ШКОЛЬНИКУ ДЛЯ РАЗВИТИЯ ИНТЕЛЛЕКТА

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

«РОСМЭН»

ШКОЛЬНИКУ ДЛЯ РАЗВИТИЯ ИНТЕЛЛЕКТА

Как развить интеллект?

Как помочь школьнику стать умнее? Как научить его действовать разумно и логично? Начните с этой книги. Она учит алгоритмическому мышлению, т. е. искусству правильно мыслить и разумно планировать свои действия. Читать ее легко и приятно. С первых страниц ребенок попадает в захватывающую атмосферу увлекательной игры, которая быстро развивает логическую смекалку. Незаметно для себя он втягивается в шумную компанию сказочных персонажей и вместе с ними учится решать все более сложные задачи.

«РОСМЭН»

ISBN 5-257-00929-3

9 785257 009297

ШКОЛЬНИКУ ДЛЯ РАЗВИТИЯ ИНТЕЛЛЕКТА

Владимир Пароджанов

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА,

ИЛИ ВОЛШЕБНЫЙ ДРАКОН В ГОСТЯХ У МУРЗИКА

МОСКВА «РОСМЭН» 2000

ББК 74.202.4
П18

Эта умная, веселая книга посвящена важному разделу информатики — алгоритмам и принципам их построения. Просто и, что самое главное, очень доступно для детей автор знакомит их с волшебным миром алгоритмов. С первых страниц ребенок попадает в мир игры и сказочных героев, которые помогают ему учиться решать все более сложные задачи. Автор объясняет, что алгоритм — это последовательность действий, ведущих к поставленной цели. Автор учит ребенка мыслить и действовать разумно, заранее планировать свои действия, правильно идти к цели. Книга удачно оформлена, содержит большое количество увлекательных примеров с рисунками алгоритмов. Ее можно рекомендовать родителям для совместного чтения с детьми, школьникам — для развития интеллекта, а также как дополнительное учебное пособие в школах при изучении темы «Алгоритмы».

Оформление серии А. Ефремова
Художник А. Лукьянов

Пароджанов В. Д.

П18 Занимательная информатика, или Волшебный Дракон в гостях у Мурзика / Оформл. серии А. Ефремова; Художн. А. Лукьянов. — Переизд. — М.: РОСМЭН, 2000. — 160 с. — (Школьнику для развития интеллекта).

ISBN 5-257-00929-3

В книге, адресованной детям младшего и среднего школьного возраста, подробно рассматривается один из важных разделов информатики — алгоритмы и принципы их построения.

ISBN 5-257-00929-3 **ББК 74.202.4**

ДЕЙСТВУЮЩИЕ ЛИЦА

Папа Циркуль, учитель информатики, самый веселый, какие только бывают. У него длинные усы вразлет, как ножки циркуля. И сам он похож на циркуль — худой и длинноногий.

Мурзик, шустрый и смышленый ученик, непоседа и забияка. Любит говорить стихами.

Алина и Хлястик, другие ученики Папы Циркуля. Они вечно что-нибудь путают и задают смешные вопросы.

Змей Горыныч, известный злодей и разбойник.

Илья Муромец, доблестный робот, который победил Змея при помощи хитроумного алгоритма.

Кащей Бессмертный, маг и чародей. Очень любит информатику. Когда сердится, из его ноздрей вырываются огонь и клубы дыма.

Волшебный Дракон, который на самом деле вовсе и не дракон. То есть он, конечно, дракон, только совсем не драконистый. В общем, сами увидите.

И множество других персонажей, с которыми происходят необыкновенные приключения.

Вся эта дружная и пестрая команда будет очень стараться и помогать вам изо всех сил. Скажу по секрету: с помощью наших забавных героев изучить информатику — проще простого. Хотя, конечно, придется и самому мозгами пошевелить.

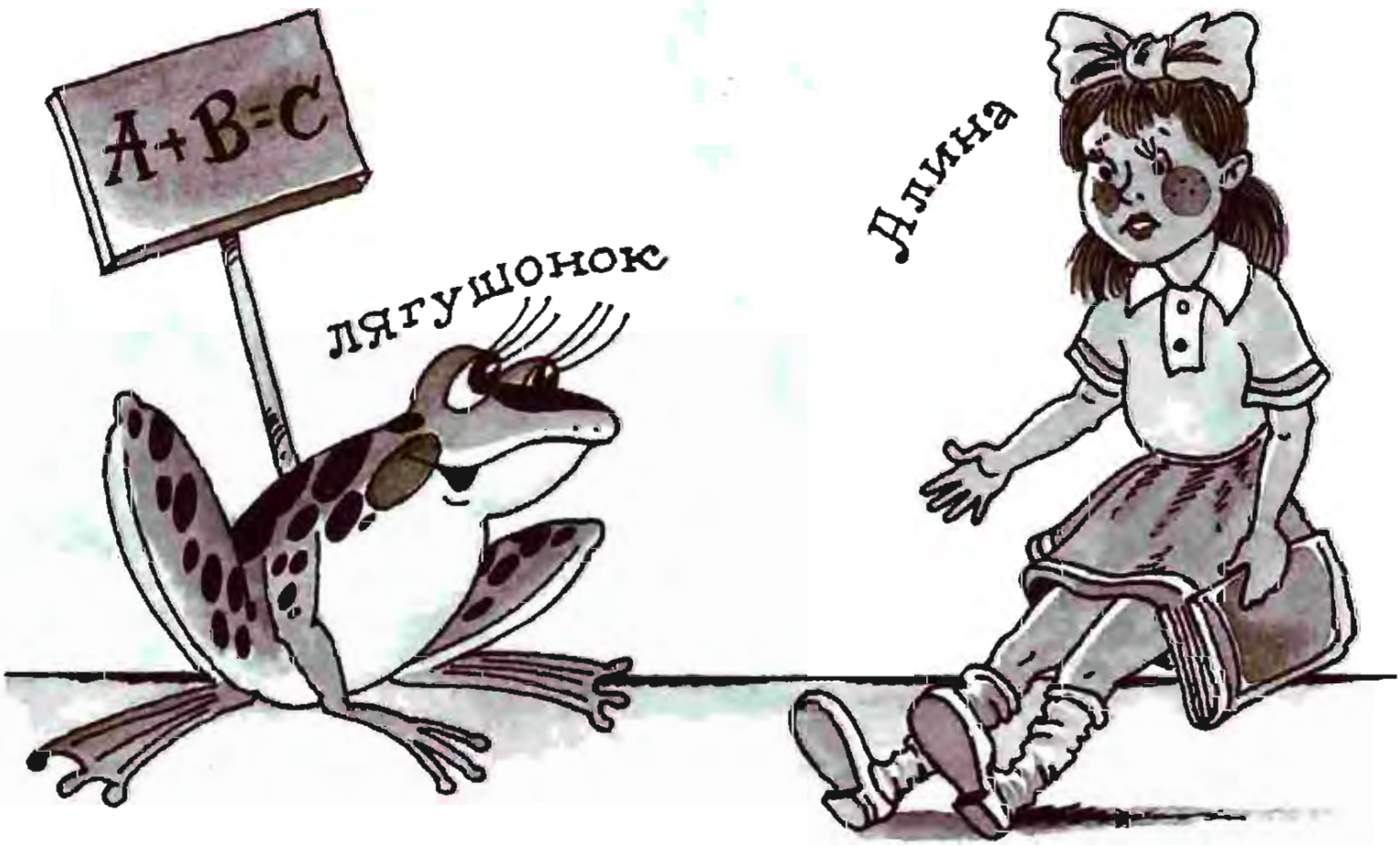
В добрый путь!

