черепашек.

Пример:

иг \*исходная графика\*

направо 90

повтори 50 [по вперед 3 жди 5 пп вперед 3 жди 5]

-----------------------------------------------------------------

повтори <число> <список инструкций> 2 аргумента

repeat команда

Повторяет <список инструкций> <число> раз.

Примеры:

повтори 4 [вперед 50 направо 90]

повтори 6 [пиши [ в пампасы! ]]

-----------------------------------------------------------------

поз без аргументов

textpos датчик

Сокращение от "ПОЗиция". Возвращает число, выражающее, в симво-

лах, позицию курсора относительно начала текста.

См. новпоз.

Пример:

ст \*сотри текст\*

пиши [все это очень благородно]

квх \*курсор вверх\*

повтори 10 [квп] \*курсор вперед\*

покажи поз

В поле команд напишет: 10

-----------------------------------------------------------------

покажи <слово/список> 1 аргумент

show команда

Замечание: команда "покажи" допускает несколько аргументов; в

том случае, когда аргументов два и более, команда вместе со все-

ми аргументами заключается в круглые скобки:

(покажи <слово/список1> <слово/список2>...)

Печатает <слово/список> в поле команд. Список печатается вместе

с внешними скобками.

См. пишикоманды, пиши, вставь, надпись.

Примеры:

ск \*сотри команды\*

покажи "Логорайтер

покажи [Пифагоровы штаны]

покажи остаток 100 7

- 51 -

-------------------------- --------------------------------------

покажиимена без аргументов

shownames команда

Пишет в поле команд все имеющиеся в памяти переменные и их зна-

чения. Переменные создаются командами "пусть" и "назови".

См. пусть, назови, пишиимена.

Примеры:

пусть "дети [Петя Толя Даша]

назови 50 "штраф

назови "август "месяц

покажиимена

-----------------------------------------------------------------

помнилист без аргументов

savepage команда

Записывает текущий лист на диск (если ему присвоено имя коман-

дой "ил").

Пример:

новлист

ил "test

помнилист

-----------------------------------------------------------------

помнирис <имя файла> 1 аргумент

savepic команда

Сокращение от "ПОМНИ РИСунок". Сохраняет графическую информацию

с текущего листа в файле <имя файла> в текущем каталоге.

Пример:

помнирис "plan.pic

-----------------------------------------------------------------

помнитекст <имя файла> 1 аргумент

savetext команда

Сохраняет текстовую информацию с текущей стороны текущего листа

в файле <имя файла> в текущем каталоге.

Пример:

помнитекст "story.txt

- 52 -

-----------------------------------------------------------------

последний <слово/список> 1 аргумент

псл <слово/список> датчик

last

Возвращает последний элемент <слова/списка>. Элементом слова яв-

ляется символ, элементом списка может быть слово или список.

См. первый.

Примеры:

ст \*сотри текст\*

пиши псл "лошадь

На листе напишет: ь

пиши псл 1234

На листе напишет: 4

пиши псл [кто виноват]

На листе напишет: виноват

-----------------------------------------------------------------

пп без аргументов

pu команда

Сокращение от "Перо Подними". Поднимает перо у всех активных че-

репашек. При движении черепашки поднятое перо не рисует и не

стирает.

См. по (перо опусти).

Пример:

иг \*исходная графика\*

направо 90

повтори 50 [по вперед 3 жди 5 пп вперед 3 жди 5]

-----------------------------------------------------------------

предэкран без аргументов

prescreen команда

Сокращение от "ПРЕДыдущий ЭКРАН". Выводит на экран предыдущую

порцию текста, т.е. предшествующие видимому тексту 19 строк.

Аналогичного эффекта можно достичь нажатием клавиши "предыду-

щий экран".

См. следэкран.

-----------------------------------------------------------------

процедуры без аргументов

toollist датчик

Возвращает список имен внешних процедур, имеющихся в памяти.

Внешние процедуры загружаются в память командой "возьмипроцеду-

ры". Изнанка текущего листа не содержит описания внешних проце-

дур.

См. возьмипроцедуры, сотрипроцедуры.

О внешних процедурах см. приложение 4.

Пример:

возьмипроцедуры "checker

покажи процедуры

сотрипроцедуры

- 53 -

-----------------------------------------------------------------

пс без аргументов

pe команда

Сокращение от "Перо Стирающее". Переводит перо всех активных че-

репашек в стирающий режим. В этом режиме черепашка стирает кон-

тур, по которому движется.

Пример:

иг \*исходная графика\*

повтори 120 [вперед 1 направо 3]

пс

повтори 120 [вп 1 пр 3]

-----------------------------------------------------------------

пусто? <слово/список> 1 аргумент

empty? логический датчик

Возвращает значение "истина", если аргумент представляет собой

пустое слово или пустой список, и "ложь" - в противном случае.

Примеры:

ст \*сотри текст\*

пиши пусто? []

На листе напишет: истина

пиши пусто? "

На листе напишет: истина

пиши пусто? [а]

На листе напишет: ложь

-----------------------------------------------------------------

пусть <имя> <слово/список> 2 аргумента

make команда

Переменной <имя> присваивается значение <слово/список>. Если

перменной с таким именем ранее не существовало, то команда соз-

дает новую переменную. Если имя переменной было указано в заго-

ловке работающей процедуры (см. примитив "это"), то переменная

является локальной, то есть после завершения работы процедуры в

переменной будет восстановлено значение, имевшееся в ней до на-

чала работы процедуры. Все остальные переменные являются гло-

бальными, то есть значение таких переменных можно использовать и

изменять в любом месте программы, в котором отсутствует локаль-

ная переменная с тем же именем.

Аналогом данной команды является команда "назови", последняя,

однако, требует противоположного порядка следования аргументов.

Значения переменных в памяти не сохраняются после окончания се-

анса работы системы.

