

Часть VI

**КОНСТРУКТОР
АЛГОРИТМОВ
И ФОРМАЛЬНОЕ
ОПИСАНИЕ ЯЗЫКА**

КОНСТРУКТОР АЛГОРИТМОВ (ПОМОЩНИК ЧЕЛОВЕКА)

§1. ЗАЧЕМ НУЖЕН ДРАКОН-КОНСТРУКТОР?

Разумеется, в случае крайней нужды дракон-схему можно нарисовать и вручную. Либо использовать универсальный графический редактор, например, Visio.

Однако это не лучший способ. Гораздо удобнее воспользоваться специальной программой, которая называется «дракон-конструктор».

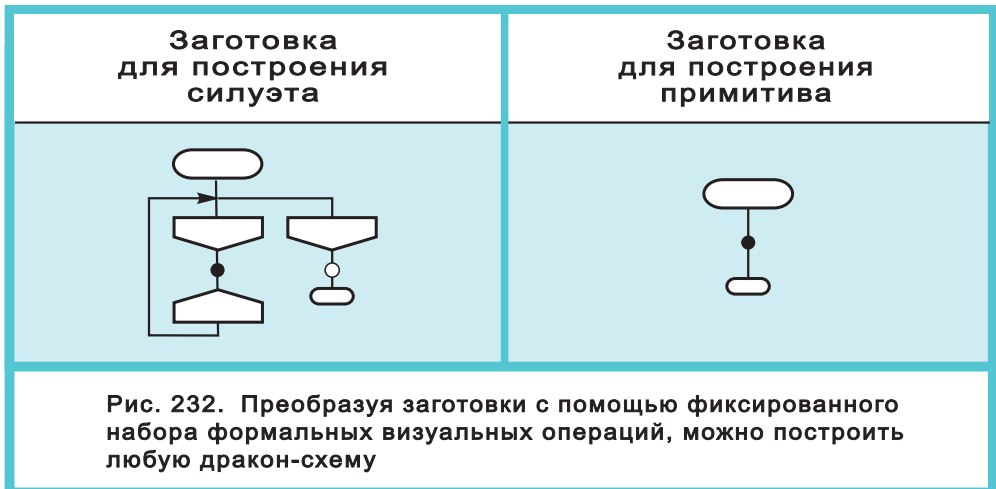
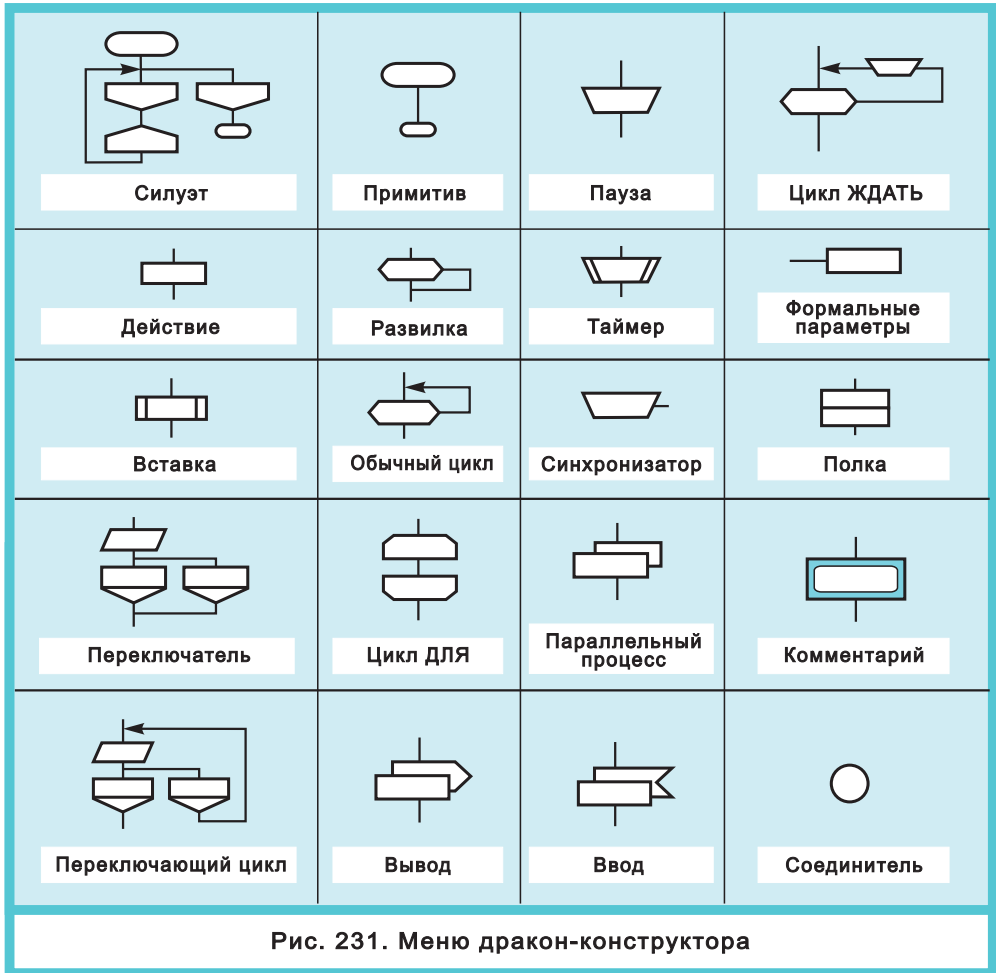
В состав дракон-конструктора входит меню графоэлементов (рис. 231). Чтобы нарисовать дракон-схему, пользователь сначала вызывает меню на экран персонального компьютера. А затем с его помощью рисует или, как говорят, конструирует схему. При этом важную роль играют так называемые заготовки.

§2. ЗАГОТОВКА-СИЛУЭТ И ЗАГОТОВКА-ПРИМИТИВ

Чтобы вырастить огромное дерево, нужно бросить в землю маленькое семечко. Любая сколько угодно сложная дракон-схема тоже вырастает из семечка, которое называется *заготовкой*.

Заготовки бывают двух сортов. Одна используется для построения дракон-схемы «силуэт». Из другой получается примитив (рис. 232).

Построение любой дракон-схемы выполняется за конечное число шагов путем соответствующих преобразований выбранной заготовки.



§3. ЧТО ТАКОЕ АТОМ?

Атом

Это элемент меню на рис. 231, который имеет два вертикальных отростка, лежащие на одной вертикали

Дракон-конструктор может выполнять несколько операций, среди которых важную роль играет команда «ввод атома» (рис. 233). Операция выполняется в два этапа:

- сначала пользователь выбирает нужный атом из меню;
- затем обращается к дракон-схеме и указывает точку, в которую нужно его ввести.

Атомы вставляются не куда попало, а только в разрешенные места, которые называются *валентными точками* дракон-схемы.

Перечень точек включает:

- валентные точки заготовок (отмечены на рис. 232);
- валентные точки макроикон (отмечены на рис. 18);
- входы и выходы атомов.

Ввод атома производится так. Сначала происходит разрыв соединительной линии в выбранной пользователем валентной точке. Затем в место разрыва вставляется атом, как показано на рис. 233.

В реальных дракон-схемах валентные точки не изображаются, а подразумеваются.

§4. ПРИМЕР ПОСТРОЕНИЯ ДРАКОН-СХЕМЫ «ПРИМИТИВ»

Дракон-схема строится на экране компьютера методом «сборки из кубиков». В начале работы пользователь вызывает на экран меню (рис. 231). И размещает его в удобном для себя месте, например в правом верхнем углу экрана. Остальная часть экрана используется как рабочее поле для построения схемы.

Предположим, нужно построить примитив. Пользователь выбирает макроикону «заготовка-примитив» и помещает ее в рабочее поле экрана.

Рассмотрим конкретный пример на рис. 234 и 235.

На первом шаге пользователь вызывает из меню макроикону «обычный цикл». Но куда ее поместить? Пользователь подводит курсор к нужной точке в заготовке-примитив. К той самой точке, в которой следует разорвать соединительную линию, чтобы в образовавшийся разрыв вставить выбранную икону. Результат операции виден на рисунке 234, шаг 1, справа.

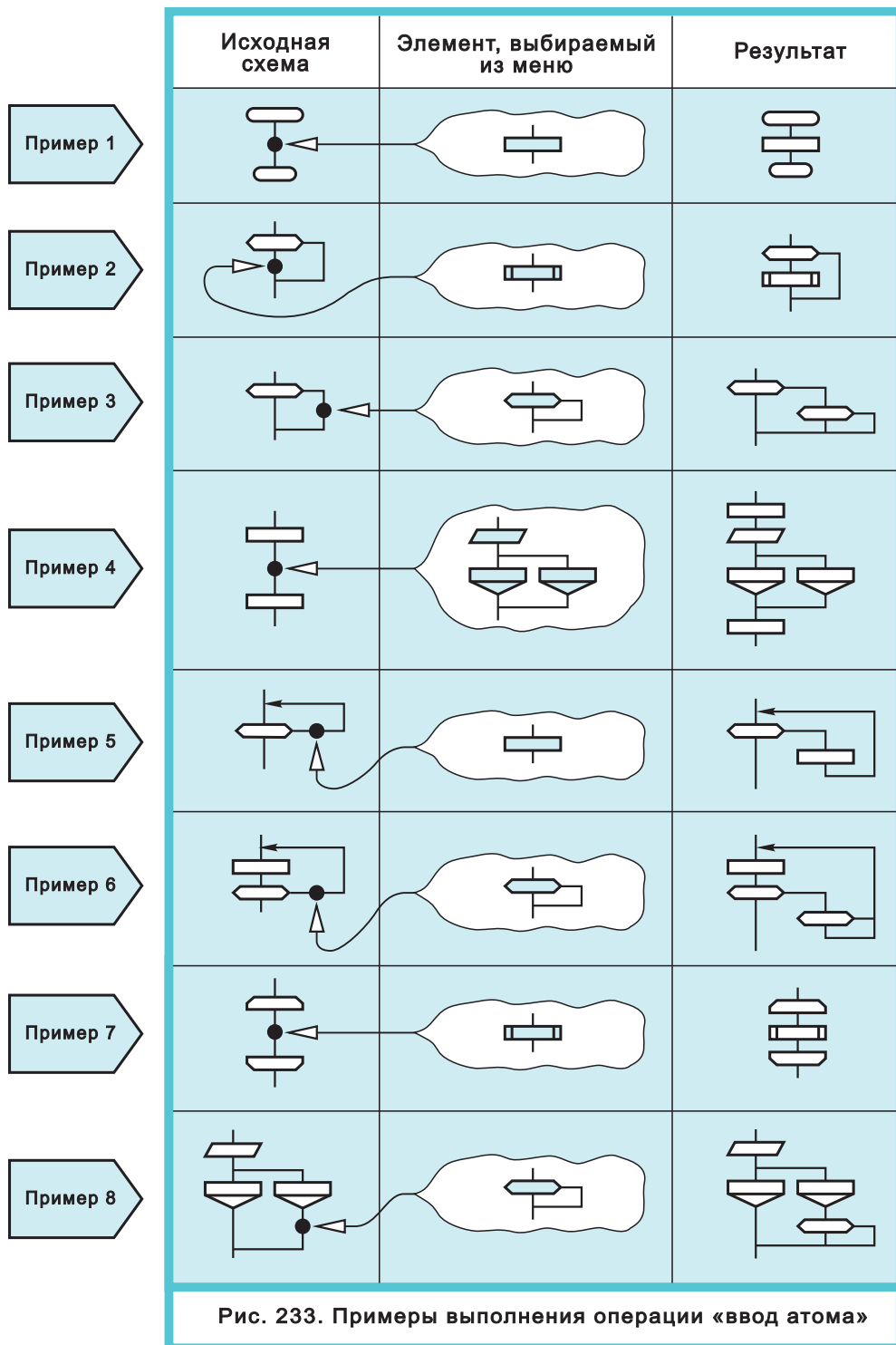


Рис. 233. Примеры выполнения операции «ввод атома»

Два следующих шага выполняются аналогично. В дракон-схему последовательно вводятся две иконы «действие» (рис. 234, шаги 2 и 3).

Далее следует ввод макроиконы «обычный цикл» (рис. 234, шаг 4).

Затем вводятся «вставка», «развилка» и «действие» (рис. 235, шаги 5–7). Результат последней операции показан на рис. 235 (шаг 7, справа).

После того как графический узор (слепыш) дракон-схемы построен, производится заполнение его текстом.

§5. ЧТО ТАКОЕ ЛИАНА?

Обезьяна, сидевшая на дереве, поймала свисавшую сверху лиану. Однако нижняя часть лианы приросла к стволу и не поддавалась. Обезьяна перегрызла ее зубами, уцепилась за конец и мигом перелетела на соседнее дерево, где прочно привязала лиану к ветке.

Нечто подобное умеет делать и дракон-конструктор.

Лиана – это часть дракон-схемы, которая:

- имеет начало и конец (*начало лианы* и *конец лианы*);
- начало лианы находится вверху, конец – внизу;
- от начала к концу проходит хотя бы один маршрут;
- началом лианы служит выход иконы «вопрос» или «вариант» (если этот выход не является петлей цикла);
- концом лианы служит точка слияния.

Лиана есть последовательность шампур-блоков или просто соединительная линия.

§6. ОПЕРАЦИЯ «ПЕРЕСАДКА ЛИАНЫ»

При некоторых условиях (подробно описанных в следующей главе) конец лианы можно оторвать от своего места и присоединить в другую точку дракон-схемы. Такая операция называется *пересадка лианы*. Примеры показаны на рис. 236 и 237.

Выполнение этой операции осуществляется в два этапа. Сначала курсор подводится к лиане, конец которой надо освободить (рис. 236, левая графа).

Куда его присоединить? Пользователь выбирает желаемую точку и отмечает ее курсором (рис. 236, средняя графа). Результат операции «пересадка лианы» показан на том же рисунке в правой графе.

Многие дракон-схемы, представленные в этой книге, построены с помощью пересадки лианы. Укажем некоторые из них: рис. 34, 53, 56, 60, 62, 82–84, 87, 105, 107, 108, 110, 123–125, 159.

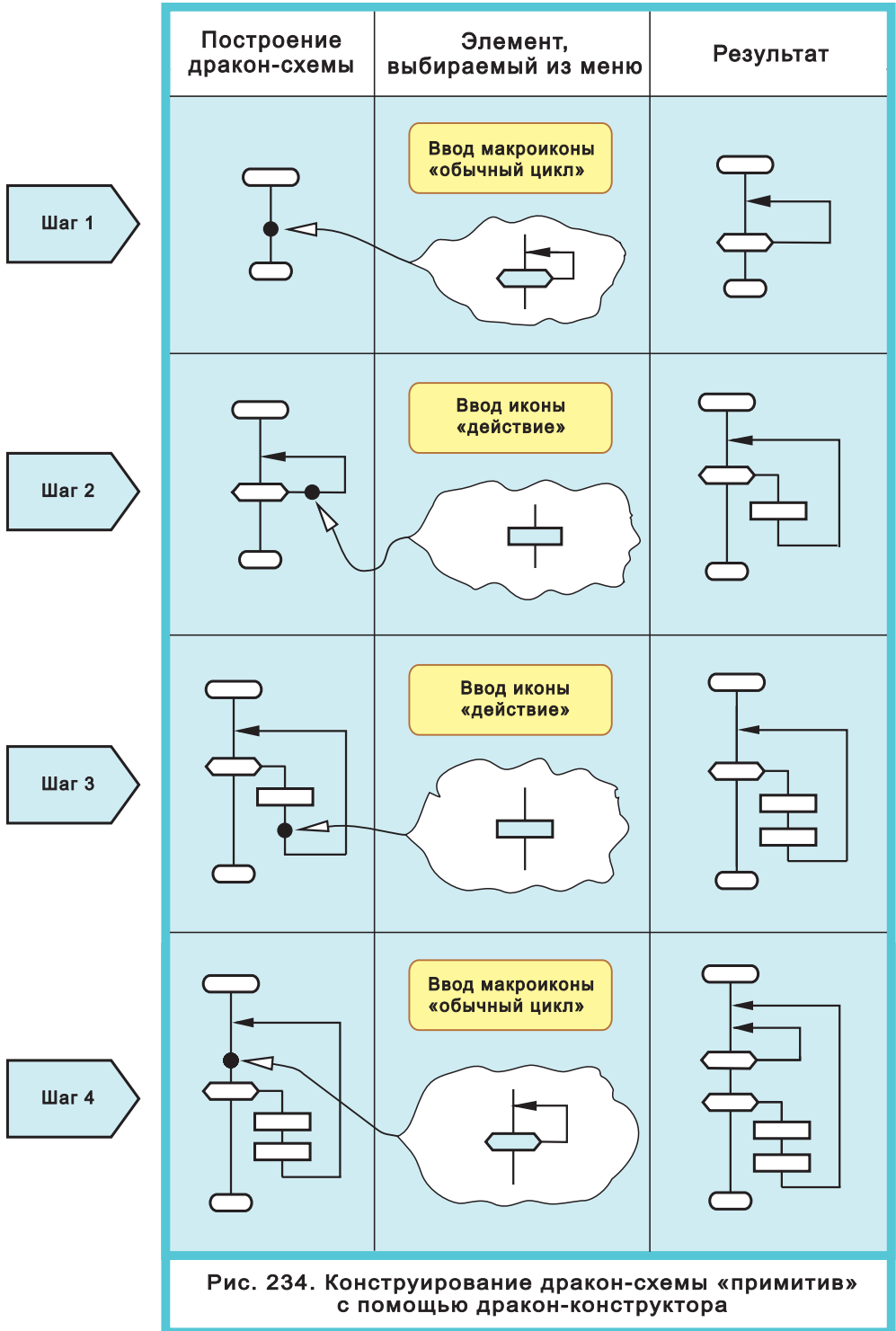
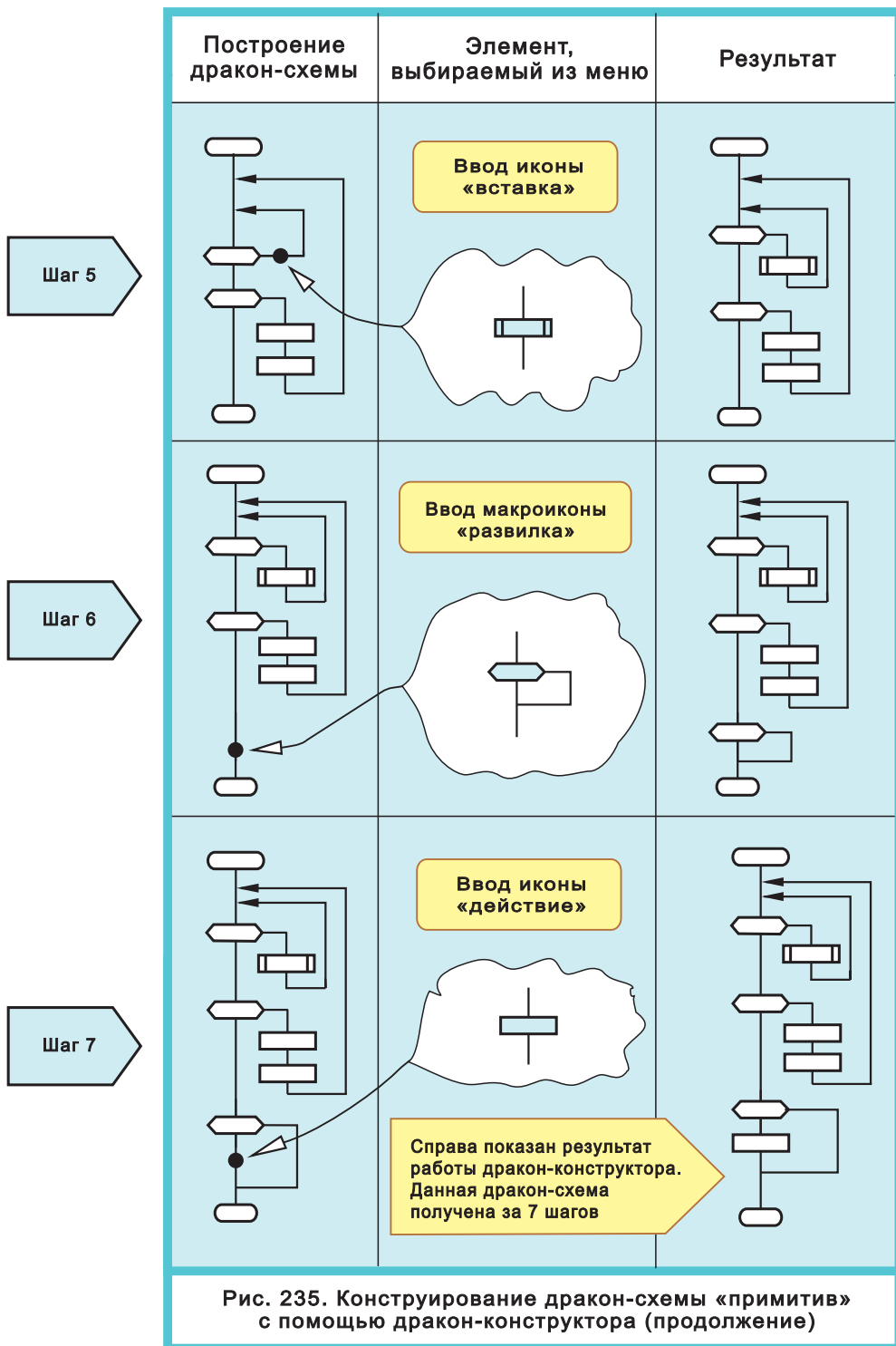
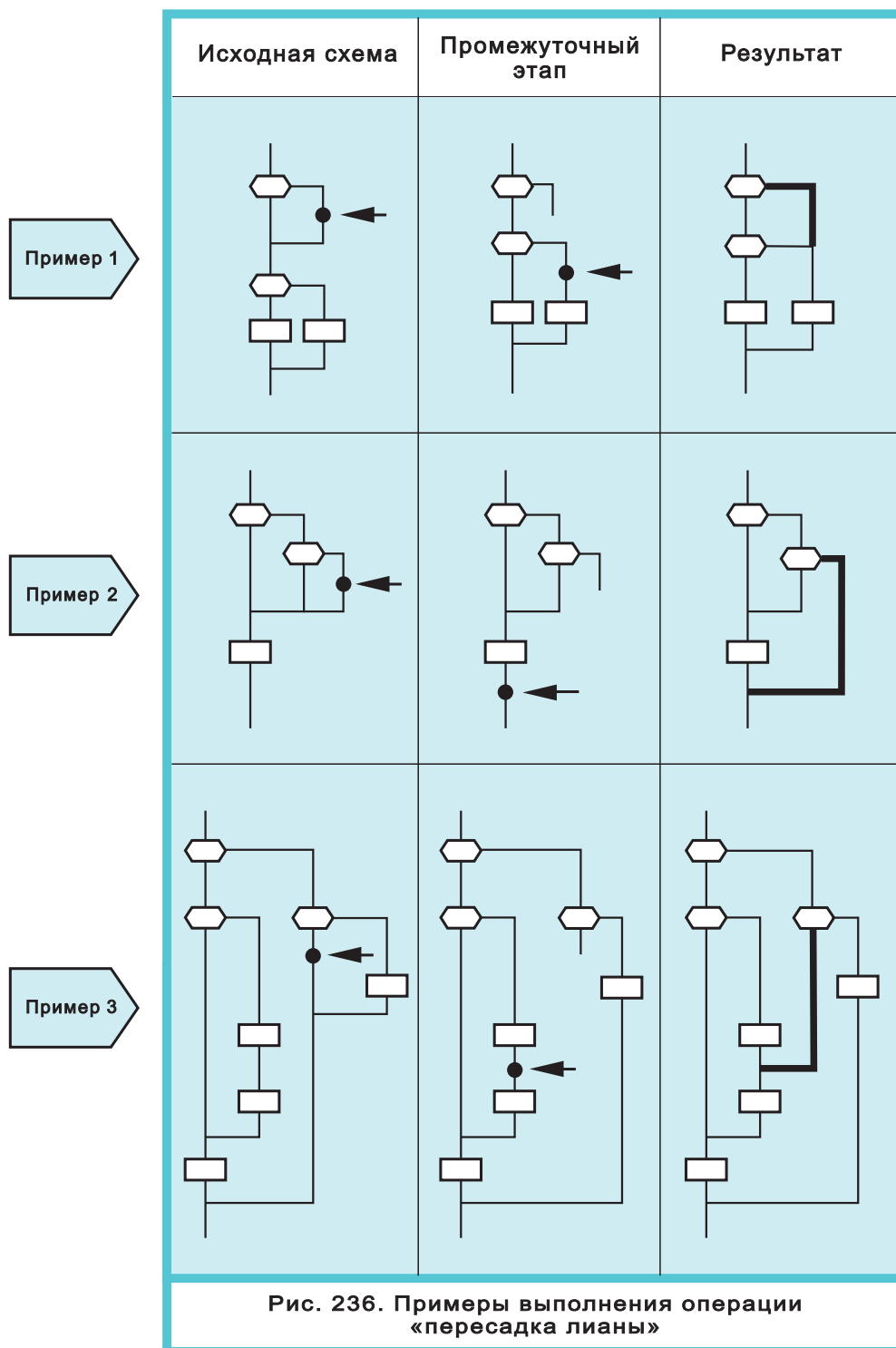
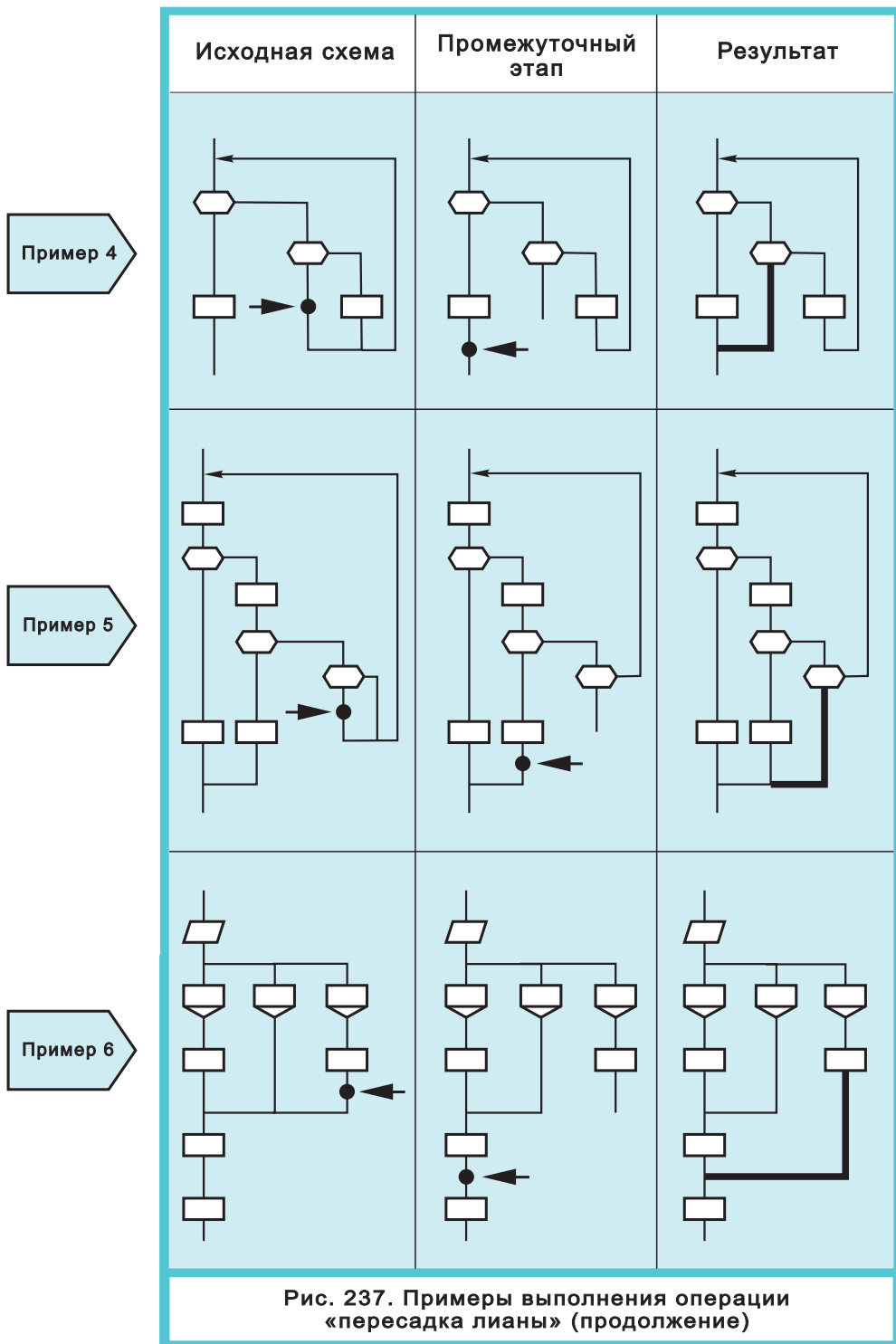


Рис. 234. Конструирование дракон-схемы «примитив» с помощью дракон-конструктора







§7. ОПЕРАЦИЯ «ЗАЗЕМЛЕНИЕ ЛИАНЫ»

Пересадка лианы применима и к примитиву, и к силуэту. В отличие от нее операция *заземление лианы* относится только к силуэту. Она служит для построения веток, имеющих несколько выходов (многоадресных веток).

Чтобы заземлить лиану, необходимо:

- организовать в ветке разветвление (с помощью макроикон «развилка» или «переключатель»);
- оторвать присоединенную к ним лиану от прежнего места;
- присоединить ее через икону «адрес» к нижней горизонтальной линии силуэта, то есть «заземлить» ее.

Операция «заземление лианы» проводится в два этапа.

Первый этап (отрыв нижнего конца лианы от своего места) осуществляется точно так же, как при пересадке лианы (рис. 238, левая графа).

На втором этапе пользователь подводит курсор к нижней линии силуэта, указывая точку, куда лиана может дотянуться, *не пересекая* других линий (рис. 238, средняя графа).

Это действие порождает автоматическое появление в нужном месте иконы «адрес». Лиана автоматически присоединяется к иконе «адрес». И через нее – к нижней линии силуэта (рис. 238, правая графа).

Заземление лианы используется при построении силуэтов с многоадресными ветками. См., например, рис. 25, 34, 85–89, 140, 144, 148, 170.

§8. ПРИМЕР ПОСТРОЕНИЯ ДРАКОН-СХЕМЫ «СИЛУЭТ»

Давайте построим силуэт, изображенный на рис. 85.

Вначале вызовем в рабочее поле заготовку-силуэт (рис. 239, вверху слева).

На первом шаге выполним операцию «добавление ветки». Для этого модифицируем заготовку, автоматически вставляя в нее еще одну ветку (рис. 239, шаг 1).

Дальнейший ход строительства ясен из рисунков 239–241.

В шагах 2–5 вставим 4 иконы «действие». В шаге 6 вставим макроикону «развилка». Результат операции виден на рис. 240, шаг 6, справа.

Затем выполним заземление лианы (шаги 7, 8). В шаге 9 добавляем признак веточного цикла (черный треугольник). Для этого помещаем два треугольника в иконы: «имя ветки» и «адрес». В заключение вставим икону «комментарий» в последнюю ветку (шаг 10).

Графическое конструирование заканчивается в момент получения желаемого слепыша (рис. 241, шаг 10, справа).

Затем в иконах записываются текстовые операторы, после чего визуальный алгоритм приобретает окончательный вид, показанный на рис. 85.

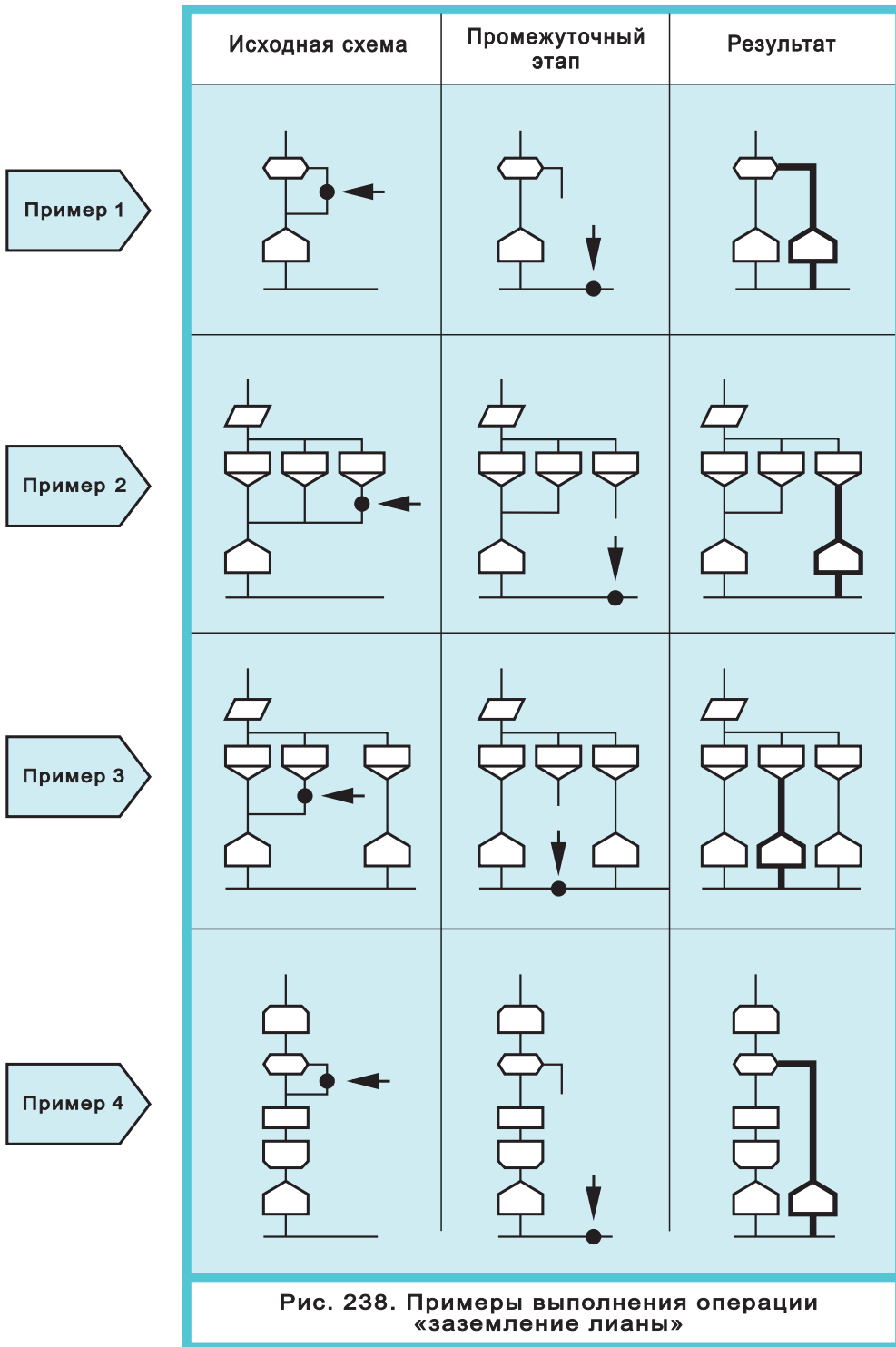


Рис. 238. Примеры выполнения операции «заземление лианы»

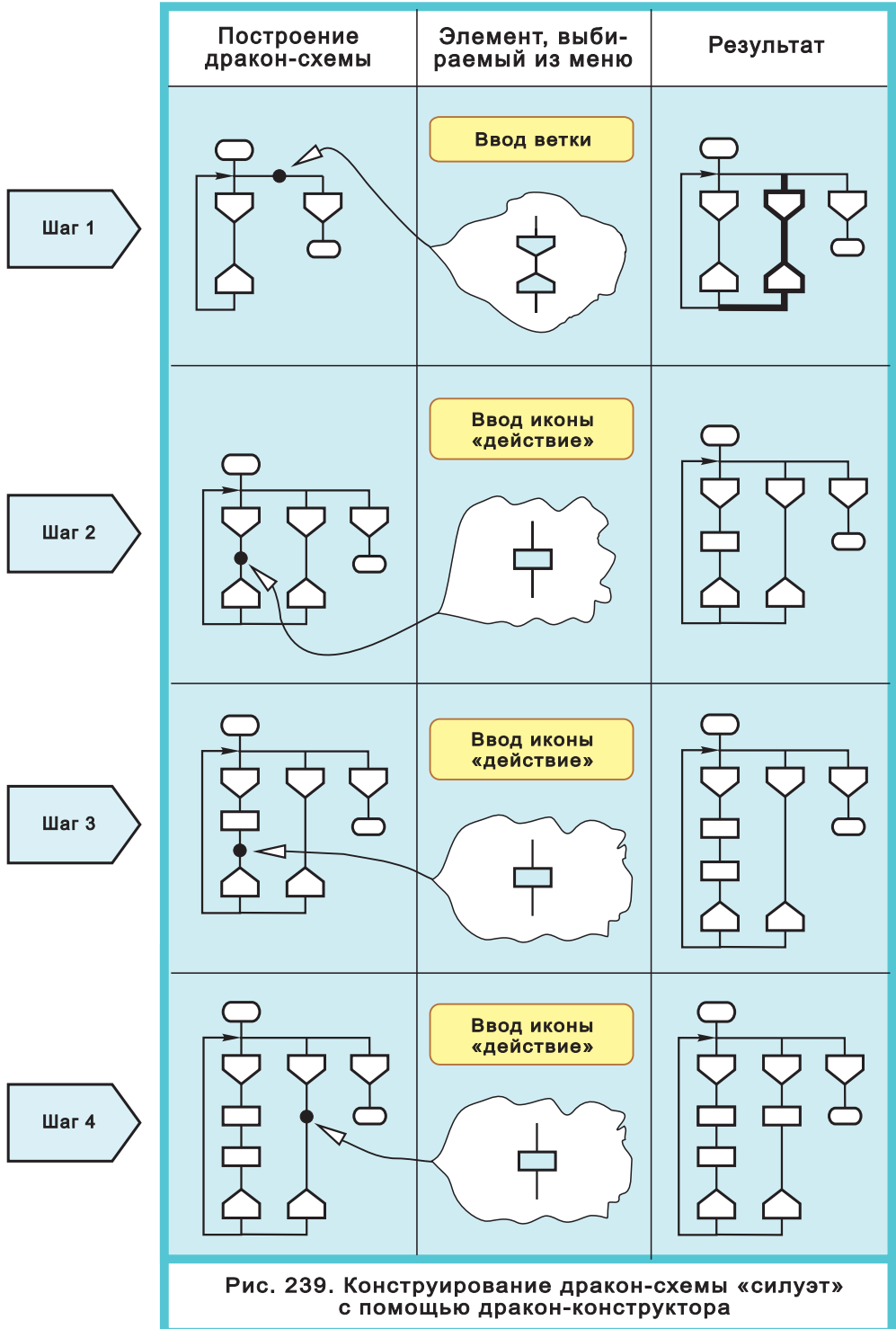
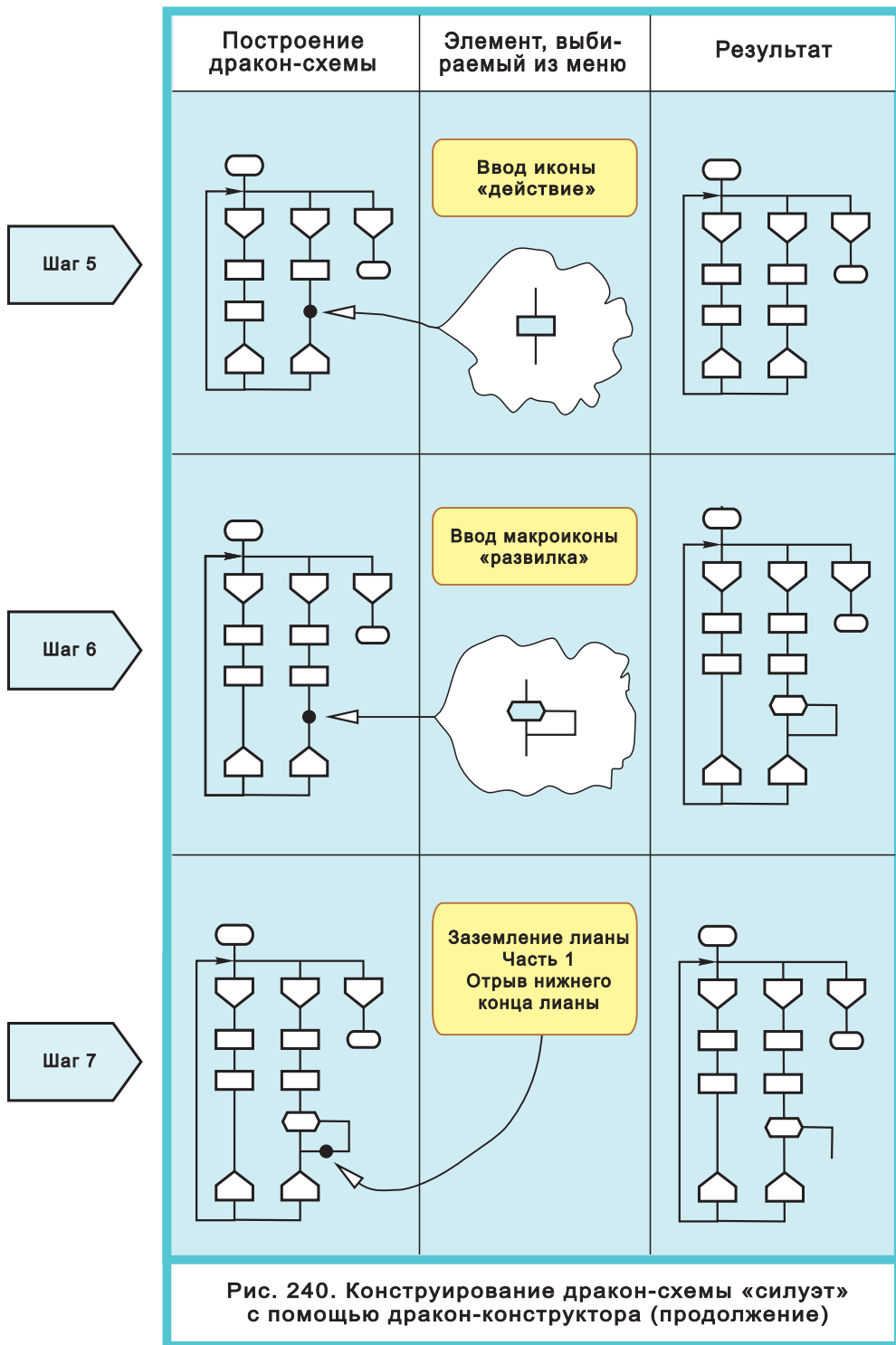


Рис. 239. Конструирование дракон-схемы «силуэт» с помощью дракон-конструктора



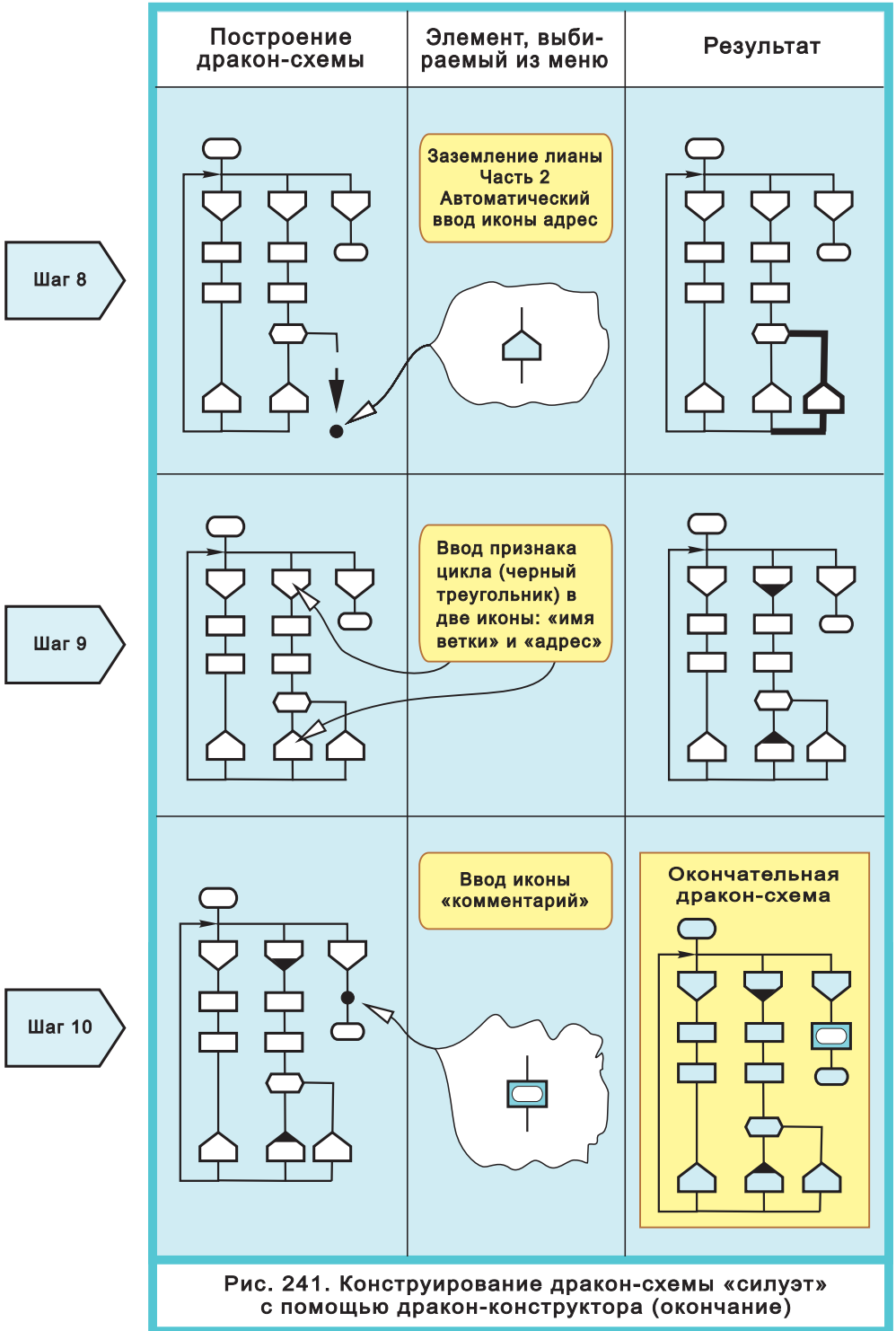


Рис. 241. Конструирование дракон-схемы «силуэт» с помощью дракон-конструктора (окончание)

§9. ФОРМИРОВАНИЕ НАДПИСЕЙ «ДА» И «НЕТ»

Возле каждой иконы «вопрос» обязательно должны быть надписи «да» и «нет». Эти надписи появляются на дракон-схеме всякий раз, когда из меню вызывается элемент, содержащий икону «вопрос» (рис. 231).

Первоначально дракон-конструктор пишет «да» у нижнего выхода иконы «вопрос» и «нет» – у правого. Пользователь может менять их местами. Для этого в конструкторе предусмотрена операция «да/нет». При выполнении этой операции, слова «да» и «нет» у выходов иконы «вопрос» меняются местами. (При этом все остальные элементы дракон-схемы остаются на своих местах).

Множественное нажатие на кнопку «да/нет» приводит к тому, что «да» и «нет» поочередно меняют свое положение.

§10. ЧЕМ ОТЛИЧАЕТСЯ ОПЕРАЦИЯ «ДА/НЕТ» ОТ РОКИРОВКИ?

При рокировке алгоритм не меняется. Потому что – вместе со сменой «да» и «нет» – плечи развилки тоже меняются местами. Следовательно, рокировка – эквивалентное преобразование алгоритма (см. главу 7).

При операции «да/нет» дело обстоит иначе, так как все остальные элементы остаются на своих местах. Следовательно, операция «да/нет» *изменяет* алгоритм.

§11. КАК ВСТАВИТЬ И УДАЛИТЬ ВЕТКУ

Рассмотрим две задачи:

- вставить ветку в силуэт;
- удалить ветку из силуэта.

Эти задачи можно решать по-разному, например, так.

Чтобы вставить ветку, надо на панели инструментов щелкнуть кнопку «вставить слева» или «вставить справа». Затем щелкнуть на иконе «имя ветки». Новая ветка будет вставлена соответственно СЛЕВА (или СПРАВА) от указанной иконы.

Чтобы удалить ветку, надо на панели инструментов щелкнуть кнопку «удалить». Затем щелкнуть на иконе «имя ветки». Данная ветка будет удалена полностью, включая все входящие в ее состав иконы.

§12. ГРАФИЧЕСКИЕ ПРАВИЛА : ХОРОШО И ПЛОХО

Рисунки, представленные в книге, во многих случаях выполнены по эргономическим правилам дракон-конструктора. Но не всегда. В ряде случаев из-за недостатка места на странице пришлось уплотнить и «искажать» рисунок. То есть нарушать эргономические правила.

Подобные искажения можно видеть на рис. 25. Укажем две ошибки:

- Многие иконы имеют разную ширину (разный горизонтальный размер).
- У некоторых икон вертикальная линия не является осью симметрии. Например, икона комментариев в ветке «Ожидание клева» и правая икона «Насади червяка» в ветке «Рыбацкая работа» расположены несимметрично из-за недостатка места.

Один из вариантов исправленной схемы дан на рис. 241а.

Рекомендация. Ширина всех икон данной ветки силуэта должна быть одинаковой. Пользователь может задать ширину по своему усмотрению. Разные ветки могут иметь разную ширину икон. (Исключениями являются иконы: «конец», «пауза», «период», «пуск таймера», «синхронизатор»).

Пример. На рис. 241а первые три ветки (Рыбная ловля, Ожидание клева, Рыбацкая работа) имеют ширину икон 19 мм. В последней ветке (Обратная дорога) ширина икон увеличена до 22 мм.

Зачем увеличена ширина? Для того, чтобы текст в двух последних иконах уместился на четырех или пяти строках (рис. 241а). Иначе (при ширине 19 мм) текст занял бы семь или восемь строк. И, следовательно, был бы неудобен для чтения (рис. 25).

Во многих случаях подобные сложности не нужны. Потому действует более простая

Рекомендация. Конструктор алгоритмов автоматически задает ширину всех икон силуэта.

Конструктор алгоритмов должен обеспечить максимальные удобства для пользователя. Программа должна по выбору пользователя выполнять любую из указанных рекомендаций (а также множество других рекомендаций, которые здесь для краткости не рассматриваются).

§13. СТРУКТУРИЗАЦИЯ ТЕКСТА В ИКОНЕ «КОММЕНТАРИЙ»

Прочитайте комментарий на рис. 171. Обратите внимание на отбивки (увеличенные интервалы между абзацами) и маркеры (черные кружки).

Отбивки и маркеры играют важную роль. Если их убрать, получим сплошной, монолитный текст, который неудобно читать. Чтобы помочь читателю, облегчить его труд и уменьшить нагрузку на глаза, конструктор алгоритмов – по указанию пользователя – должен уметь выполнять *структуризацию* текста в иконе «комментарий».

Структуризация улучшает читабельность текста за счет дробления монолитного текста на несколько приятных для глаза блоков. При этом взгляд читателя *легко* переходит от одного абзаца к другому.

Интервал до или после абзаца служит для визуального отделения одного абзаца от другого за счет пустого пространства между ними.

Структуризация призвана сделать длинный текст удобочитаемым. Обратите внимание на следующие детали (см. комментарий на рис. 171):

- Текст в иконе комментарий разбивается на части (в нашем случае, на 6 частей).
- Все части сверху и снизу разделяются отбивками (число отбивок на единицу больше числа частей; на данном рисунке 7 отбивок).
- Первая часть, если она есть, играет роль заголовка (например, «Промывать глаз можно»). Заголовок не имеет маркера.
- Остальные части начинаются с маркера.

Более простой случай показан на рис. 196 в правой ветке. Там изображены 3 маркированные части и 4 отбивки. Заголовка нет.

Возможны вариации. На рис. 186 комментарий имеет 3 части и 4 отбивки. Третья часть в свою очередь делится на три однострочных абзаца, выделяемые маркерами. Обратите внимание: однострочные абзацы не имеют разделяющих отбивок. Кроме того, в данном случае слева от маркера для выразительности делается отступ.

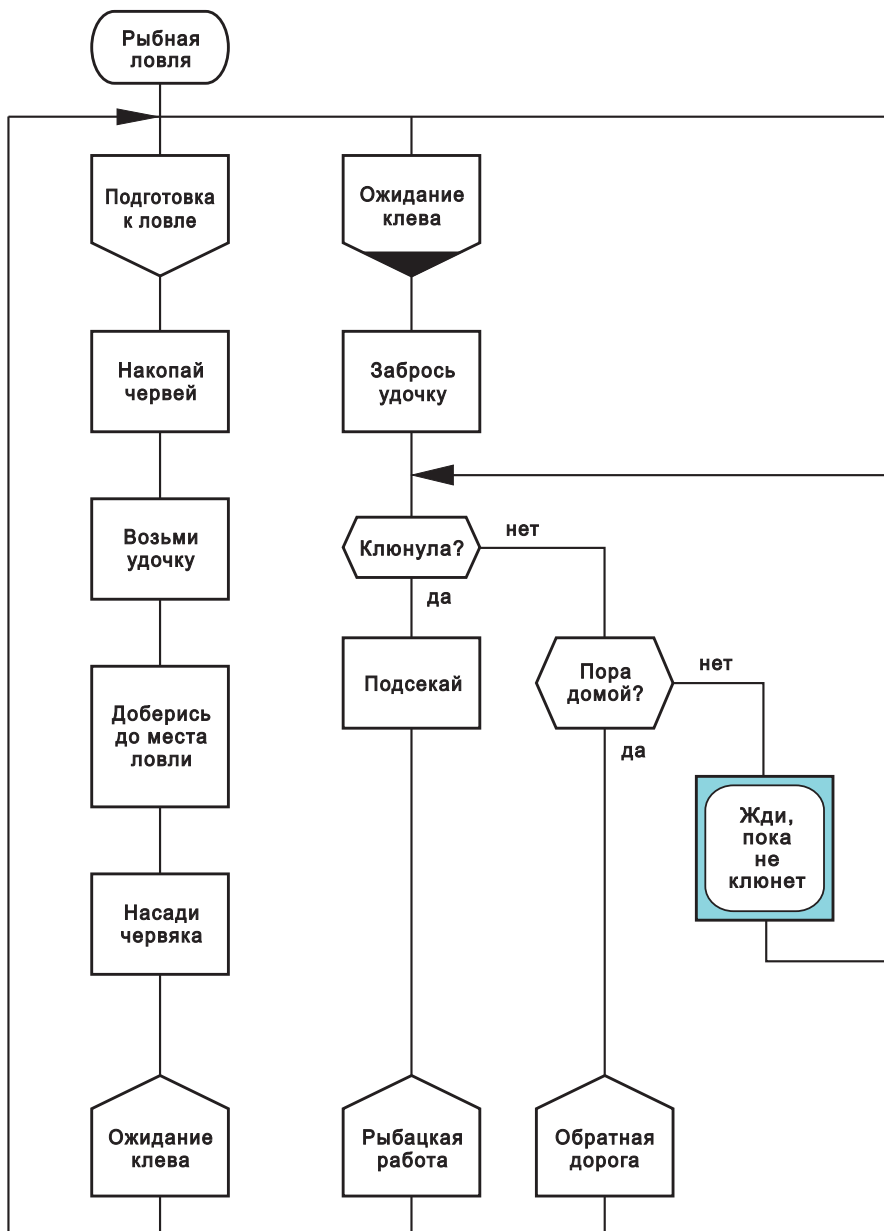
См. также рис. 187 и 197.

Чтобы обеспечить эргономичную структуризацию текста в иконе комментарий, конструктор алгоритмов должен поддерживать, как минимум, три инструмента: *отбивки, маркеры и отступы*.

Можно предложить и более серьезные требования, но здесь не место говорить о деталях.

§14. СТРУКТУРИЗАЦИЯ ТЕКСТА В ИКОНЕ «ДЕЙСТВИЕ»

Рассмотрим случай, когда надо создать дракон-схему алгоритма и больше ничего. То есть нужна просто «картинка», а трансляция в программный код и исполнение на компьютере не требуются.



Сравните рис. 25 и рис. 241а.

На рис. 25 есть недочеты:

- многие иконы имеют неодинаковый горизонтальный размер (ширину);
- у некоторых икон вертикальная линия не является осью симметрии

На рис. 241а недочеты исправлены так:

- первые три ветки имеют ширину икон 19 мм;
- в последней ветке ширина увеличена до 23 мм. Эти изменения позволили улучшить эргономичность схемы и читабельность текста в двух последних иконах.

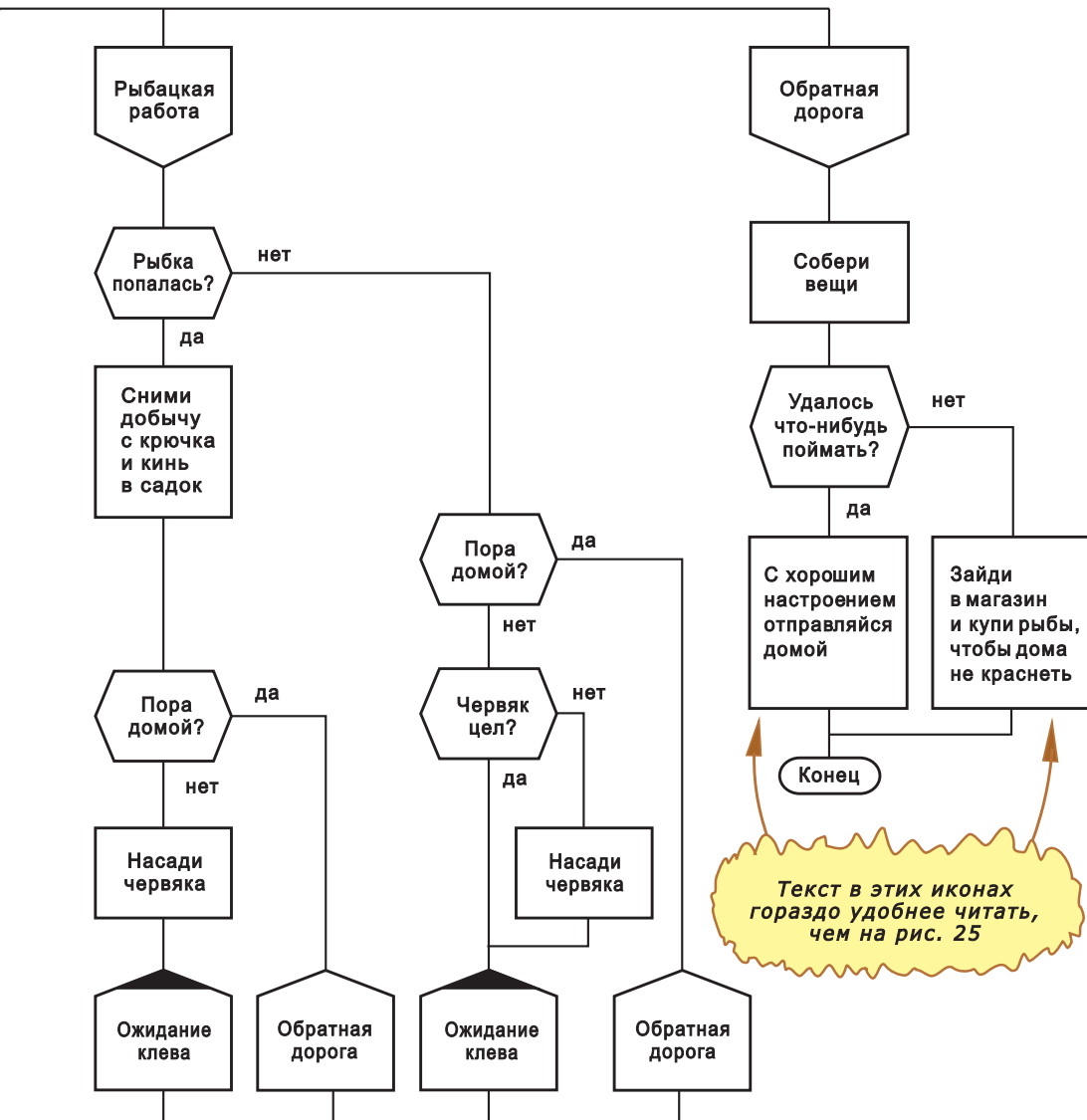


Рис. 241а. Алгоритм «Рыбная ловля»

При таких условиях иногда можно ослабить формальные (математические) требования к дракон-схеме. И разрешить писать в иконе «действие» словесное описание действий. Это может привести к ситуации, когда в иконе «действие» появится длинный текст.

Длинный, то есть сплошной, монолитный текст неудобен для чтения. Возникает уже знакомая нам ситуация. Но на этот раз применительно не к комментарию, а к тексту в иконе «действие».

Решение проблемы подробно описано в §13.

Примеры структуризации текста в иконе «действие» приведены на рис. 170, 196, 197, 206.

§15. ГДЕ СКАЧАТЬ ДРАКОН-КОНСТРУКТОР?

Первоначально среда разработки алгоритмов и программ на языке ДРАКОН была создана в Федеральном космическом агентстве для проектирования систем управления ракетно-космической техники. Она успешно применяется для разработки алгоритмов и программ бортового компьютера «Бисер», установленного на борту космических ракет – см. с. 515.

На втором этапе возникла необходимость приспособить эту среду для гражданских нужд широкого применения – для эксплуатации на персональных компьютерах, ноутбуках и др.

Эту задачу попытался решить Геннадий Тышов. Созданная им программа называется ИС Дракон. Ее можно скачать здесь:

<http://forum.oberoncore.ru/viewtopic.php?p=22669#p22669>

<http://forum.oberoncore.ru/viewforum.php?f=79>

Интернет-консультации можно получить по адресам:

Визуальный язык ДРАКОН.

<http://forum.oberoncore.ru/viewforum.php?f=62>

Алгоритмы в дракон-схемах. Обсуждение способов и правил изображения алгоритмов. Примеры.

<http://forum.oberoncore.ru/viewforum.php?f=78>

§16. ВЫВОДЫ

1. Хотя общее число икон и макроикон языка ДРАКОН равно 45, для построения любой дракон-схемы достаточно иметь небольшое меню, содержащее всего 20 графоэлементов.
2. Графическое меню существенно облегчает работу пользователя – оно дает возможность конструировать дракон-схему методом «сборки из кубиков». Для этого служит операция «ввод атома», позволяющая доставать кубики из меню и укладывать их в проектируемую дракон-схему.
3. Другие операции («пересадка лианы», «заземление лианы» и т. д.) разрешают вносить в схему логические детали, расширяющие ее функциональные возможности и улучшающие эргономическое качество.
4. Во внутренних алгоритмах дракон-конструктора реализован полный набор правил визуального синтаксиса языка ДРАКОН, что освобождает пользователя от необходимости подробно запоминать синтаксические правила.
5. Дракон-конструктор создает только правильно построенные графические схемы и исключает возможность появления запрещенных схем. Отсюда вытекает, что при работе с дракон-конструктором ошибки визуального синтаксиса *принципиально невозможны*.

ГРАФИЧЕСКИЙ СИНТАКСИС ЯЗЫКА ДРАКОН

§1. ОБЩИЕ ПОНЯТИЯ

Тезис 1. Иконы – визуальные буквы, образующие визуальный алфавит языка ДРАКОН, представленный на рис. 17.

Тезис 2. Заготовка-примитив и заготовка-силуэт – фигуры, показанные на рис. 232.

Предварительный тезис 3. Примитив – фигура, полученная путем преобразования заготовки-примитив за конечное число шагов с помощью фиксированного набора операций (перечисленных ниже в тезисе 36).

Предварительный тезис 4. Силуэт – фигура, полученная путем преобразования заготовки-силуэт за конечное число шагов с помощью фиксированного набора операций (перечисленных ниже в тезисе 37).

Тезис 5. Дракон-схема – общее понятие для обозначения примитива и силуэта.

§2. ШАМПУР-БЛОК

Тезис 6. Шампур-блок – часть дракон-схемы, имеющая один вход сверху и один выход снизу, содержащая одну или несколько икон, причем:

- вход и выход лежат на одной вертикали, через которую проходит путь от входа к выходу;
- через каждую икону, входящую в состав шампур-блока, проходит хотя бы один путь от входа к выходу;
- в состав шампур-блока могут входить любые иконы за исключением следующих: заголовок, конец, формальные параметры, петля силуэта, соединитель.

Тезис 7. Главная вертикаль шампур-блока – вертикаль, соединяющая его вход и выход.

Остальные вертикали, если они есть, находятся правее главной. Все вертикали шампур-блока ориентированы сверху вниз, кроме цепей, используемых для организации петли цикла.

§3. ОПЕРАЦИЯ «ВВОД АТОМА»

Тезис 8. Атомы – фигуры, изображенные на рис. 242. Эти фигуры используются в операции «ввод атома». Любой атом является шампур-блоком.

Тезис 9. Валентная точка – точка, принадлежащая заготовке или дракон-схеме, в которой разрешается произвести разрыв соединительной линии, чтобы в место разрыв вставить атом с помощью операции «ввод атома». Для краткости валентную точку можно называть *точкой ввода*.

Тезис 10. Ввод атома – преобразование заготовки или дракон-схемы, выполняемое следующим образом: производится разрыв соединительной линии в валентной точке и в это место вставляется атом, как показано на рис. 233.

§4. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ ОБ АТОМАХ

Тезис 11. Атомы делятся на простые и составные. Простой атом содержит одну икону, составной – не менее двух или разветвление (рис. 242).

Тезис 12. Функциональный атом – простой атом, не являющийся пустым оператором. Таковы все простые атомы, кроме комментария.

Тезис 13. Составные атомы бывают пустые и непустые. В непустом есть хотя бы один функциональный атом. В пустом нет ни одного.

Тезис 14. В полностью законченной дракон-схеме не должно быть ни одного пустого атома (так как последний эквивалентен пустому оператору). Пустые атомы разрешается использовать на всех этапах построения дракон-схемы, кроме заключительного.

§5. КРИТИЧЕСКИЕ И НЕЙТРАЛЬНЫЕ ТОЧКИ

Тезис 15. Валентные точки (точки ввода) делятся на нейтральные и критические.

Тезис 16. Точка называется нейтральной, если применение операции «ввод атома» к данной точке является возможным, но не обязательным. В отличие от нее критическая точка требует обязательного ввода атома.

Тезис 17. Валентные точки находятся в заготовках и атомах. Они показаны на рис. 232 и 242, где нейтральные точки обозначены светлыми кружками, критические – жирными точками.

Тезис 18. Если в фигуре (заготовке или атоме) одна критическая точка, ввод атома обязательно производится именно в нее; при этом крити-

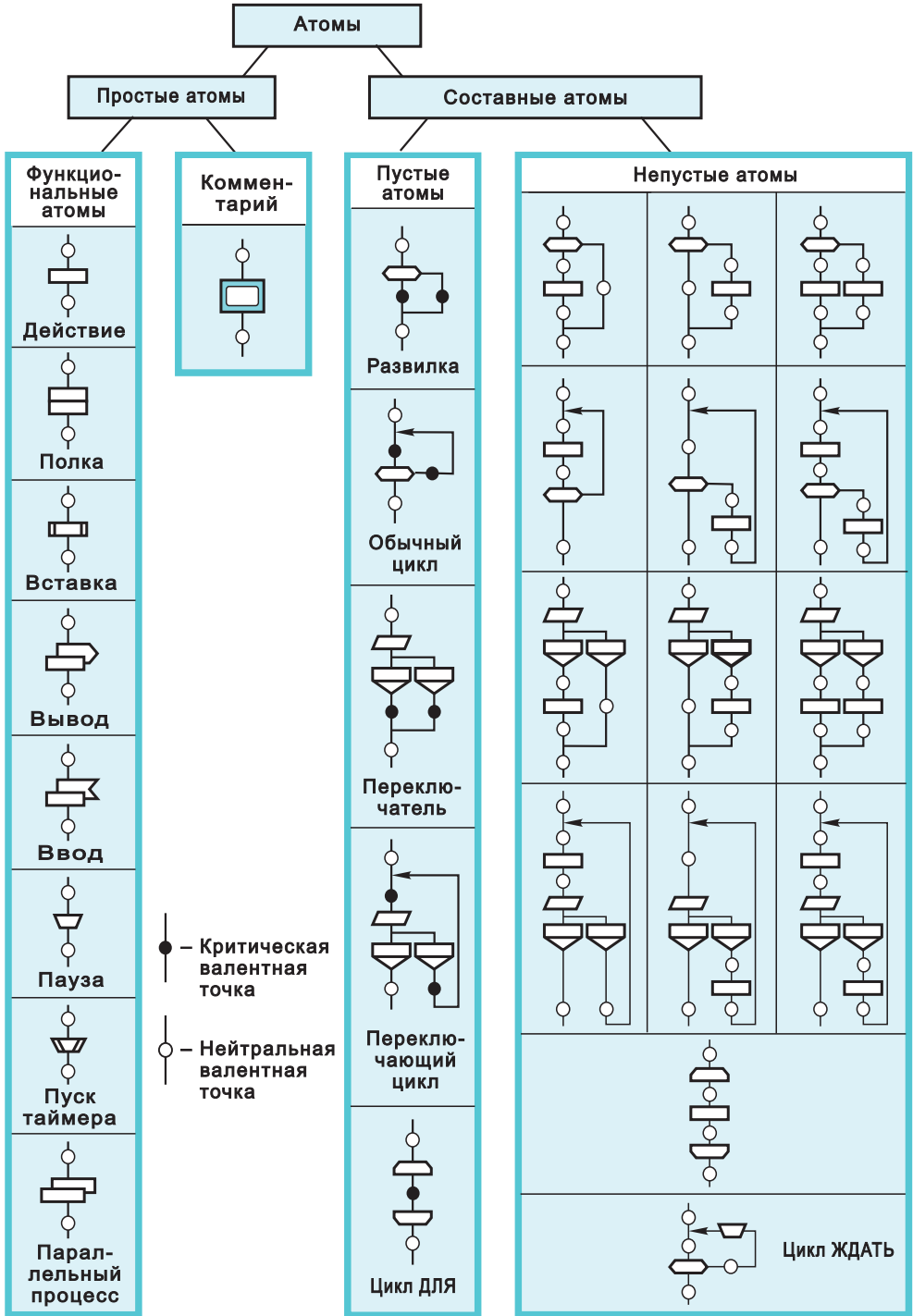


Рис. 242. Атомы и валентные точки

ческая точка уничтожается. Если фигура имеет две критические точки, обязательный ввод атома делается только в одну из них; при этом критическая точка, в которую произведен ввод, уничтожается, а другая критическая точка нейтрализуется, т. е. становится нейтральной.

Тезис 19. Полная совокупность критических точек охватывает:

- критические точки в пустых атомах;
- одну критическую точку в заготовке-примитив;
- одну критическую точку в заготовке-силуэт.

Тезис 20. Полная совокупность нейтральных точек охватывает:

- входные и выходные точки атомов;
- две внутренние точки в атоме «цикл ЖДАТЬ»;
- одну точку в заготовке-силуэт;
- точки, полученные в результате нейтрализации критических точек.

§6. ПРАВИЛА ИСПОЛЬЗОВАНИЯ ОПЕРАЦИИ «ВВОД АТОМА» ПРИ ПОСТРОЕНИИ ДРАКОН-СХЕМЫ

Тезис 21. Операция «ввод атома» применяется для ввода только простых и пустых атомов, а также цикла ЖДАТЬ. Ввод непустого атома осуществляется в два этапа; сначала вводится пустой атом, затем в его критическую точку вводится функциональный атом.

Пояснение. Ввод пустого атома – очень удобный строительный прием. Он позволяет обеспечить богатство и разнообразие создаваемых дракон-схем и используемых в них конфигураций. Среди последних особую роль играет так называемая «матрешка».

Тезис 22. *Матрешка* – фигура, полученная путем ввода пустого атома в критическую точку пустого атома, а также путем многократного вложения пустых и непустых атомов друг в друга (рис. 243).

Тезис 23. Матрешка бывает пустой (если все содержащиеся в ней атомы пустые), частично пустой (если в ней есть как пустые, так и непустые атомы) и непустой (если все ее атомы непустые). См. рис. 244–246.

Пояснение. После того как пользователь эффективно использовал пустые атомы для придания дракон-схеме желаемой конфигурации, он должен убрать их из схемы.

Тезис 24. Чтобы устранить пустые атомы из дракон-схемы, есть два способа:

- превратить пустой атом в непустой;
- преобразовать пустой атом в пустую матрешку, затем превратить ее в непустую.

Тезис 25. Устранение из дракон-схемы пустых атомов автоматически приводит к уничтожению всех критических точек.

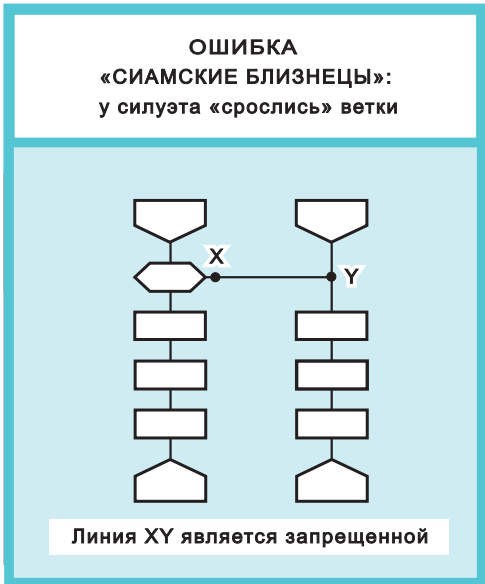
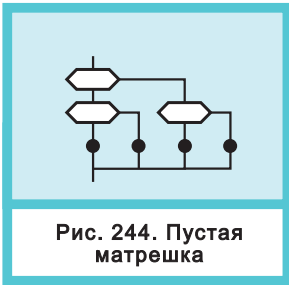
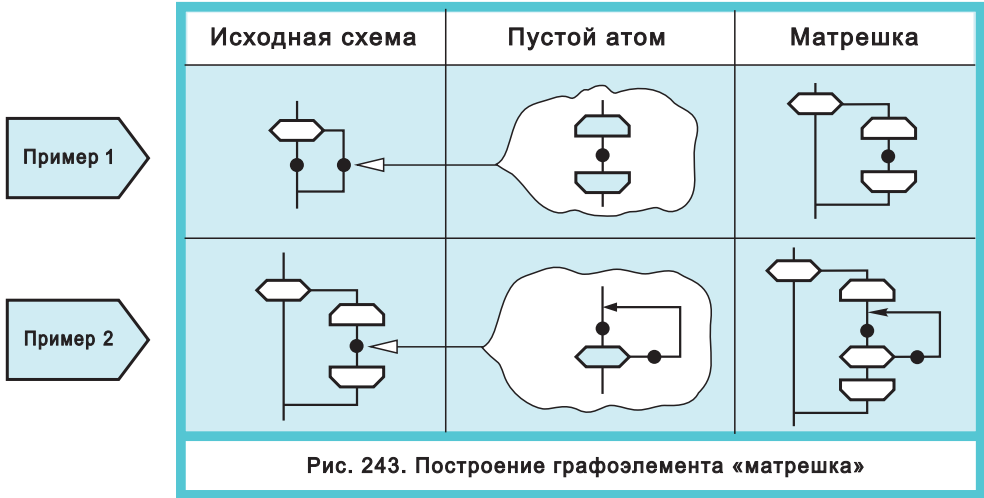


Рис. 247. Ошибка «Сиамские близнецы» — это запрещенная связь между ветками

§7. ЛИАНА

Тезис 26. Лиана – часть дракон-схемы, имеющая один вход сверху и один выход снизу, именуемые «началом лианы» и «концом лианы» соответственно. Началом лианы может быть любой выход икон «вопрос» и «вариант», если он (выход) не является петлей цикла. Концом лианы считается точка слияния, в которой нижняя часть лианы соединяется с другой линией (концом лианы не может быть неразветвленный вход иконы).

Тезис 27. Лиана может быть нагруженной (если она содержит иконы) и ненагруженной (если это просто линия).

§8. ПЕРЕСАДКА ЛИАНЫ

Тезис 28. Пересадка лианы – преобразование дракон-схемы, выполняемое за четыре шага.

Шаг 1. Производится отрыв конца лианы от точки присоединения (рис. 236, 237).

Шаг 2. Конец лианы с помощью вертикальных и горизонтальных линий присоединяется к любой валентной точке, куда лиана может дотянуться без пересечения с другими линиями (рис. 236, 237). При этом запрещается:

- формировать второй вход в ветку (ошибка «сиамские близнецы» – см. рис. 247);
- образовывать новый цикл;
- создавать второй вход в цикл.

Однако разрешается строить новый путь из середины обычного цикла к единственному входу в этот цикл, создавая визуальный эквивалент оператора *continue* языка Си (см. рис. 167, пример 7, а также рис. 83).

Шаг 3. Производится эквивалентное преобразование топологии дракон-схемы, чтобы

- лиане не пришлось загибаться наверх (рис. 248);
- соблюдались правила построения шампур-блока (рис. 249).

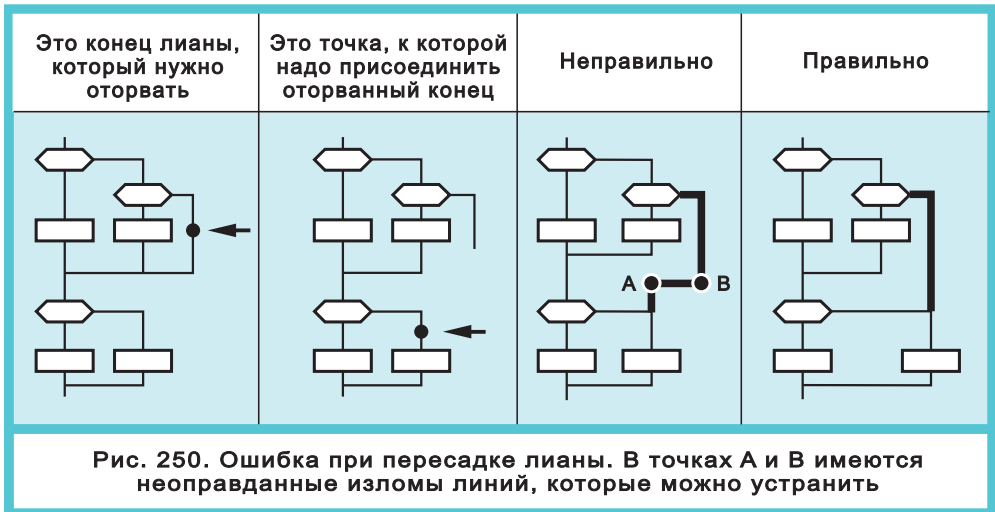
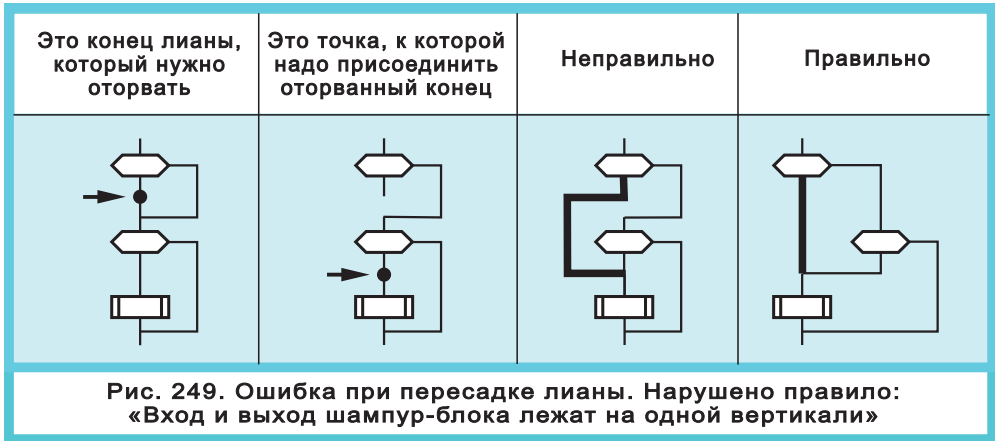
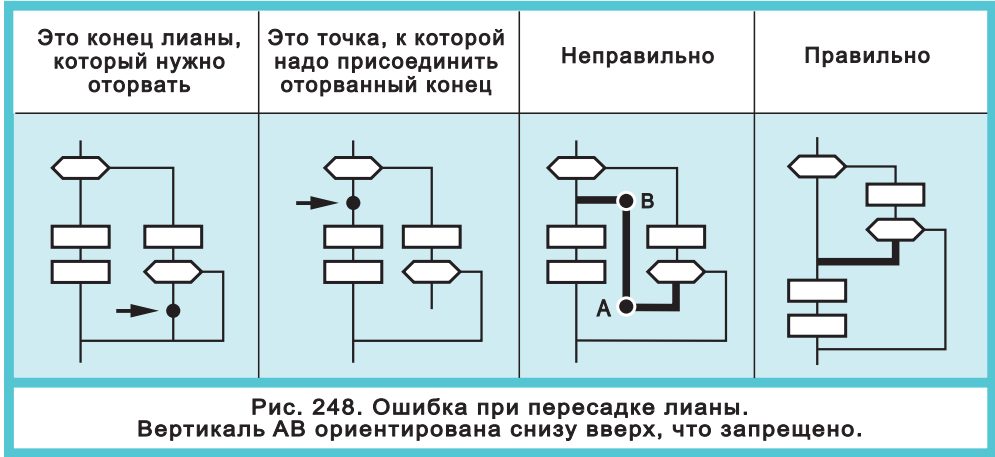
Шаг 4. Устраняются неоправданные изломы линий (рис. 250).

§9. ЗАЗЕМЛЕНИЕ ЛИАНЫ

Тезис 29. Заземление лианы – преобразование дракон-схемы, выполняемое за четыре шага.

Шаг 1. Производится отрыв конца лианы от точки присоединения (рис. 238).

Шаг 2. Конец лианы с помощью вертикальной линии присоединяется к любой точке нижней горизонтальной линии силуэта, куда он



может дотянуться, не пересекая другие линии.

Шаг 3. Производится разрыв линии в нижней части лианы и в место разрыва вставляется икона «адрес» (рис. 238, 240 (шаг 7), 241 (шаг 8)).

Шаг 4. Устраняются неоправданные изломы линий.

§10. ПРОЧИЕ ОПЕРАЦИИ

Тезис 30. Боковое присоединение – преобразование дракон-схемы, с помощью которого в схему добавляются иконы «синхронизатор» или «формальные параметры».

Икона «синхронизатор» размещается слева от другой иконы и соединяется с ней горизонтальным отростком. Перечень икон, к которым осуществляется боковое присоединение синхронизатора, показан на рис. 18 (п. 8–20).

Икона «формальные параметры» размещается справа от иконы «заголовок» и соединяется с ней горизонтальным отростком, как показано на рис. 18 (п. 1).

Тезис 31. Добавление варианта – преобразование дракон-схемы, с помощью которого в атом «переключатель» добавляется еще одна икона «вариант». Число добавлений не более 14, так что максимальное число вариантов в переключателе равно 16. Эти цифры могут быть увеличены.

Тезис 32. Добавление ветки – преобразование силуэта, в который добавляется еще одна ветка. Число добавлений не более 30, так что максимальное число веток в силуэте равно 32. Эти цифры могут быть увеличены.

Тезис 33. Удаление последней ветки – преобразование силуэта, при котором удаляется крайняя правая ветка. Этот прием используется при описании бесконечного процесса, как показано в примерах на рис. 140 и 148.

Ограничение. В силуэте не может быть меньше двух веток.

Тезис 34. Удаление конца примитива – преобразование примитива, при котором удаляется икона «конец». Это необходимо для описания бесконечного параллельного процесса.

Тезис 35. Дополнительный вход – преобразование силуэта, с помощью которого добавляется еще одна икона «заголовок», которая размещается над любой иконой «имя ветки» (кроме левой) и соединяется с ней вертикальным отростком. При этом на верхней горизонтальной линии силуэта рисуют направленную вправо стрелку, как показано в примере на рис. 144 справа.

Ограничение. При наличии веточного цикла запрещается присоединять дополнительный заголовок к середине веточного цикла.

§11. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Тезис 36. Любая правильно построенная дракон-схема «примитив» является результатом преобразования заготовки-примитив с помощью конечного числа операций: ввод атома, пересадка лианы, добавление варианта, боковое присоединение, удаление конца примитива.

Тезис 37. Любая правильно построенная дракон-схема «силуэт» является результатом преобразования заготовки-силуэт с помощью конечного числа операций: ввод атома, добавление ветки, пересадка лианы, заземление лианы, добавление варианта, боковое присоединение, удаление последней ветки, дополнительный вход.

Пояснение. Тезисы 36 и 37 могут рассматриваться как окончательные определения понятий «примитив» и «силуэт».

§12. ВЫВОДЫ

1. Изложенные выше 37 тезисов (вместе с рисунками, на которые они ссылаются) дают однозначное описание визуального синтаксиса.
2. Это описание является достаточным для построения дракон-конструктора, способного решить две задачи:
 - нарисовать (в соответствии с указаниями пользователя) любую абстрактную дракон-схему (слепыш), принадлежащую множеству правильно построенных (удовлетворяющих требованиям визуального синтаксиса) дракон-схем;
 - создать в памяти компьютера формальное описание построенной схемы, пригодное (после заполнения икон текстовыми операторами) для трансляции в объектные коды или для выполнения программы в режиме интерпретации.