Алина. Это ужасно! Я просто в отчаянии! Все вокруг только и говорят о компьютерах и информации. А как же я? Я тоже мечтаю изучить информатику, но увы... Ведь для этого надо знать математику, а я ее терпеть не могу. Когда я вижу математическую формулу, мне кажется, что я проглотила лягушку!

Папа Циркуль. Не печалься, Алина, твоему горю нетрудно помочь. Эта книжка как раз для тебя. В ней почти совсем нет формул. Ну, а если какой мелкий лягушонок случайно заскочит на твою дорожку, не пугайся — твои верные друзья — Мурзик и я — в случае чего всегда тебя выручат.

Итак, для кого написана эта книга?

Для тех, кто математику
С рождения боится,
Кто в панике от формул
Ныряет под кровать,
Но хочет с Информатикой
Теснее подружиться
И все ее секреты
Скорее разузнать!



СТРАНА ВОЛШЕБНЫХ АЛГОРИТМОВ. ЛЕГКОЕ И ПРИЯТНОЕ ПУТЕШЕСТВИЕ ДЛЯ НАЧИНАЮЩИХ

§ 1. АЛГОРИТМ «СТРОГАЯ МАМА»

Папа Циркуль. Здравствуй, Мурзик! Начинаем наш первый урок. Внимательно посмотри на рис. 1 и своими словами перескажи его содержание. Тебе все понятно?

Рис. 1. Алгоритм «Строгая мама»

Мурзик. Тут и понимать нечего. Мама с дочкой то ладят, то не ладят. Если ладят, дочка довольна, если нет, она плачет. Правда, там есть загадочное слово «алгоритм», которое я слышу первый раз в жизни. Понятия не имею, что оно означает.

Папа Циркуль. Тогда взгляни на рисунки 2 и 3. Может быть, они тебе подскажут смысл слова «алгоритм».

Мурзик. Постойте-постойте... Кажется, я начинаю догадываться...

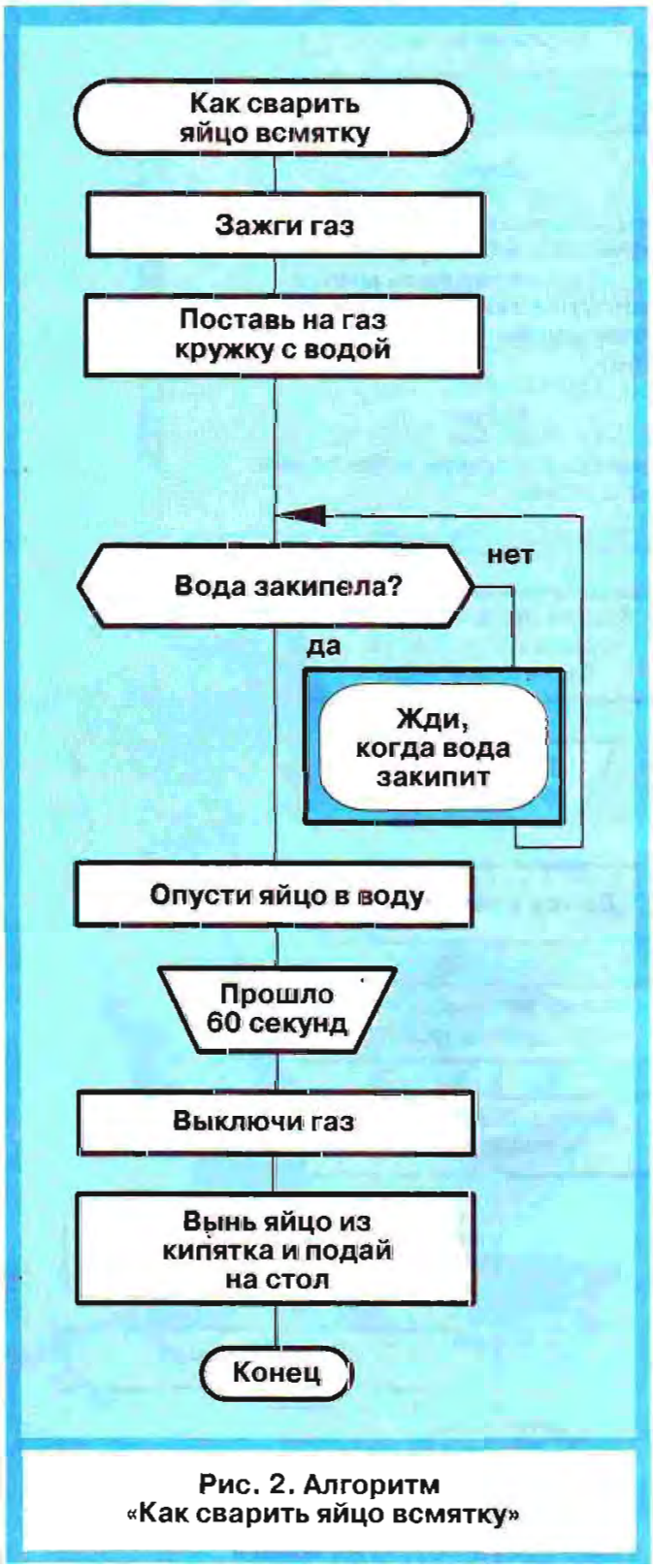


Рис. 2. Алгоритм «Как сварить яйцо всмятку»

Рис. 3. Алгоритм «Как сварить яйцо вкрутую»

§ 2. О ЧЕМ ДОГАДАЛСЯ МУРЗИК, ИЛИ ЧТО ТАКОЕ АЛГОРИТМ

— Вам хочется чаю? Сейчас организуем. Для этого нужно:

- Вскипятить воду.
- Налить в чашку заварку.
- Добавить кипятку.
- Положить сахар.
- Размешать сахар ложечкой.
- Выпить чай.

Чтобы приготовить чай или кофе, испечь яблочный пирог или сделать рагу из молодого барашка с луком, хреном и петрушкой, надо приложить руки, т. е. выполнить последовательность действий. Эта последовательность называется *алгоритм*. Таким образом, алгоритм очень похож на кулинарный рецепт. С другой стороны, алгоритм похож на военный приказ. Например, армейская прибаутка:

«Направо-налево равняйсь!
На первый-второй рассчитайсь!»

есть не что иное, как шутливый приказ (алгоритм), состоящий из трех команд:

- Повернись направо.
- Равняйся налево.
- Рассчитайся на первый-второй.

Мурзик. Я тоже знаю алгоритм из двух команд:

- Пойди туда, не знаю куда.
- Принеси то, не знаю что.

Папа Циркуль. Нет, дружок, это не алгоритм, а белиберда. Потому что твои команды невозможно выполнить. В средние века среди ученых монахов ходил примерно такой же анекдот под названием «рецепт алхимика»:

- Возьми что нужно.
- Сделай как следует.
- Получишь то, что желаешь.

Мурзик. Чем же он плох? По-моему, все логично.

Папа Циркуль. Тем, что этот рецепт (алгоритм) неконкретный, неоднозначный. Всяк может толковать его на свой лад. И начнется: кто в лес, кто по дрова. Один возьмет топор и построит дом, другой сорвет яблоко и съест его, третий женится и родит ребенка. И каждый может сказать: я честно выполнил все команды алгоритма — сделал как следует, и получил то, что желал. И будет по-своему прав. Потому что рецепт алхимика — это не алгоритм, а полная неразбериха.

Запомни, Мурзик: приказ, рецепт, план, правило или инструкцию только тогда можно назвать алгоритмом, если все команды, предписания и условия описаны детально, точно и однозначно. Тогда все исполнители команд будут понимать их одинаково и делать в точности одно и то же. Если уж ты взялся сочинять алгоритмы, будь добр — устрани малейшую неясность и двусмысленность. Иначе работа пойдет насмарку. Порою даже лишняя запятая может оказаться коварной ловушкой. Кстати, помнишь, один царь, решая судьбу человека, долго думал, какой из указов подписать:

Указ № 1. Казнить, нельзя помиловать.
Указ № 2. Казнить нельзя, помиловать.

Ты видишь — отличие только в запятой. А смысл алгоритма меняется на противоположный.

Мурзик. Ну, это сказка...

Папа Циркуль. Сказка ложь, да в ней намек. Между прочим, в Америке один горе-ученый тоже ошибся в запятой. А знаешь, что получилось? Компьютер неверно понял команду, и ракета, вместо того, чтобы лететь на Венеру, помчалась неизвестно куда. Все были в шоке, а главный конструктор неделю рыдал и рвал на себе волосы — ведь такая ракета дороже золота. Ее строили много лет. Пропал огромный труд. А ты говоришь, запятая.

Что такое алгоритм

Это последовательность действий, ведущая к поставленной цели

§ 3. КАК НАРИСОВАТЬ АЛГОРИТМ?

— Васёк! Слетай в булочную — одна нога здесь, другая там. Купи буханку белого и буханку черного.

Можно ли изобразить эти команды в виде чертежа? Нет ничего проще. Выделим главное, отбросив ненужные словесные бантики, и получим картинку на рис. 4. Эта картинка называется *блок-схема алгоритма* или просто *блок-схема*. Алгоритм состоит из четырех команд, каждая из которых помещена в отдельную прямоугольную рамочку:

- Зайди в булочную
- Купи буханку белого
- Купи буханку черного
- Отнеси хлеб домой

«Идем за хлебом» — это название алгоритма. А «Конец» — он и есть конец. Кончил дело — гуляй смело!

Мурзик. А если в булочной нет хлеба? Что тогда?

Папа Циркуль. Ты попал не в бровь, а в глаз. Алгоритм на рис. 4 имеет смысл только в том случае, когда в булочной обязательно есть и белый, и черный хлеб. Если это не так, если ты пришел в булочную, а хлеба нет, значит, команду «Купи хлеб» невозможно исполнить. Такой алгоритм (в котором есть неисполнимые команды) никому не нужен. Он не работает. Пользы от него — никакой.

Мурзик. Что же делать?

Папа Циркуль. Надо исправить

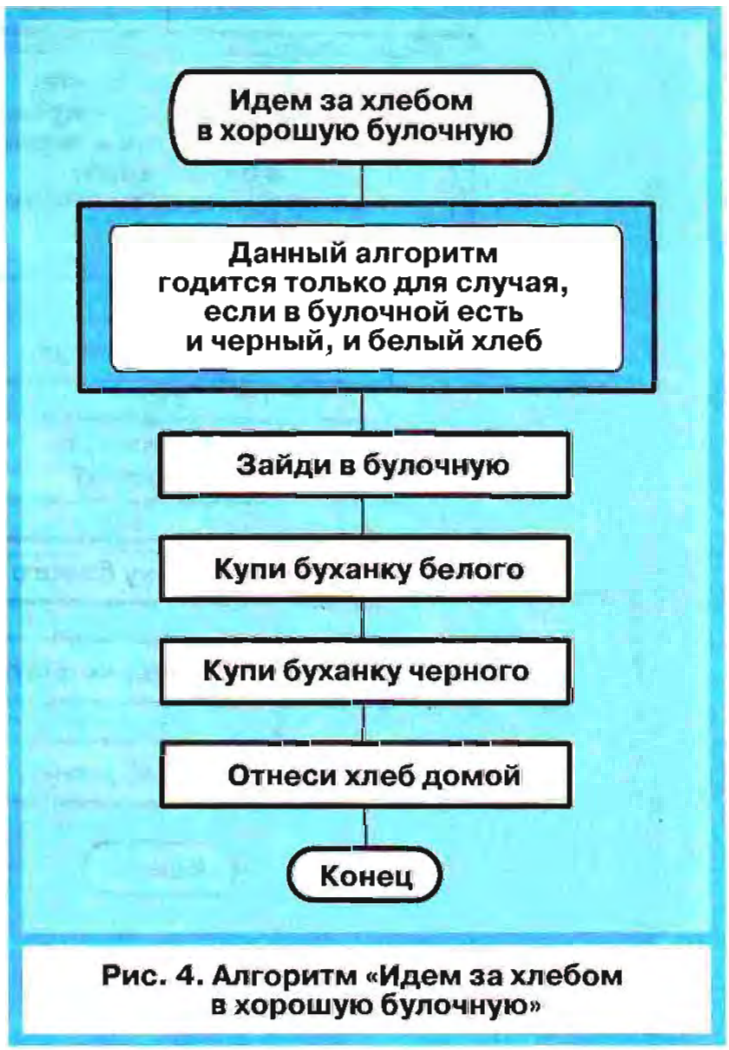


Рис. 4. Алгоритм «Идем за хлебом в хорошую булочную»

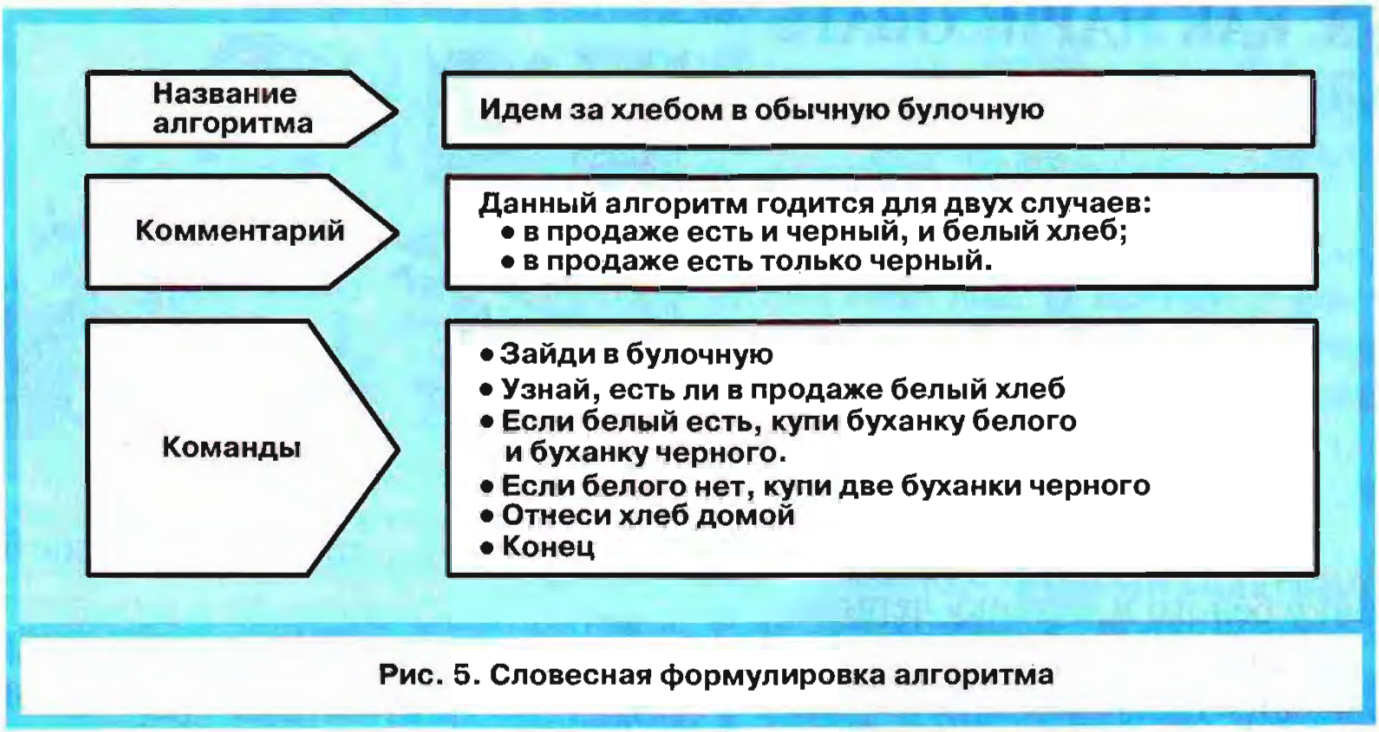


Рис. 5. Словесная формулировка алгоритма

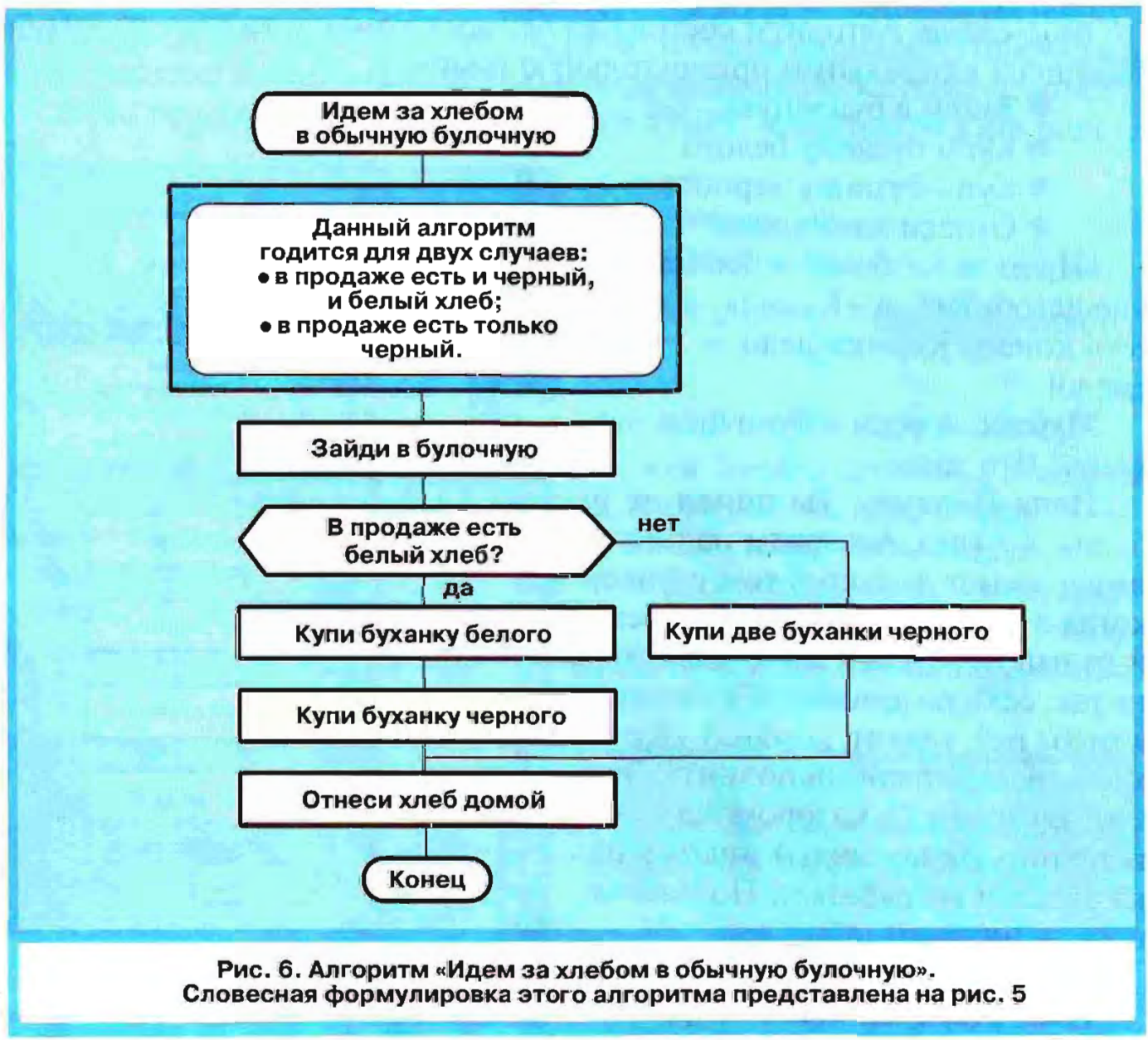


Рис. 6. Алгоритм «Идем за хлебом в обычную булочную». Словесная формулировка этого алгоритма представлена на рис. 5

ошибку и придумать другой алгоритм, который учитывает реальные условия. Предположим, в булочной черный хлеб водится всегда, а с белым случаются перебои. В этой ситуации алгоритм будет выглядеть так:

«Васёк! Сбегай в булочную. Буханку белого и буханку черного. Если белого не будет, возьмешь две черного».

Мурзик. Какой же это алгоритм? Это Васин брат из окна кричит.

Папа Циркуль. Тем не менее он излагает свою мысль очень четко. Из его на первый взгляд легкомысленных слов вытекает безупречная формулировка алгоритма, показанная на рис. 5. Этот алгоритм, как и любой другой, можно изобразить в виде блок-схемы (см. рис. 6).

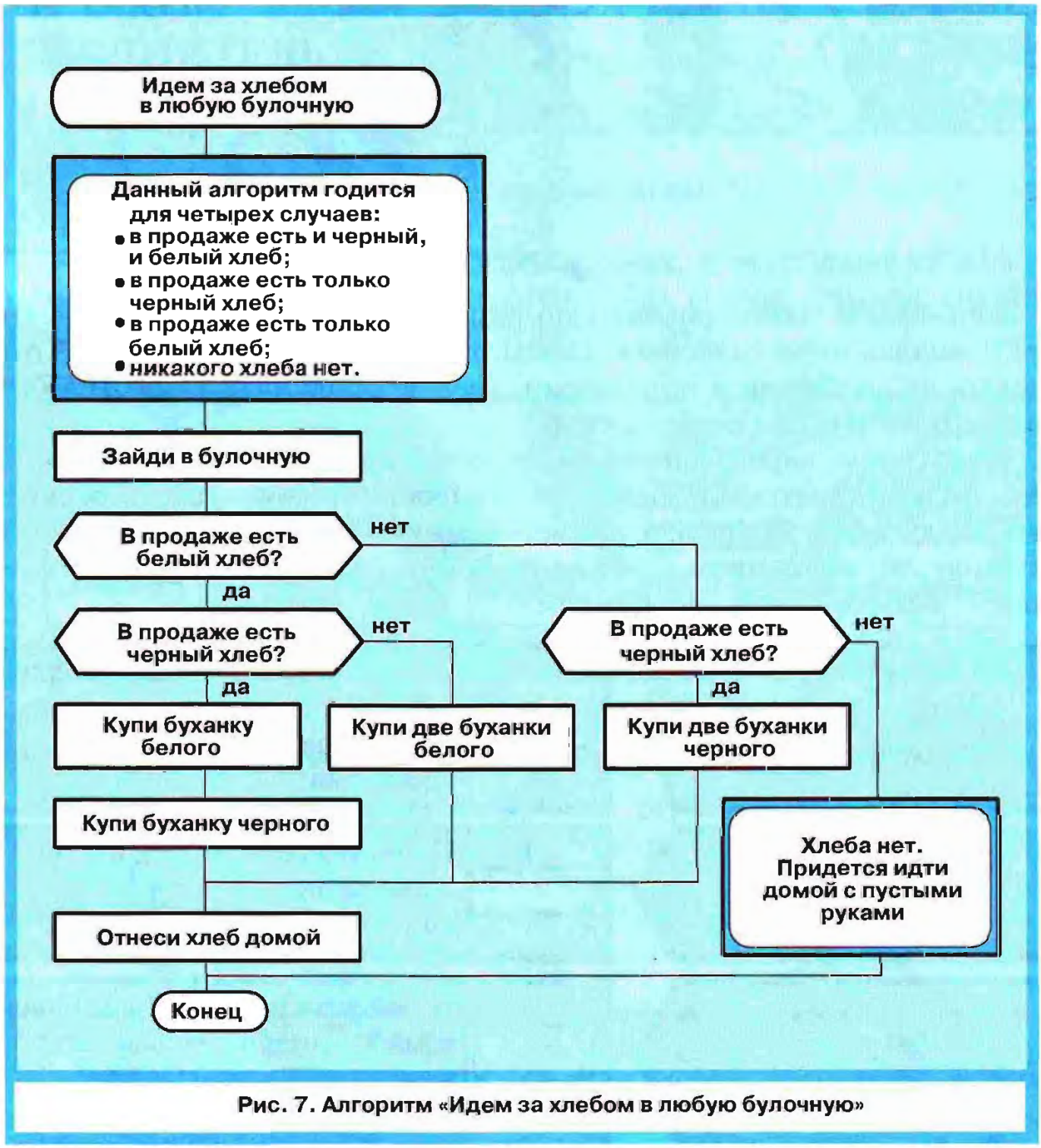


Рис. 7. Алгоритм «Идем за хлебом в любую булочную»

Алина. А для нашего магазина последний алгоритм все равно не годится. У нас булочная совсем плохая, придешь, а на дверях записка: «Сегодня хлеба нет и не будет».

Папа Циркуль. Верно, Алина, для твоей булочной нужен другой, более сложный алгоритм. Например, такой:

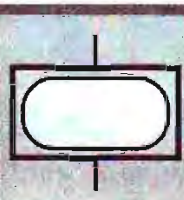
«Василек! Сходи за хлебом. Одну белого и одну черного. Белого не будет — купи две черного. И наоборот. А если никакого нет, обойдемся».

Соответствующая блок-схема показана на рис. 7.

§ 4. ЧТО ТАКОЕ ЯЗЫК «ДРАКОН»?

Блок-схемы можно рисовать по-разному. В старину, при царе Горохе, применялись неудачные блок-схемы, громоздкие и запутанные. Чтобы сделать их удобными и понятными, ученые придумали язык ДРАКОН, которым мы и будем пользоваться.

Блок-схемы, нарисованные по правилам языка ДРАКОН, называются *дракон-схемы*. Дракон-схемы состоят из простых рамок — *икон*, соединенных между собой. Например, на рис. 4 семь икон. Верхняя называется «Заголовок», та, что пониже — «Комментарий», следующие четыре — «Действие», а самая нижняя — «Конец».

Что такое «Заголовок»	 **Овальная икона, расположенная в самом начале схемы, внутри которой пишут название алгоритма**
Что такое «Действие»	 **Прямоугольная икона, внутри которой пишут команды алгоритма**
Что такое «Комментарий»	 **Прямоугольная икона с двойной рамкой, внутри которой пишут не команды, а пояснения**

Что такое «Конец»

Маленькая овальная икона, расположенная в самом конце алгоритма, внутри которой пишут слово «Конец»

§ 5. ЧТО ТАКОЕ СОЧИНИТЕЛЬ? ЧТО ТАКОЕ ИСПОЛНИТЕЛЬ?

Сочинитель — тот, кто придумывает алгоритмы. *Исполнитель* — тот, кто их исполняет (рис. 8).

Сочинять алгоритмы может только человек. А исполняют их всевозможные машины, например, компьютер, робот, станок, луноход, спутник и даже некоторые механические игрушки.

Рис. 8. Как взаимодействуют сочинитель и исполнитель алгоритма

Между сочинителем и исполнителем — огромная разница. Сочинитель —это творец. Его разум и вдохновение позволяют создавать умные алгоритмы, которые управляют полетом ракет, предсказывают погоду и делают много других полезных вещей. Дело же исполнителя — точно и безропотно исполнять команды алгоритма.

Пояснение

- **Сочинитель — это человек**
- **Исполнитель — это машина**

§ 6. ЧЕМ ОТЛИЧАЕТСЯ МАЛЬЧИК ВАСЯ ОТ РОБОТА ВАСИ

Предположим, что Вася из примера на рис. 4 — робот. Давайте разберемся: чем отличается такой робот от настоящего живого мальчика?

Мальчик Вася, ученик школы №12, умеет многое: играть в футбол, ходить в кино, драться с Петькой, есть пирожки и учить уроки. А робот Вася из примера на рис. 4 ничего этого не знает и не может. Он не умеет есть и пить, дразнить девчонок и ходить в школу. Он не может кричать, свистеть, визжать от восторга и удирать от погони. Он не способен прыгать на одной ноге и стрелять из рогатки. Единственное, что он знает и понимает — это четыре команды, запечатанные в его электронных мозгах: 1) Зайди в булочную. 2) Купи буханку белого. 3) Купи буханку черного. 4) Отнеси хлеб домой. Точное выполнение этих команд — его единственная обязанность.

Если мы хоть чуть-чуть изменим вызубренную им команду — например, вместо «Купи буханку белого» скажем: «Купи кило сушек», — он нас не поймет и откажется повиноваться. И уж тем более он не поймет команду: «Если деньги останутся, купи себе мороженое».

Бедный робот Вася! Ему не позавидуешь! Он обречен всю жизнь, до самой смерти (пока не остановится его механическое сердце, сделанное из металлических клапанов, винтиков и шпунтиков) выполнять свои четыре команды, с помощью которых творец-сочинитель навеки запрограммировал его поведение.

Мурзик. Я все-таки не понял: почему живой человек не может быть механическим исполнителем?

Папа Циркуль. Предположим, в алгоритме есть команда: «Мурзик! Прыгни с десятого этажа!» Ты прыгнешь?

Мурзик. Что я, дурак, что ли!

Папа Циркуль. Вот то-то и оно. Ты рассуждаешь! Этим ты и отличаешься от исполнителя, который выполняет команды не рассуждая.

§ 7. ЧТО ТАКОЕ РЕПЕРТУАР ИСПОЛНИТЕЛЯ?

Исполнители бывают разные. Самый простой исполнитель — реле, которое включает прожектор, освещающий памятник Пушкину. Исполнитель под названием «реле» может выполнять всего две команды:

1. Включи прожектор.
2. Выключи прожектор.

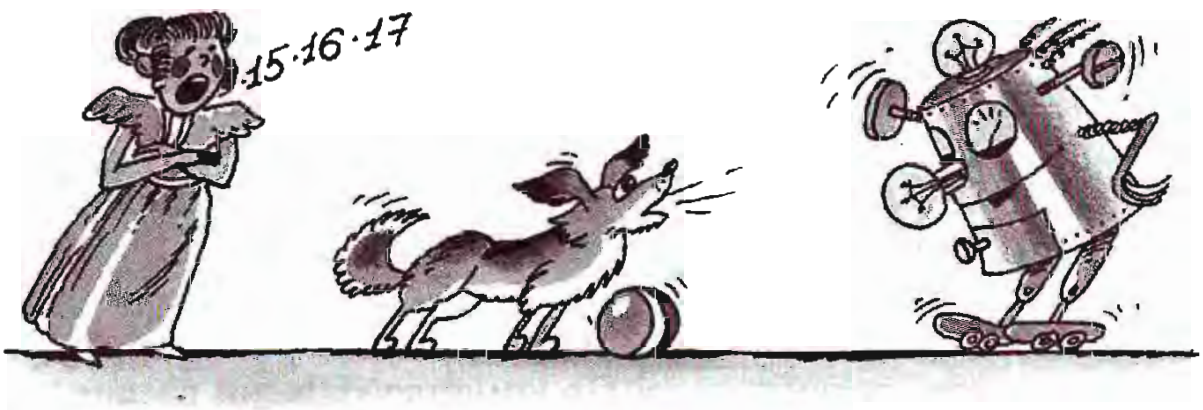
Исполнитель Ванька-встанька тоже знает две команды: «Ложись!» и «Встань!». Если скомандовать ему «Беги!», — он пропустит такой приказ мимо ушей, просто потому что не умеет бегать. Это значит, что команда «Беги!» не входит в его репертуар.

Мурзик. При чем здесь репертуар?

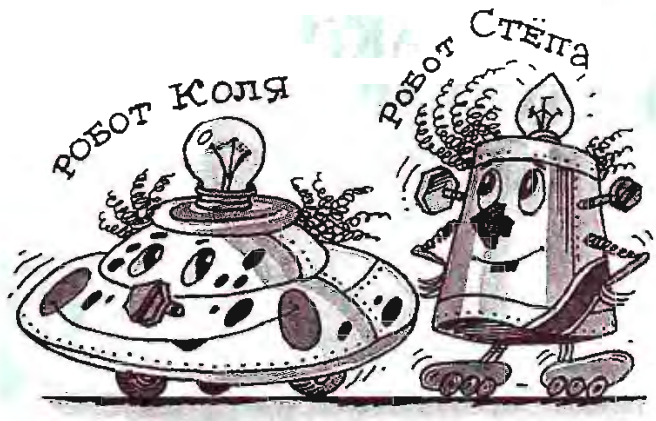
Папа Циркуль. Если в театре можно посмотреть тридцать пьес — говорят, что эти пьесы составляют его репертуар. Если в репертуаре певицы семнадцать песен, значит, она может исполнять со сцены ровно семнадцать песен. А восемнадцатую она петь не станет, потому что еще не выучила. Точно так же и робот-исполнитель. В его репертуар входят только те команды, которые он «выучил» — а точнее, которые запрограммированы в его электронных мозгах.

Мурзик. Понятно. Значит, репертуар моей Жучки состоит всего из трех команд: «Сидеть!», «К ноге!» и «Сторожи!». Ничего другого она не умеет. Потому что бестолковая. Но я ее все равно люблю.

Что такое репертуар исполнителя → **Это набор команд, которые исполнитель может выполнять**



§ 8. О ТОМ, КАК ДВА РОБОТА ПОСПОРИЛИ, ЧЕЙ РЕПЕРТУАР ЛУЧШЕ



Жили-были два закадычных друга, два исполнителя, два робота — Коля и Степа. Жили они в Шахматной Стране, где, как известно, вся земля поделена на клетки. Каждый робот целиком умещается в одной клетке и может ходить из клетки в клетку лишь по определенным правилам. Колю будем обозначать кружком, а Степу — стрелкой (рис. 9).

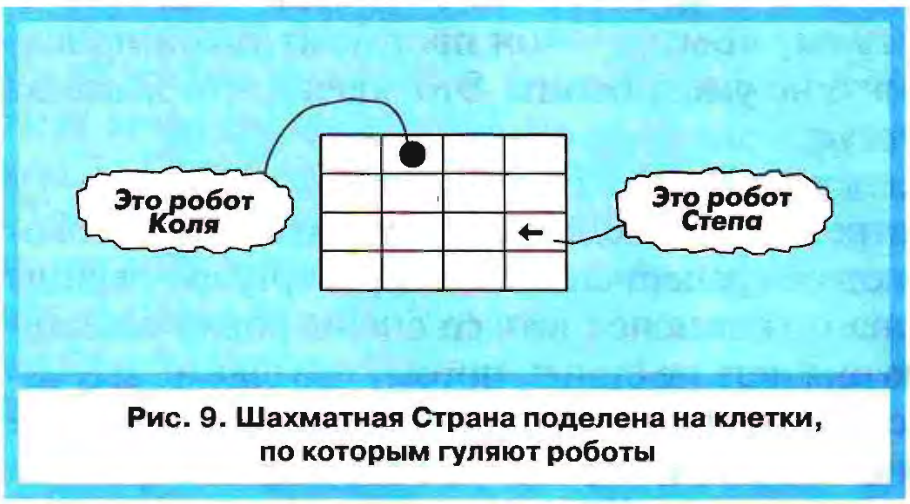


Рис. 9. Шахматная Страна поделена на клетки, по которым гуляют роботы

Мурзик. Я придумал правило. «Коля» и «Кружок» начинаются на букву «К». «Степа» и «Стрелка» — на букву «С». Поэтому легко запомнить: Коля — Кружок, а Степа — Стрелка.

Про Колю знает каждый пес:
Он Круглый, как тарелка.

Куда направлен Степин нос,
Показывает Стрелка.

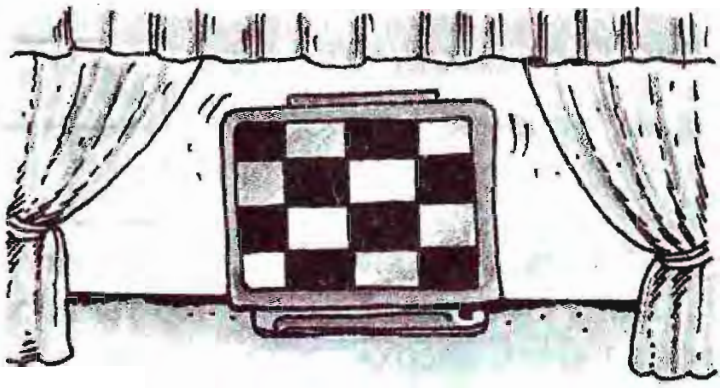
И начали роботы спорить. Чуть до драки не дошло! «У меня репертуар замечательный! Самый лучший!» — кричит Коля. Но и Степа не сдается: «Много ты понимаешь! У меня в сто раз лучше!» Оба врут, конечно. Потому что у обоих, если честно, репертуар слабенький. Коля умеет исполнять четыре команды: «Шагни вправо», «Шагни влево», «Шагни вверх», «Шагни вниз». А Степа выучил только две команды: «Шагни вперед» и «Повернись налево».

Услышав команду «Шагни вправо», Коля делает ровно один шаг и переходит в соседнюю клетку справа. По команде «Шагни влево» он делает

один шаг влево. Команда «Шагни вверх» перемещает его в соседнюю верхнюю клетку. «Шагни вниз» — и Коля спускается вниз на один шаг.

А у Степы совсем другие правила. Выполняя команду «Шагни вперед», Степа делает один шаг вперед и тоже переходит в соседнюю клетку. Но в какую? Это зависит от того, куда смотрит его нос, т. е. от направления стрелки. Если стрелка направлена вниз — Степа опустится в нижнюю клетку, если вверх — поднимется в верхнюю и т. д. Услышав команду «Повернись налево», Степа, не покидая клетки, поворачивается налево.

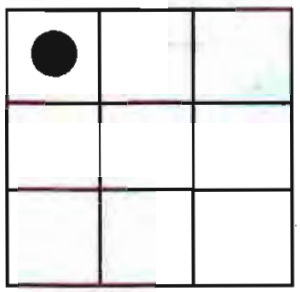
§ 9. МУРЗИК И РОБОТ КОЛЯ



Занавес медленно поднимается. На сцене — огромный светящийся экран, изображающий Шахматную Страну.

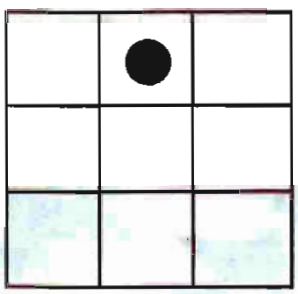
Алина. Какая красивая игрушка!

Папа Циркуль. С помощью этой игрушки мы научимся командовать роботом Колей. Смотрите внимательно и хорошенько запоминайте. Я помещаю Колю в одну из клеток и передаю слово Мурзику. Мурзик, начинай командовать!



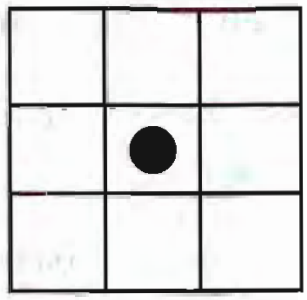
Мурзик. Шагни вправо.

Папа Циркуль. Смотрите все! Коля сделал ровно один шаг вправо. Получилось вот что:

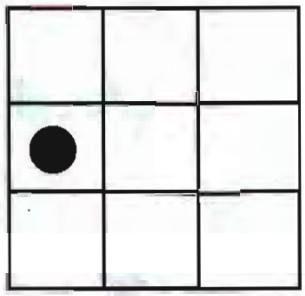


Мурзик. Шагни вниз.

Папа Циркуль. Следите за картинкой! Коля делает то, что приказано.



Мурзик. Шагни влево.



Мурзик. Иди вверх. Ой! А почему он не слушается?

Папа Циркуль. Потому что ты перепутал слова. Вместо «Шагни вверх» сказал «Иди вверх».

Мурзик. Ну и что? Ведь смысл команды не изменился!

Папа Циркуль. Это для тебя не изменился. А для робота очень даже изменился.

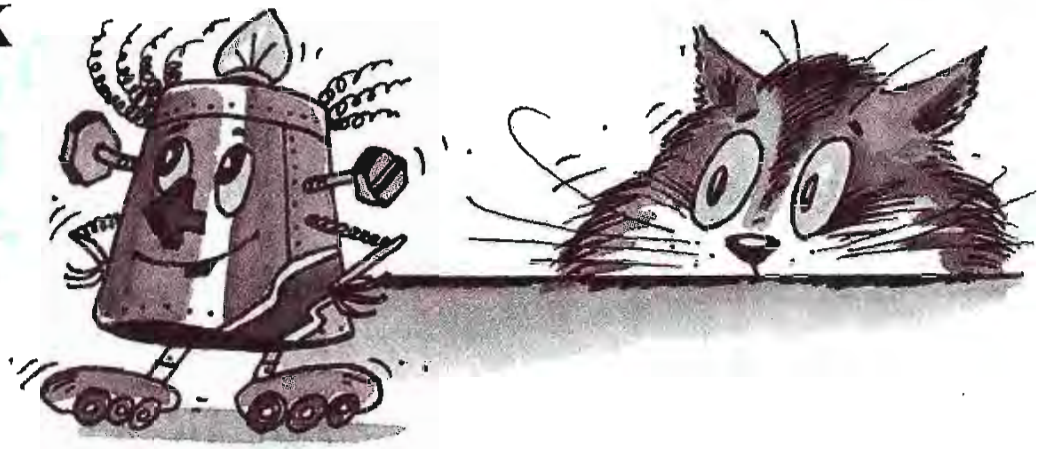
Мурзик. Почему?

Папа Циркуль. Робот понимает только те команды, которые входят в его репертуар. Причем эти команды нужно писать очень точно. Если ты перепутаешь всего одну букву, он тебя не поймет и «забастует». Исправь ошибку и скажи команду правильно.

Мурзик. Шагни вверх.

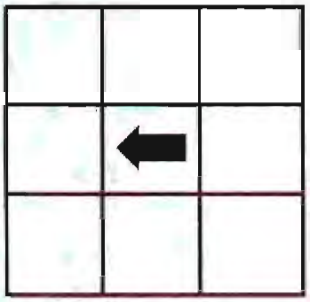
Папа Циркуль. Вот видишь, все в порядке! Робот узнал команду и выполнил ее.

§ 10. МУРЗИК И РОБОТ СТЕПА

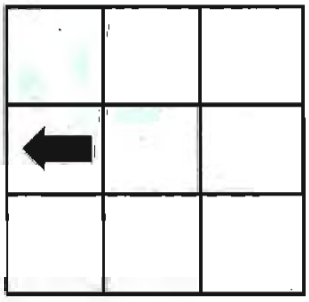


Мурзик. Ну ладно, с Колей все ясно. Теперь я хочу поиграть со Степой.

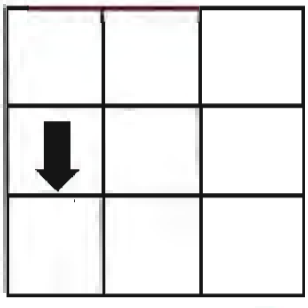
Папа Циркуль. Изволь. Я помещаю Степу в центр экрана носом влево. Начинай командовать.



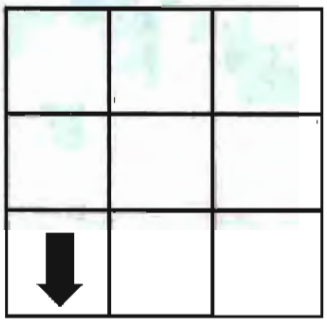
Мурзик. Шагни вперед.



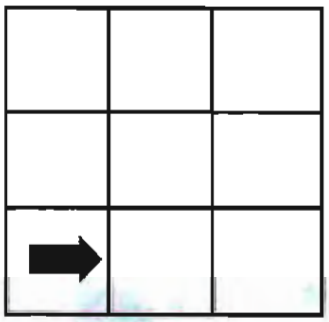
Мурзик. Повернись налево.



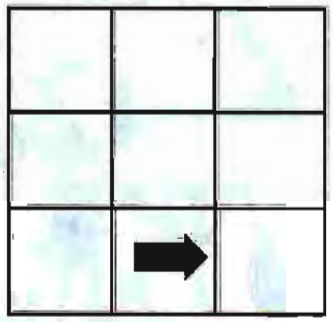
Мурзик. Шагни вперед.



Мурзик. Повернись налево.

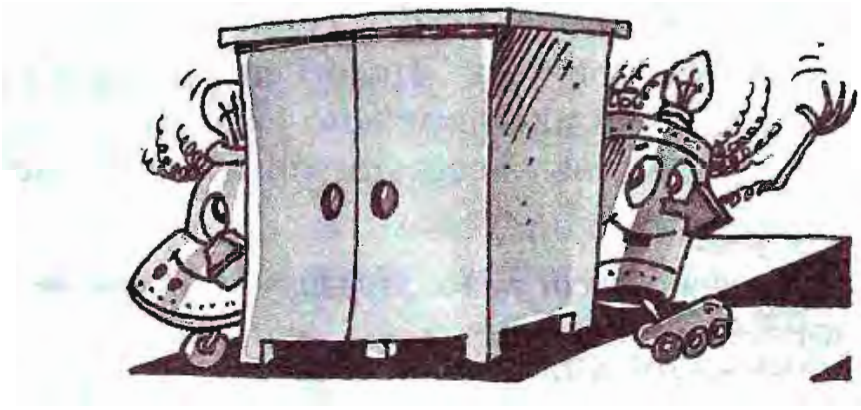


Мурзик. Шагни вперед.



Папа Циркуль. Молодец, Мурзик!
Ты командуешь как настоящий генерал.

§ 11. АЛГОРИТМ «ИГРА В ПРЯТКИ»



— Давайте будем играть в прятки.

В одной из клеток Шахматной Страны стоит шкаф (рис. 10). Робот находится перед шкафом в клетке А. Нужно написать алгоритм, помогающий роботу спрятаться за шкафом, т. е. перейти на клетку В.

Рис. 10. Посредине Шахматной Страны стоит шкаф

Решение задачи показано на рис. 11 (для робота Коли) и рис. 12 (для робота Степы).

Рис. 11. Коля прячется за шкафом

Рис. 12. Степа прячется за шкафом

Алина. По-моему, рисунки 11 и 12 совсем никудышные.

Папа Циркуль (*обиженным голосом*). Это еще почему?

Алина. Потому что из них ничего не видно. Я не вижу Шахматную Страну. Не понимаю, где стоит шкаф. Не вижу, откуда и куда бегают роботы. Мурзик, дружочек! Нарисуй мне такую картинку, чтобы все было понятно.

Мурзик (*чешет затылок*). Ох уж эти женские капризы! Как же ей угодить? Ага, придумал! Надо использовать икону «Комментарий». Внутри этой удобной иконы можно рисовать любые картинки и ответить на все вопросы Алины. Результат показан на рис. 13 (для Коли) и рис. 14 (для Степы).

Алина (*бросается к Мурзику и целует его*). Какой же ты умница! Теперь все стало понятно.

Вопросы Саши Ехидного

1. Какой репертуар у робота Коли?
2. Какой репертуар у робота Степы?
3. Как называется первая команда, которую выполнил Коля (рис. 13)? В какой клетке он будет находиться после первой команды? После второй? После третьей?
4. Как называется первая команда, которую выполнил Степа (рис. 14)? В какой клетке он будет находиться после первой команды? После второй? После четвертой?
5. Сравнивая рис. 13 и 14, скажи: кому пришлось выполнить больше команд — Коле или Степе? Почему?
6. Сколько икон на рис. 13? Как они называются? Есть ли среди них одинаковые? Какие?

Задачи Миши Проверялкина

1. Напиши алгоритм, заставляющий Степу, не выходя из клетки, повернуться кругом и занять прежнее положение.
2. В какую сторону смотрит Степа после выполнения алгоритма на рис. 14? Напиши алгоритм, заставляющий его оглянуться.
3. После выполнения алгоритма на рис. 14 Степа застыл на месте. Напиши алгоритм, заставляющий его вернуться в клетку А по своим следам и занять прежнюю позицию.
4. Напиши алгоритм, заставляющий Степу вернуться из клетки В в клетку А, обойдя шкаф с другой стороны.
5. Заставь робота Колю (когда он находится в клетке А) обойти вокруг шкафа по часовой стрелке и вернуться в клетку А.
6. Заставь робота Колю (когда он находится в клетке В) обойти вокруг шкафа против часовой стрелки и вернуться в клетку В.

Мурзик (*обиженно и возмущенно*). Кто такой Саша Ехидный? Откуда взялся этот, как его, Проверялкин? Почему они здесь командуют?

Папа Циркуль (*ласково*). Не кипятись! Это мои помощники. Они делают важное дело — придумывают умные вопросы и хитрые задачи, чтобы ты мог проверить свои знания и проявить смекалку. Слушайся их и будь настойчив. Постарайся понять все вопросы и выполнить все задания. