

Программа.

Программа — алгоритм записанный на языке исполнителя. В нашем случае, программа записывается на псевдокоде.

Программа состоит из: заголовка программы, секции импорта, секции описания процедур и тела программы. В любой части может быть блок комментариев.

Секция описания процедур может быть пустой, если нет вспомогательных алгоритмов.

Пример:

МОДУЛЬ задание;
ПОДКЛЮЧИТЬ Ч := i21eduЧерепашка;
ПРОЦЕДУРА Вперед; НАЧАЛО Ч.Идти(100); Ч.ПоднятьПеро; Ч.Идти(50); КОНЕЦ Вперед;
НАЧАЛО Ч.Очистить; Вперед; КОНЕЦ задание.

Заголовок программы

Сообщает компьютеру, что тут начинается описание программы, у нас заголовок должен находиться в начале файла (перед ним могут быть только комментарии в (* *)).

Заголовок выглядит так:

МОДУЛЬ Имя;

МОДУЛЬ — служебное слово, сообщает, что это заголовок программы. Пишется большими буквами! Отступ от края — 0.

Имя — имя программы, подчиняется правилу имен.

МОДУЛЬ и *Имя* обязательно отделяются друг от друга пробелом.

Заканчивается заголовок программы точкой с запятой «;».

Пример:

МОДУЛЬ задача1;

ПРОВЕРКА Заголовка программы:

-блок должен быть в начале программы.

-отступ у блока 0.

-проверить правильность написание служебного слова: все буквы есть, лишних символов нет, все буквы большие.

-проверить правильность имени программы по правилу имен.

-проверить наличие разделительного пробела между служебным словом и именем.

-проверить наличие «;» в конце заголовка (после имени).

Тело программы.

Тело программы — блок, в котором записываются команды. Выполнение программы начинается с первой команды этого блока. Тело программы стоит в конце всей программы, после секции описания процедур. Тело программы заключается в операторные скобки.

Тело программы выглядит так:

НАЧАЛО

ПерваяКоманда;

КОНЕЦ ИмяПрограммы.

НАЧАЛО и *КОНЕЦ* — служебные слова, тут являются операторными скобками. Отделяют команды программы от других блоков. Эти служебные слова имеют отступ 0. После *КОНЕЦ* через пробел следует повтор *ИмениПрограммы*, и вся программа завершается точкой «.».

Команды отделяются друг от друга точкой с запятой «;». Все команды в блоке имеют отступ 1.

Пример:

НАЧАЛО

Ч.Очистить;

Ч.Идти(100);

КОНЕЦ задание.

ПРОВЕРКА Тела программы:

-блок должен быть описан в конце программы.

-блок должен содержать операторные скобки НАЧАЛО и КОНЕЦ

-операторные скобки имеют отступ 0

-проверить правильность записи служебных слов

нет пропущенных букв;

все буквы большие;

нет лишних символов и пробелов.

-после слова КОНЕЦ, через пробел следует повтор имени программы (сравнить с именем в заголовке программы), и заканчивается все точкой «.»

-проверить имена всех команд (проверка команд):

в записи команд не допущены ошибки:

нужный размер букв,

нет пропусков и лишних символов и пробелов.

-все команды отделяются (заканчиваются) «;»

-все команды имеют отступ 1.

Вспомогательный алгоритм (процедура).

Часто алгоритм хочется использовать несколько раз, для этого его выделяют в процедуру, тогда его можно будет вызвать по **имени**. Можно было бы просто скопировать текст процедуры в нужное место, но если окажется, что алгоритм с ошибкой, то потом придется ходить и править ее везде, где мы вставили текст алгоритма, а если оформить в виде процедуры, то можно будет исправить ее только в самой процедуры и она будет вызываться по **имени** уже везде правильно. Так же модифицировать программу с процедурами гораздо легче.

Процедура состоит из: заголовка процедуры и тела процедуры.