Для обращения к значению переменной используется следующая

запись: ":<имя переменной>". Кроме того, значение переменной мо-

жно получить с помощью датчика "значение".

См. назови, это, сотриимя, сотриимена, значение.

Примеры:

пусть "Коля "друг

покажи значение "Коля

В поле команд напишет: друг

пусть "дети [Коля Толя Петя Маша]

покажи :дети

В поле команд напишет: [Коля Толя Петя Маша]

- 54 -

-----------------------------------------------------------------

пч без аргументов

st команда

Сокращение от "Покажи Черепашку". Делает всех активных черепашек

видимыми.

См. сч (спрячь черепашку).

Пример:

иг \*исходная графика\*

для [1 2 3]

пч

сч \*спрячь черепашку\*

-----------------------------------------------------------------

равны? <слово/список1> <слово/список2> 2 аргумента

equal? логический датчик

Возвращает значение "истина", если <слово/список1> и <слово/спи-

сок2> равны. Проверка равенства не различает прописных и строч-

ных латинских букв, но различает прописные и строчные русские

буквы. Слово и список, содержащий ровно это слово, не равны.

Замечание: русская и латинская буквы, имеющие сходные начерта-

ния, для компьютера суть разные буквы.

См. одинаковы?, = .

Примеры:

ст \*сотри текст\*

пиши равны? "а "А

На листе напишет: ложь

пиши равны? "один [один]

На листе напишет: ложь

пиши равны? "f "F

На листе напишет: истина

- 55 -

-----------------------------------------------------------------

разбери <слово> 1 аргумент

parse датчик

Преобразует строку, возвращаемую датчиками "выделенное" или

"буфер", в список и возвращает последний. Элементами этого

списка являются слова, составляющие преобразуемую строку. Воз-

вращаемый список может использоваться в качестве аргумента при-

митивов, оперирующих со списками.

Примеры:

ст \*сотри текст\*

пиши [рыба сидела на дереве]

выделяй

квх \*курсор вверх\*

копируй

покажи разбери буфер

В поле команд напишет: [рыба сидела на дереве]

-----------------------------------------------------------------

рев без аргументов

px команда

Сокращение от "РЕВерсивное перо". Переводит перо всех активных

черепашек в реверсивный режим: при движении по чистому полю перо

рисует, по нарисованному - стирает.

Пример:

иг \*исходная графика\*

рев

медленно

повтори 4 [повтори 3

[вперед 40 направо 90] вперед 20 направо 90]]

-----------------------------------------------------------------

ручка <число> 1 аргумент

paddle датчик

Возвращает число, соответствующее положению игрового регулятора

с номером <число>.

Пример:

покажи ручка 0

-----------------------------------------------------------------

сг без аргументов

cg команда

Сокращение от "Сотри Графику". Стирает графическое изображение,

при этом черепашки возвращаются в исходное положение и поворачи-

ваются в направлении 0. "сг" не меняет цвет фона, а также цвет и

состояние пера черепашек. К графическому изображению относятся

все линии и контуры, нарисованные, закрашенные или отштампован-

ные, а также надписи (т.е. тексты, написанные командой "надпись"

или в одноименном режиме).

См. сотри, иг.

Пример:

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

повтори 4 [вперед 30 направо 90]

вп 80 \*вперед 80\*

надпись [у-у-у]

пиши [это текст]

новфон 3

новформа 22

сг

- 56 -

-----------------------------------------------------------------

символ <число> 1 аргумент

char датчик

Возвращает символ с ASCII-кодом, равным <числу>. ASCII - это

американский стандартный код для обмена информацией.

Таблицу ASCII-кодов см. в приложении 6.

См. код.

Примеры:

ст \*сотри текст\*

пиши символ 65

На листе напишет: А

пиши символ 97

На листе напишет: а

-----------------------------------------------------------------

ск без аргументов

cc команда

Сокращение от "Сотри Команды". Стирает весь текст в поле команд.

-----------------------------------------------------------------

скажи <черепашка/список черепашек> <список инструкций>

ask 2 аргумента

команда или датчик

Временно активизирует <черепашку> (или черепашек), выполняя для

нее (или для них) <список инструкций>. Этот примитив не изменяет

список активных черепашек, установленный последней командой

"для". Если <список инструкций> - датчик, то "скажи" возвращает

ту же информацию, что и <список инструкций>.

См. для.

Пример:

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

направо 90

вперед 50

скажи 0 [направо 90 вперед 60]

направо 90

вперед 40

покажи скажи 3 [место]

В поле команд напишет: [10 -40]

- 57 -

-----------------------------------------------------------------

сколько <слово/список> 1 аргумент

count датчик

Возвращает количество элементов в слове или списке. Элементами

слова являются символы (включая цифры), элементами списка могут

быть слова или списки.

Примеры:

ст \*сотри текст\*

пиши сколько "одуванчик

На листе напишет: 9

пиши сколько [хорошо живет на свете]

На листе напишет: 4

пиши сколько []

На листе напишет: 0

-----------------------------------------------------------------

следэкран без аргументов

nextscreen команда

Сокращение от "СЛЕДующий ЭКРАН". Выводит на экран следующую пор-

цию текста, т.е. следующие за видимым текстом 19 строк. Аналогич-

ный эффект дает нажатие клавиши "следующий экран".

См. предэкран.

-----------------------------------------------------------------

слово <слово1> <слово2> 2 аргумента

word датчик

Замечание: датчик "слово" допускает более двух аргументов; в том

случае, когда аргументов три и более, датчик вместе со всеми ар-

гументами заключается в круглые скобки:

(слово <слово1> <слово2> <слово3>...)

"Склеивает" все аргументы в одно слово и возвращает последнее.

Примеры:

ст \*сотри текст\*

пиши (слово "на "при "мер)

На листе напишет: например

-----------------------------------------------------------------

слово? <слово/список> 1 аргумент

word? логический датчик

Возвращает значение "истина", если аргумент является словом, и

"ложь" - в противном случае.