Заголовок может включать в себя описание параметров процедуры.

Пример:

ПРОЦЕДУРА Вперед;
НАЧАЛО
Ч.Идти(100);
Ч.ПоднятьПеро;
Ч.Идти(50);
КОНЕЦ Вперед;

Параметры процедуры.

Часто алгоритмы могут быть почти одинаковыми. Так, процедура рисования квадрата со стороной 50 будет отличаться от процедуры квадрата со стороной 100 только вызовами команды Ч.Идти(50) и Ч.Идти(100). В таких случаях вводят параметры, чтобы можно было записать одну процедуру, а не много почти одинаковых. Параметры — величины, конкретные значения которых подставляются при вызове команд. Пример: П.НаСтарт(10,10). У команды были параметры X, Y, при вызове мы подставили конкретные значения 10,10.

Заголовок процедуры.

Сообщает компьютеру, что тут начинается описание вспомогательного алгоритма - процедуры.

Заголовок процедуры выглядит так:

ПРОЦЕДУРА Имя;

ПРОЦЕДУРА — служебное слово, сообщает, что это вспомогательный алгоритм. Пишется большими буквами! Отступ от края — 1.

Имя — имя вспомогательного алгоритма, подчиняется правилу имен. У каждой процедуры должно быть своё уникальное имя.

ПРОЦЕДУРА и *Имя* обязательно отделяются друг от друга пробелом.

Заканчивается заголовок вспомогательного алгоритма точкой с запятой «;».

В скобках между именем и точкой с запятой может следовать описание параметров процедуры.

Пример:

 ПРОЦЕДУРА Цветок;

ПРОВЕРКА Заголовка процедуры:

-блок должен быть в начале процедуры в секции описания процедур.

-отступ у блока 1.

-проверить правильность написание ключевого слова:

- все буквы есть
- лишних символов нет
- все буквы большие.

-проверить правильность имени программы по правилу имен.

-проверить наличие разделительного пробела между служебным словом и именем.

-проверить наличие «;» в конце заголовка (после имени или, если есть параметры, после закрывающей скобки).

-проверить секцию описания переменных.

Тело процедуры.

Тело процедуры — блок, в котором записываются **команды**. При вызове процедуры начинают выполняться **команды** записанные в теле процедуры. Тело процедуры стоит после заголовка процедуры. Тело процедуры заключается в операторные скобки.

Тело процедуры выглядит так:

НАЧАЛО

ПерваяКоамада;

КОНЕЦ ИмяПроцедуры.

НАЧАЛО и *КОНЕЦ* — служебные слова, тут являются операторными скобками. Отделяют команды процедуры от других блоков. Эти служебные слова имеют отступ 1.

После *КОНЕЦ* через пробел следует повтор имени процедуры, и вся процедура завешается точкой с запятой «;».

Команды отделяются друг от друга точкой с запятой «;». Все **команды** в блоке имеют отступ 2.

Пример:

 НАЧАЛО

 Ч.Идти(100);

 Ч.ПоднятьПеро;

 Ч.Идти(100);

 КОНЕЦ Черта;

ПРОВЕРКА Тела процедур:

-блок должен быть описан сразу после заголовка процедуры.

-блок должен содержать операторные скобки НАЧАЛО и КОНЕЦ, операторные скобки имеют отступ 1,

-проверить правильность записи служебных слов(нет пропущенных букв, все буквы большие, нет лишних символов и пробелов).

-после слова КОНЕЦ, через пробел следует повтор имени программы (сравнить с именем в заголовке программы), и заканчивается все точкой «.»

-проверить имена всех команд (проверка команд): в записи команд не допущены ошибки (нужный размер букв, нет пропусков и лишних символов и пробелов).

-все команды отделяются (заканчиваются) «;»

-все **команды** имеют отступ 2.

Описание параметров процедуры.

В этой секции сообщаются, какие параметры будут у процедуры (какого они будут типа, и какие будут иметь имена). Так выглядит описание параметров (в скобках):

ПРОЦЕДУРА Имя (имяПараметра1, имяПараметра : ТИП1 ; имяПараметра3, имяПараметра4 : ТИП2)

Параметры записываются в скобках после имени процедуры.

У каждого параметра есть **имя** и **тип**.

У нас пока есть один тип параметра «ЦЕЛОЕ, ЦЕЛЫЕ, ЦЕЛ» - целое число.

Сначала, пишется список имен параметров разделяющихся запятой «,», потом следует записать тип параметра.

Двоеточие разделяет список имен параметров от указания типа параметров.

Если есть параметры других типов, то секция повторяется. Разделителем секций выступает «;».

Пример:

 ПРОЦЕДУРА Прямоугольник (сторона1, сторона2 : ЦЕЛЫЕ);

 ПРОЦЕДУРА Квадрат (сторона : ЦЕЛОЕ);

 ПРОЦЕДУРА Крест (сторона : ЦЕЛ ; перо : ЛОГИЧ);

ПРОВЕРКА Описания параметров процедур:

-блок должен быть описан сразу после имени процедуры, в скобках.

-проверить правильность написания типов параметров:

- все буквы есть
- лишних символов нет
- все буквы большие.

-проверить правильность имени параметров по правилу имен, так же все имена различны.

-проверить наличие разделителей:

 «;» - разделяет список имен параметров одного типа;

 «:» - отделяет список имен от типа параметров;

 «;» - разделяет несколько списков параметров различных типов.

-проверить наличие «;» в конце заголовка, после скобок.

Секция импорта.

Секция импорта служит для указания списка исполнителей, к которым мы будем обращаться в программе.

Секция импорта выглядит так:

ПОДКЛЮЧИТЬ сокрИмя1 := имяИсполнителя1, сокрИмя2 := имяИсполнителя2;

ПОДКЛЮЧИТЬ — служебное слово, сообщающее, что начинается секция импорта. За служебным словом, через пробел следует список исполнителей.

Исполнитель описывается парой имен: сокращенным именем (используется для удобства, краткая запись, «ник») и собственным длинным именем.

СокрИмя1 — сокращенное имя исполнителя 1.

ИмяИсполнителя1 — полное имя исполнителя 1.

Разделяются сокращенное и собственное имя парой символов «:=» - пишутся слитно.

Заканчивается секция импорта символом точка с запятой «;»

Секция импорта должна иметь отступ равный 1 от края.

У нас должен быть описан хотя бы один исполнитель.

Примеры:

ПОДКЛЮЧИТЬ Ч := i21eduЧерепашка, Ввод := i21eduВвод, Вывод := i21eduВывод;

ПОДКЛЮЧИТЬ П := i21eduParketAnimRu, Ввод := i21eduВвод, Вывод := i21eduВывод;

ПОДКЛЮЧИТЬ Ч := i21eduЧерепашка;

ПРОВЕРКА Секции импорта:

-блок должен быть описан сразу после заголовка программы (в следующей строке).

-проверить правильность написания ключевого слова:

все буквы есть

лишних символов нет

все буквы большие.

-проверить правильность имен исполнителей по правилу имен.

-проверить наличие разделителей:

« » - пробел должен отделять служебное слово от первого имени исполнителя.

- разделяет список имен исполнителей;

«:=» - отделяет сокращенное имя исполнителя от полного имени;

-проверить наличие «;» в конце секции.

Блок комментариев.

Комментарии — блок текста, который не обрабатывается компьютером. Текст в комментариях служит для различных пояснений человеку (что делает программа, как работает процедура, за что отвечают параметры). Блок комментариев должен быть заключен в операторные скобки.

Блок комментариев выглядит так:

(* тут произвольный текст *)

(* *) - операторные скобки блока комментария.

«(*)» - начало блока комментария.

«*)» - конец блока комментария.

ПРОВЕРКА Блока комментариев

– должны быть верными операторные скобки (* *).

– внутри комментариев не должно быть отдельных закрывающих или открывающих операторных скобок. Нельзя:

– (* текст (* текст *),

– (* текст *) текст *).

Имя

Имя — у всех объектов в программе есть свое имя: у программы есть имя, у исполнителя есть имя, у процедуры есть имя, у параметров есть имя. По этим именам мы можем обращаться к ним.

Правило составления имен.

Имя, может начинаться только с буквы, а дальше содержать буквы или цифры, пробелов в имени быть не должно. Имя не должно совпадать со служебными словами: МОДУЛЬ, ПРОЦЕДУРА и др.

Примеры правильных имен:

Задача, задание10, урок1, Треугольник, сторона, длина, Квадрат.

Регистр букв важен. «Задача» и «задача» - два разных имени!

При вызове процедуры по имени, нужно следить, чтобы имена были в точности одинаковыми. Если процедура имеет имя «Треугольник», то не получится вызвать ее по «триголнк», и даже по «треугольник».

Служебные слова.

Служебные слова нужны для указания компьютеру различных конструкций программы. Все служебные слова пишутся большими буквами не могут быть использованы для имен.

Примеры служебных слов: МОДУЛЬ, ПРОЦЕДУРА, ПОДКЛЮЧИТЬ, НАЧАЛО, КОНЕЦ и другие.

Команды.

Команды — действия, которые мы вызываем. Команды могут содержать параметры. Команды могут быть двух типов: команды исполнителя и процедуры.

Вызов команды исполнителя выглядят:

ИмяИсполнителя.ИмяКоманды;

Вызов команды исполнителя — составной, он состоит из *имени исполнителя* и *имени его команды*, разделенные точкой «.».

Вызов процедуры:

ИмяПроцедуры;

Все команды заканчиваются точкой с запятой «;». Она является разделителем, отделяет команды друг от друга.

Вызов процедуры осуществляется по её имени.

Если у команды есть параметры, они должны быть подставлены в вызов. Это могут быть: конкретные значения, значение параметра (параметр должен быть описан внутри блока, где вызывается), выражение из значений и параметров.

Примеры:

Треугольник;

Квадрат (100);

Квадрат (сторона); (*если вызов происходит в блоке, где есть параметр «сторона»*)

Ч.Идти(100);

Ч.ОпуститьПеро;

ПРОВЕРКА Команд:

-проверить команды и имена параметров блока на правильность:

нет ошибок в записи (для параметров сравнить с секцией описания параметров)

нет пробелов или других символов

все буквы нужного размера.

-проверить, что все команды разделены «;»

-проверить, что у команд требующих параметры:

параметры записаны в скобках

если параметров несколько, то они разделены «,» и их порядок соответствует

порядку в секции описания параметров;

-если команда является командой исполнителя:

проверить правильность имени исполнителя

проверить правильность команды исполнителя

проверить наличие «.» разделяющей имя исполнителя и команды (должна быть

«.» и никаких пробелов)

Операторные скобки.

Операторные скобки служат для отделения одного блока программы от другого. В разных случаях они принимают различный вид. Но всегда есть начало блока и конец, соответственно, есть открывающая скобка и закрывающая.

Как выглядят операторные скобки в различных случаях (иногда скобка может являться концом одного блока и началом другого):

Тело программы/процедуры

НАЧАЛО КОНЕЦ;

Секция импорта:

ПОДКЛЮЧИТЬ ;

Заголовок программы:

МОДУЛЬ ;

заголовок процедуры:

ПРОЦЕДУРА ;

Окончание программы:

КОНЕЦ .

Конец процедуры:

КОНЕЦ ;

Операторные скобки отделяющие блок комментариев.

(* *)

Список, блок параметров:

()

Внутри секций, отделенных служебными словами НАЧАЛО и КОНЕЦ, все команды должны иметь отступ на 1 больший, чем у самих этих слов.