Примеры:

ст \*сотри текст\*

пиши слово? "Пух

На листе напишет: истина

пиши слово? [Пух]

На листе напишет: ложь

если слово? "башня [пиши сколько "башня]

На листе напишет: 5

- 58 -

-----------------------------------------------------------------

случайный <число> 1 аргумент

сл <число> датчик

random

Возвращает случайное неотрицательное целое число, меньшее зна-

чения аргумента.

Примеры:

покажи случайный 20

иг \*исходная графика\*

медленно

повтори 20 [направо 90 вперед случайный 100]

-----------------------------------------------------------------

сотри без аргументов

clean команда

Стирает графическое изображение; при этом не меняется цвет фона,

а также цвет, положение, направление и состояние пера черепашек.

К графическому изображению относятся все линии и контуры, нари-

сованные, закрашенные или отштампованные, а также надписи (т.е.

тексты, напечатанные с помощью команды "надпись" или в одноимен-

ном режиме).

См. иг, сг.

Пример:

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

повтори 3[вперед 40 направо 120]

назад 40

новкурс 90

новфон 3

новцвет 2

сотри

-----------------------------------------------------------------

сотриимена без аргументов

clearnames команда

Стирает имена и значения всех имеющихся в памяти переменных. Пе-

ременные создаются командами "пусть" и "назови".

См. пусть, назови.

Примеры:

назови 8976 "число

покажи :число

В поле команд напишет: 8976

сотриимена

покажи :число

В поле команд напишет: число не присвоено значение

- 59 -

-----------------------------------------------------------------

сотриимя <слово/список> 1 аргумент

clearname команда

Стирает имя и значение (имена и значения) указанной переменной

(переменных). Переменные создаются командами "пусть" и "назови".

См. пусть, назови.

Примеры:

пусть "дети [Миша Гена Люда]

назови 8976 "число

покажи :число

В поле команд напишет: 8976

сотриимя "число

покажи :число

В поле команд напишет: число не присвоено значение

сотриимя [дети число]

-----------------------------------------------------------------

сотриконтроль без аргументов

clearevents команда

Стирает все назначения, сделанные с помощью команды "контроль".

См. контроль.

Пример:

контроль "z [пишикоманды "сотриконтроль]

Нажмите "Ctrl+z"

Нажмите "ввод" (выполняется команда "сотриконтроль")

Нажмите "Ctrl+z"

-----------------------------------------------------------------

сотрилист <имя листа> 1 аргумент

erpage команда

Уничтожает лист <имя листа> на диске. Уничтоженный лист не может

быть восстановлен в LogoWriter'е.

Пример:

сотрилист "picture

-----------------------------------------------------------------

сотрипроцедуры без аргументов

cleartools команда

Стирает все имеющиеся в памяти внешние процедуры. Внешние проце-

дуры загружаются в память командой "возьмипроцедуры". На изнанке

текущего листа нет описания внешних процедур. Внешние процедуры

не уничтожаются при смене листов. Данная команда есть единствен-

ный способ очистить память от внешних процедур.

См. возьмипроцедуры, процедуры.

О внешних процедурах см. также приложение 4.

-----------------------------------------------------------------

сотрифайл <имя файла> 1 аргумент

erasefile команда

Уничтожает файл <имя файла>.

Пример:

сотрифайл "joke

- 60 -

-----------------------------------------------------------------

список <слово/список1> <слово/список2> 2 аргумента

list датчик

Замечание: данный датчик допускает более двух аргументов; в том

случае, когда аргументов три и более, датчик вместе со всеми ар-

гументами заключается в круглые скобки:

(список <слово/список1> <слово/список2> <слово/список3>...)

Возвращает список, элементами которого являются <слово/список1>

и <слово/список2>.

См. ос (объединенный список).

Примеры:

покажи список "Винни "Пух

В поле команд Вы увидите: [Винни Пух]

покажи список [Винни] [Пух]

В поле команд Вы увидите: [[Винни] [Пух]]

-----------------------------------------------------------------

список? <слово/список> 1 аргумент

list? логический датчик

Возвращает значение "истина", если <слово/список> является спис-

ком, и "ложь" - в противном случае.

Примеры:

ст \*сотри текст\*

пиши список? [Винни Пух]

На листе напишет: истина

пиши список? "невидимка

На листе напишет: ложь

если список? [травка зеленеет]

[пиши сколько[травка зеленеет]]

На листе напишет: 2

-----------------------------------------------------------------

спк без аргументов

cursorchar датчик

Сокращение от "Символ Под Курсором". Возвращает символ, на кото-

ром стоит курсор.

Примеры:

ст \*сотри текст\*

иг \*исходная графика\*

сч \*спрячь черепашку\*

пп \*перо подними\*

новместо [-130 60]

вставь [Винни Пух]

повтори 9 [кнд надпись спк жди 30 ]

\*кнд - курсор назад\*

- 61 -

-----------------------------------------------------------------

спч без аргументов

charunder датчик

Сокращение от "Символ Под Черепашкой". Возвращает символ, на

котором находится черепашка. Если под черепашкой чистое поле

или символ, являющийся частью надписи, то "спч" возвращает про-

бел.

Пример:

ст \*сотри текст\*

пиши "Пух

иг \*исходная графика\*

пп \*перо подними\*

кнд \*курсор назад\*

повтори 3 [кнд \*курсор назад\*

новместо мк \*мк - место курсора\*

покажи спч жди 30]

-----------------------------------------------------------------

ст без аргументов

ct команда

Сокращение от "Сотри Текст". Уничтожает всю текстовую информацию

на текущей стороне листа. К текстовой информации не относятся

тексты, напечатанные в режиме "надпись" или с помощью одноимен-

ной команды.

Команда "ст" является одним из способов уничтожения всех проце-

дур на изнанке листа.

Пример:

иг \*исходная графика\*

пиши [это текст]

надпись [это не текст]

ст

-----------------------------------------------------------------

стоп без аргументов

stop команда

Прекращает работу процедуры и возвращает управление в поле ко-

манд (если процедура является подпроцедурой, то управление пере-

дается в головную процедуру). "Стоп" может использоваться только

в описании процедур.

Пример:

Опишите такую процедуру на изнанке листа:

это счет :число

если :число = 0 [пиши :число стоп]

пиши :число

жди 5

счет :число - 1

конец

Перейдите в поле команд и дайте инструкции:

счет 10

счет 20

- 62 -

-----------------------------------------------------------------

сч без аргументов

ht команда

Сокращение от "Спрячь Черепашку". Делает всех активных черепашек

невидимыми.

См. пч (покажи черепашку).

Пример:

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

сч

-----------------------------------------------------------------

таб без аргументов

tab команда

Сокращение от "ТАБлица". Перемещает курсор вперед на ближайший

табличный маркер. Табличные маркеры располагаются через 5 симво-

лов.

Пример:

ст \*сотри текст\*

вставь "Наташа таб

вставь "Ира таб

пиши "Света

вставь 15 таб

вставь 14 таб

пиши 10

-----------------------------------------------------------------

точка без аргументов

setpoint команда

Устанавливает десятичную точку для записи вещественных чисел.

См. запятая.

-----------------------------------------------------------------

у <число> 1 аргумент

sety команда

Для всех активных черепашек выполняется следующее: не изменяя х-

координаты черепашки, LogoWriter перемещает ее в точку, у-коор-

дината которой равна <числу>.

Замечание: в русском варианте примитива "у" может быть буквой

как русского, так и латинского алфавита.

Пример:

иг \*исходная графика\*

у -60

повтори 4 [вперед 20 направо 90]

у -30

повтори 4 [вперед 20 направо 90]

у 0

повтори 4 [вперед 20 направо 90]

- 63 -

-----------------------------------------------------------------

угол <[х у]> 1 аргумент

towards датчик

Возвращает значение направления прямой, соединяющей черепашку с

точкой <[х у]>. Если активными являются несколько черепашек, то

данный датчик применяется только к одной черепашке, а именно, к

той, которая стоит первой в списке, возвращаемом датчиком "кто".

Пример:

иг \*исходная графика\*

покажи угол [100 60]

В поле команд напишет: 59.0372

скажи 2 [покажи угол [0 0]]

В поле команд напишет: 45

-----------------------------------------------------------------

удали без аргументов

delete команда

Удаляет с листа символ, на котором стоит курсор.

Пример:

ст \*сотри текст\*

пиши "Пух

внт \*в начало текста\*

удали

-----------------------------------------------------------------

укоор без аргументов

ycor датчик

Возвращает значение у-координаты черепашки. Если активными явля-

ются несколько черепашек, то данный датчик применяется только к

одной черепашке, а именно, к той, которая стоит первой в списке,

возвращаемом датчиком "кто".

Замечание: в русском варианте примитива "у" может быть буквой

как русского, так и латинского алфавита.

Примеры:

иг \*исходная графика\*

покажи укоор

В поле команд напишет: 0

скажи 2 [покажи укоор]

В поле команд напишет: -40

- 64 -

-----------------------------------------------------------------

установидиск <буква> 1 аргумент

setdisk команда

На компьютере с более чем одним накопителем после выполнения ко-

манды вся информация будет записываться на накопитель и считы-

ваться с накопителя с именем <буква>. Первый накопитель имеет

имя "А", второй - "В" или "С", в зависимости от системы.

Можно скопировать лист с накопителя "А" на накопитель "В", дав

следующие инструкции:

установидиск "а

возьмилист "<имя листа>

установидиск "b

оглавление

-----------------------------------------------------------------

уходи без аргументов

leavepage команда

Переходит к оглавлению без сохранения текущего листа.

-----------------------------------------------------------------

файлы без аргументов

filelist датчик

Возвращает список файлов, имеющихся в текущем каталоге.

Пример:

покажи файлы

-----------------------------------------------------------------

фон без аргументов

bg датчик

Возвращает число, соответствующее цвету фона текущего листа.

Цвет фона листа есть часть графической информации, которая запи-

сывается на диск.

См. новфон.

Таблицу цветов см. в приложении 5.

Пример:

иг \*исходная графика\*

покажи фон

В поле команд напишет: 0

новфон 3

покажи фон

В поле команд напишет: 3

-----------------------------------------------------------------

форма без аргументов

форму

shape датчик

Возвращает номер формы активной черепашки. Если активными явля-

ются несколько черепашек, то данный датчик применяется только к

одной черепашке, а именно, к той, которая стоит первой в списке,

возвращаемом датчиком "кто".

См. новформа.

Примеры:

иг \*исходная графика\*

новформа 11

покажи форму

новформа форма + 1

покажи форму

В поле команд напишет: 12

- 65 -

-----------------------------------------------------------------

формы без аргументов

shapes команда

Сохраняет текущий лист и переходит к листу форм. Лист форм -

это специальный лист, на котором хранятся и могут редактировать-

ся и создаваться изображения (формы) черепашки.

-----------------------------------------------------------------

х <число> 1 аргумент

setx команда

Для всех активных черепашек выполняется следующее: не изменяя у-

координаты черепашки, LogoWriter перемещает ее в точку, х-коор-

дината которой равна <числу>.

Замечание: в русском варианте примитива "х" может быть буквой

как русского, так и английского алфавита.

Пример:

иг \*исходная графика\*

х -60

повтори 4 [вперед 20 направо 90]

х -30

повтори 4 [вперед 20 направо 90]

х 0

повтори 4 [вперед 20 направо 90]

-----------------------------------------------------------------

хкоор без аргументов

хcor датчик

Возвращает значение х-координаты черепашки. Если активными явля-

ются несколько черепашек, то данный датчик применяется только к

одной черепашке, а именно, к той, которая стоит первой в списке,

возвращаемом датчиком "кто".

Замечание: в русском варианте примитива "х" может быть буквой

как русского, так и английского алфавита.

Примеры:

иг \*исходная графика\*

покажи хкоор

В поле команд напишет: 0

скажи 2 [покажи хкоор]

В поле команд напишет: -40

- 66 -

-----------------------------------------------------------------

цвет без аргументов

color датчик

Возвращает число, соответствующее цвету черепашки (именно этим

цветом черепашка рисует). Если активными являются несколько че-

репашек, то данный датчик применяется только к одной черепашке,

а именно, к той, которая стоит первой в списке, возвращаемом

датчиком "кто".

См. новцвет.

Пример:

иг \*исходная графика\*

новцвет 3

повтори 6 [вперед 30 направо 60]

покажи цвет

В поле команд напишет: 3

-----------------------------------------------------------------

цветполя без аргументов

цп датчик

colorunder

Возвращает число, соответствующее цвету поля, на котором нахо-

дится черепашка (точнее, центр черепашки). Если центр черепашки

приходится на контур, то "цветполя" возвращает число, соответст-

вующее цвету контура.

Примеры:

иг \*исходная графика\*

покажи цп

В поле команд напишет: 0

новцвет 3

вп 50 \*вперед 50\*

покажи цп

В поле команд напишет: 3

пп \*перо подними\*

вперед 10

покажи цп

В поле команд напишет: 0

-----------------------------------------------------------------

целое <число> 1 аргумент

int датчик

Возвращает значение целой части вещественного числа.

Примеры:

покажи целое 123.666

покажи целое угол [55 20]

-----------------------------------------------------------------

цт без аргументов

tc датчик

Сокращение от "Цвет Текста". Возвращает число, соответствующее

цвету текста, установленному последней командой "новцт".

См. новцт (новый цвет текста).

Пример:

ст \*сотри текст\*

пиши [травка зеленеет]

покажи цт

новцт 3 \*новый цвет текста\*

пиши [солнышко блестит]

покажи цт

В поле команд напишет: 3

- 67 -

-----------------------------------------------------------------

число? <слово/список> 1 аргумент

number? логический датчик

Возвращает значение "истина", если аргумент является числом, и

"ложь" - в противном случае. Число есть частный случай слова.

Примеры:

покажи число? 99

В поле команд напишет: истина

покажи число? [99]

В поле команд напишет: ложь

покажи число? 77 / 5

В поле команд напишет: истина

-----------------------------------------------------------------

штамп без аргументов

stamp команда

В текущей позиции черепашка оставляет копию своего изображения.

Черепашка ставит "штампы" в соответствии со своим цветом и сос-

тоянием пера (перо может быть опущенным, поднятым, стирающим и

реверсивным).

Примеры:

иг \*исходная графика\*

штамп

вп 20 \*вперед 20\*

новформа 11

штамп вперед 40

сч \*спрячь черепашку\*

-----------------------------------------------------------------

экран без аргументов

printscreen команда

Подает на печатающее устройство (принтер) всю текстовую и графи-

ческую информацию с текущего листа, видимую на экране. Текст,

находящийся в поле команд и в заголовке листа, на принтер не по-

дается.

См. печатай, печатай80.

Пример:

иг \*исходная графика\*

ст \*сотри текст\*

повтори 4 [вперед 50 направо 90]

пиши [Винни Пух]

назад 20

сч \*спрячь черепашку\*

надпись [дело было вечером]

экран

- 68 -

-----------------------------------------------------------------

элемент <число> <слово/список> 2 аргумента

item датчик

Возвращает элемент <слова/списка> с номером, равным <числу>.

Элементами слова являются символы, элементами списка могут быть

слова или списки.

Примеры:

покажи элемент 3 [хорошо живет на свете]

В поле команд напишет: на

покажи элемент 3 "Пух

В поле команд напишет: х

-----------------------------------------------------------------

это специальное слово

to

С примитива "это" начинается заголовок процедуры. После слова

"это" должно идти имя процедуры и далее, через пробелы, при не-

обходимости, перечисляются формальные параметры процедуры. Каж-

дый параметр имеет вид: :<имя переменной>. Все переменные, име-

на которых упомянуты в заголовке процедуры, являются локальными,

то есть, после завершения работы процедуры в переменных восста-

новятся значения, имевшиеся в них до начала работы процедуры

(если переменные с такими именами существовали). Начальные зна-

чения этих переменных должны быть заданы в момент вызова проце-

дуры.

Пример:

Опишите на изнаночной стороне листа следующую процедуру:

это фигура :углы :сторона

повтори :углы [вперед :сторона направо 360 / :углы]

конец

Перейдите в поле команд. Теперь можно вызвать эту процеруру, на-

пример, с помощью такой инструкции:

фигура 5 30

-----------------------------------------------------------------

arctg <число> 1 аргумент

arctan датчик

Возвращает значение угла (в градусах), тангенс которого равен

<числу>.

Пример:

покажи arctg 67.98

иг \*исходная графика\*

медленно

повтори 10 [новкурс arctg укоор вперед 10]

- 69 -

-----------------------------------------------------------------

cos <число> 1 аргумент

cos датчик

Возвращает значение косинуса угла, содержащего <число> градусов.

Примеры:

покажи cos 45

иг \*исходная графика\*

покажи cos курс

-----------------------------------------------------------------

sin <число> 1 аргумент

sin датчик

Возвращает значение синуса угла, содержащего <число> градусов.

Примеры:

покажи sin курс

иг \*исходная графика\*

новместо [-100 -80]

сотри

повтори 350 [новкурс курс + 1 вперед sin курс]

-----------------------------------------------------------------

startup без аргументов

специальное слово

Startup - это имя специальной процедуры или специального листа

LogoWriter'а. Непосредственно после загрузки листа (из оглавле-

ния или командой "возьмилист"), на изнанке которого есть проце-

дура с именем "startup", система автоматически выполняет эту

процедуру. Если на диске имеется лист с именем "startup", то при

запуске система автоматически загружает в память лист "startup".

Подробности см. в приложении 2.

Пример:

Опишите процедуру "startup" на изнанке нового листа:

это startup

иг \*исходная графика\*

для всех

пч \*покажи черепашку\*

повтори 4 [вперед 40 направо 90]

сч \*спрячь черепашку\*

конец

Перейдите в поле команд и напечатайте инструкции:

ил "test \*имя листа\*

новлист

возьмилист "test

-----------------------------------------------------------------

Часть III. Приложения.

Приложение 1. Тематический список примитивов.

Графика сч

------- у

быстро угол

возьмирис (вр) укоор

вперед (вп) фон

все всех всем форма форму

для х

домой хкоор

доточки цвет

заполни цветполя (цп)

иг штамп

каждая

крась Листы

кто -----

курс беззащиты

медленно возьмилист (вл)

место возьмипроцедуры

мк возьмиформы

надпись восстанови

назад (нд) добавь

налево (лв) защита

направо (пр) ил

новкурс (нк) листы

новместо (нм) лицо?

новфон новлист (нл)

новформа (нф) оглавление

новцвет (нц) переверни

по помнилист

помнирис процедуры

пп сотрилист

пс сотрипроцедуры

пч уходи

рев формы

скажи

сотри

спч

.

Текстовый редактор/ пусто?

слова и списки равны?

------------------- разбери

буфер символ

верни сколько

вксп слово

вкст слово?

вкт список

внсп список?

внст спк

внт таб

возьмитекст удали

вставь цт

входит? число?

выделенное элемент

выделяй

вырежи Контроль исполнения/

длина логика

замени --------------------

ищи выход (вых)

квп делай

квх если

кнд еслииначе

кнз и

код или

копируй не

кпрв останов

кпсл повтори

нашла? стоп

невыделяй

новпоз Печать

новцт (нцт) ------

одинаковы? дваинтервала

ос одининтервал

первый (прв) печатай

пиши (п) печатай80

пишикоманды (пк) экран

поз

покажи Пауза

помнитекст -----

последний (псл) жди

.

Память Контрольные клавиши

------ -------------------

мусор контроль

память сотриконтроль

Специальные слова Экран

----------------- -----

startup ол

истина предэкран

конец сг

ложь ск

это следэкран

ст

Переменные

---------- Ввод данных

значение -----------

имя? ждикоманды

назови ждисимвол

пишиимена ждисписок

покажиимена клавиша?

пусть кнопка?

сотриимена ручка

сотриимя

ДОС

Математика ---

---------- диск

arctg дос

cos копируйфайл

sin переименуй

запятая сотрифайл

кк установидиск

минус файлы

остаток

оч Звук

случайный (сл) ----

точка звук

целое

Приложение 2. Листы "help" и "startup", процедура

"startup".

1. Лист "help".

Вы можете создать и использовать лист с именем "help". Та-

кой лист создается и редактируется точно так же, как и любой

другой обычный лист. Отличается он от остальных листов тем, что

его содержимое доступно для чтения (но не для изменения) при ра-

боте с любым другим листом. Если во время работы с обычным лис-

том нажать клавишу "помощь", то на экране появится содержимое

листа "help". Обычно на этот лист записываются справочные све-

дения, часто используемые учащимися в процессе занятий.

После нажатия клавиши "помощь" Вы не можете редактировать

текст, переворачивать лист и работать с полем команд. Нажатие

клавиши "выход" восстанавливает исходный режим работы.

2. Лист "startup".

Если в каталоге имеется лист с именем "startup", то сеанс

работы с системой несколько отличается от описанного в I.3. Сра-

зу после вызова система загружает в память лист с именем

"startup" и переходит к работе с этим листом. Такой лист может

быть использован для

- ознакомления учащихся с предварительными сведениями в на-

чале урока.

- предварительной загрузки внешних процедур и задания на-

чальных значений переменных.

- загрузки листа, содержащего процедуру с именем "startup",

что позволяет начать автоматическое выполнение некоторой прог-

раммы.

3. Процедура "startup".

Любой лист может содержать процедуру с именем "startup".

Если на изнаночной стороне листа имеется такая процедура, то

после загрузки листа в память (после выбора листа в оглавлении

или после выполнения команды "возьмилист"),система автоматически

переходит к выполнению данной процедуры. Процедура "startup" не

выполняется автоматически после команд "добавь" и "возьмипроце-

дуры".

Процедура "startup" может быть использована для задания на-

чальных значений переменных, загрузки внешних процедур, предва-

рительного программирования клавиш и так далее.

Приложение 3. Список служебных сообщений LogoWriter'а.

В % не хватает входных данных.

Примитив % требует большего числа аргументов (см. описание

примитивов в части II).

В $ не хватает элементов.

$ содержит меньше элементов, чем требуется по смыслу прог-

раммы. Например, запрашивается n-ый элемент списка, в то время

как список содержит n-1 элементов.

В $ не хватает [

Удостоверьтесь, что в процедуре $ открывающих скобок столь-

ко же, сколько и закрывающих. Сообщение выдается, когда

LogoWriter загружает процедуры в рабочую область памяти (при пе-

реходе с одной стороны листа на другую, загрузке листа, выполне-

нии команды "добавь").

В $ не хватает ]

Удостоверьтесь, что в процедуре $ открывающих скобок столь-

ко же, сколько и закрывающих. Сообщение выдается, когда

LogoWriter загружает процедуры в рабочую область памяти (при пе-

реходе с одной стороны листа на другую, загрузке листа, выполне-

нии команды "добавь").

Диск защищен от записи.

LogoWriter не может записать лист на диск (выполняя команду

"помнилист", "помнирис", "помнитекст" или в результате нажатия

клавиши "выход"), если диск защищен от записи.

Диск переполнен.

На диске, который Вы используете, нет свободного пространс-

тва. Если Вам не удалось записать на него текущий лист, восполь-

зуйтесь другим диском.

ДОС не допускает эту операцию.

Имя листа должно содержать не более 8 символов.

Лист или файл защищен от записи.

Сообщение выдается при попытке записать на диск новую вер-

сию защищенного листа командой "помнилист", при попытке стереть

защищенный лист (или файл) командой "сотрилист" ("сотрифайл")

или при попытке применения команды "защита" к уже защищенному

листу.

Лист с таким именем защищен от записи.

Назовите, пожалуйста, этот лист.

LogoWriter не может записать текущий лист на диск, если

- 2 -

лист не имеет имени. Если Вы хотите сохранить текущий лист, на-

зовите его командой "ил". Перейти в оглавление без сохранения

текущего листа можно командой "уходи".

На этом листе больше нет места для текста.

Попытка применить команду "верни", "добавь" и другие коман-

ды, связанные с печатью текста на листе, привела к тому, что об-

щий объем текста на обеих сторонах листа превысил максимально

допустимое число символов (4096 символов).

Неверный тип файла.

Сообщение выдается при попытке загрузить в качестве листа

(из оглавления или командой "возьмилист") файл, не являющийся

листом LogoWriter'а.

Не знаю, как выполнить #.

LogoWriter не может выполнить слово, не являющееся именем

примитива, процедуры или внешней процедуры.

Не знаю, что делать с @.

Инструкция, предложенная LogoWritr'у для выполнения, начи-

нается с датчика, тогда как всякая инструкция должна начинаться

с команды.

Не могу загрузить лист из-за чрезмерного его объема.

Малый объем (свободной) памяти машины не позволяет загру-

зить лист слишком большого объема.

Не могу закрасить или заполнить.

LogoWriter не может закрасить или заполнить слишком сложную

фигуру, т.е. имеющую слишком много сторон.

Не могу использовать "это" как команду.

Специальное слово "это" было ошибочно использовано либо в

поле команд, либо внутри процедуры. Проверьте, стоит ли слово

"конец" в конце предыдущей процедуры. Слово "это" должно указы-

вать на начало процедуры.

Не могу использовать $ в имени листа.

Некоторые символы (в том числе !, @, #, $, ^ и все буквы

русского алфавита) не могут быть использованы в имени листа.

Не могу разделить на ноль.

Число нельзя разделить на ноль. Проверьте делитель. Это мо-

жет быть слишком малое число, которое, с точностью до представ-

ления чисел в LogoWriter'е, равняется нулю. Это может быть пере-

менная, которой присвоено значение 0.

Неполадки с дисководом или диском.

Диск защищен от записи, или в дисководе вообще нет диска,

или дисковод не закрыт и т.п.

- 3 -

Неполадки с печатающим устройством.

К компьютеру не подключен принтер, или выключен режим "on

line", или на принтере нет бумаги, или бумага застряла и т.п.

Не присоединен джойстик или ручка.

Нет листа с именем $.

В качестве аргумента примитивов "возьмилист", "добавь",

"возьмипроцедуры" использовалось слово, не являющееся именем

листа LogoWriter'а. Список листов можно найти в оглавлении или

получить с помощью датчика "листы".

Нет свободной памяти.

Свободная память и свободное место на листе - это не одно

и то же, т.к. переменные и внешние процедуры не занимают места

на листе, но занимают место в памяти. Часть памяти можно освобо-

дить командами "сотриимена" (стирает из памяти все перемен-

ные) и "сотрипроцедуры" (стирает из памяти все внешние процеду-

ры).

Нет файла с именем $.

В качестве аргумента примитивов "возьмирис", "возьмитекст",

"копируйфайл" , "переименуй", "сотрифайл" использовано имя, ко-

торое не является именем какого-либо файла из имеющихся в теку-

щем каталоге.

Не хватает пространства для загрузки.

Имела место попытка загрузить слишком большой файл командой

"возьмитекст" или "добавь".

Прервана!!!

Была нажата клавиша "останов".

Слишком большое число.

Максимальное число, которое может использовать данная вер-

сия LogoWriter'а - 999999999999.

Слишком длинная строка.

Слишком много внутри ()

Слишком большая глубина вложения скобок. Попробуйте упрос-

тить инструкцию.

Уже есть лист с именем $.

На одном диске не может быть двух листов с одинаковыми име-

нами.

Уже есть файл с именем $.

В качестве аргумента примитивов "помнирис", "помнитекст"

использовалось имя, которое является именем некоторого файла,

- 4 -

находящегося в текущем каталоге.

) не на месте.

Неправильно расставлены скобки (открывающих скобок должно

быть столько же, сколько и закрывающих).

% не допускает $ на входе.

Примитивы LogoWriter'а требуют входных данных определенного

типа. Типы допустимых аргументов см. в описании примитивов

(часть II).

% может использоваться только в процедуре.

Команда "стоп" или "выход" давалась из поля команд, тогда

как обе эти команды могут использоваться только в процедурах.

$ не передает значение в %.

Примитив или процедура % требует некоторого аргумента, при

этом следующий за % примитив или процедура $ не передает значе-

ние в %.

$ не присвоено значение.

LogoWriter не может производить действия над переменной,

если предварительно ей не было присвоено значение. Переменной

можно присвоить значение, либо указав ее в названии процедуры,

либо с помощью команд "пусть" и "назови".

$ ни истинно, ни ложно.

$ используется на входе примитива, требующего в качестве

аргумента слово "истина" или "ложь".

$ является именем примитива.

Используется процедура с именем, являющимся именем примити-

ва LogoWriter'а. Имена примитивов не могут использоваться в ка-

честве имен Ваших процедур.

Примечание: в реальных сообщениях LogoWriter'а служебные

символы @, $, % заменяются соответствующими именами процедур,

переменных и т.д.

Приложение 4. Внешние процедуры.

Система LogoWriter позволяет использовать внешние процеду-

ры. Такие процедуры очень напоминают примитивы языка LogoWriter:

текст внешних процедур недоступен пользователю, они остаются в

памяти при смене листов.

Укажем ряд отличий внешних процедур от обычных:

1. Внешние процедуры сохраняются в памяти при смене листов.

2. Текст внешних процедур нельзя изменить.

3. Внешние процедуры можно заменить или удалить во время

работы с листом, использовав команды "возьмипроцедуры" и "сот-

рипроцедуры".

4. Внешние процедуры не занимают места на листе, хотя и за-

нимают место в памяти компьютера. Поэтому использование большого

числа внешних процедур может привести к нехватке памяти.

Внешними процедурами могут быть процедуры, записанные на

любом листе. Чтобы использовать процедуры, записанные на ка-

ком-либо листе в качестве внешних процедур, следует выполнить

команду "возьмипроцедуры".

Приведем пример автоматической загрузки внешних процедур.

Пусть на изнаночной стороне листа "startup" написан следующий

текст

это startup

возьмипроцедуры "math

оглавление

конец

В этом случае система, перед работой с оглавлением загрузит про-

цедуры листа MATH в качестве внешних процедур.

Если на текущем листе имеется процедура с тем же именем,

что и некоторая внешняя процедура, то такая внешняя процедура

игнорируется, то есть вызов процедуры с данным именем приведет к

выполнению процедуры, записанной на текущем листе.

Приложение 5. Таблица цветов, таблица частот по октавам.

5.1 Таблица цветов.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Число Цвет изображения

-------------------------------------

0 черный

1 белый

2 голубой

3 розовый

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Число Цвет фона

-------------------------------------

0 черный

1 белый

2 голубой

3 фиолетовый

4 красный

5 синий

6 красный

7 зеленый

8 черный

9 светло-серый

10 светло-голубой

11 розовый

12 светло-красный

13 светло-синий

14 желтый

15 светло-зеленый

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.2 Таблица частот по октавам.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нота Частота по октавам

----------------------------------------------------------

большая малая первая вторая

B 123 247 494 988

A# 117 233 466 932

A 110 220 440 880

G# 104 208 415 830

G 98 196 392 784

F# 92 185 370 740

F 87 175 349 698

E 82 165 330 659

D# 78 156 311 622

D 73 147 294 587

C# 69 139 277 554

C 65 131 262 523

----------------------------------------------------------

Приложение 8. Англо-русский словарь примитивов.

all все, всех, всем

and и

arctan arctg

ascii код

ask скажи

back, bk назад, нд

bg фон

bottom вкт

butfirst, bf кпрв

butlast, bl кпсл

button? кнопка?

cb кнд

cc ск

cd кнз

cf квп

cg сг

char символ

charunder спч

clean сотри

clearevents сотриконтроль

clearname сотриимя

clearnames сотриимена

cleartools сотрипроцедуры

clipboard буфер

color цвет

colorunder цветполя, цп

contents оглавление

copy копируй

copyfile копируйфайл

cos cos

count сколько

cp ол

ct ст

cu квх

cursorchar спк

cursorpos мк

cut вырежи

delete удали

disk диск

distance доточки

dos дос

dspace дваинтервала

each каждая

empty? пусто?

end конец

eol вкст

equal? равны?

erasefile сотрифайл

erpage сотрилист

false ложь

fastturtle быстро

- 2 -

filelist файлы

fill крась

first первый, прв

flip переверни

forward, fd вперед, вп

found? нашла?

fput внсп

front? лицо?

getpage, gp возьмилист, вл

getshapes возьмиформы

gettools возьмипроцедуры

heading курс

home домой

ht сч

identical? одинаковы?

if если

ifelse еслииначе

insert вставь

int целое

item элемент

key? клавиша?

lable надпись

last последний, псл

leavepage уходи

left, lt налево, лв

list список

list? список?

load добавь

loadpic возьмирис, вр

loadtext возьмитекст

lock защита

lput вксп

make пусть

member? входит?

minus минус

name назови

name? имя?

namepage, np ил

newpage новлист, нл

nextscreen следэкран

not не

number? число?

or или

output, op выход, вых

paddle ручка

pagelist листы

parse разбери

paste верни

pd по

pe пс

pos место

- 3 -

prescreen предэкран

print, pr пиши, п

printnames пишиимена

printscreen экран

printtext печатай

printtext80 печатай80

pu пп

px рев

random случайный, сл

readchar ждисимвол

readlist ждисписок

readlistcc ждикоманды

recycle мусор

remainder остаток

rename переименуй

repeat повтори

replace замени

restore восстанови

rg иг

right, rt направо, пр

round оч

run делай

savepage помнилист

savepic помнирис

savetext помнитекст

search ищи

select выделяй

selected выделенное

sentence, se ос

setbg новфон

setc новцвет, нц

setcomma запятая

setdisk установидиск

seth новкурс, нк

setpoint точка

setpos новместо, нм

setsh новформа, нф

settc новцт, нцт

settextpos новпоз

setx x

sety y

shade заполни

shape форма, форму

shapes формы

show покажи

shownames покажиимена

sin sin

slowturtle медленно

sol внст

space память

sqrt кк

- 4 -

sspace одининтервал

st пч

stamp штамп

startup startup

stop стоп

stopall останов

tab таб

tc цт

tell для

textlen длина, длину

textpos поз

thing значение

to это

tone звук

toollist процедуры

top внт

towards угол

true истина

type пишикоманды, пк

unlock беззащиты

unselect невыделяй

wait жди

when контроль

who кто

word слово

word? слово?

xcor хкоор

ycor укоор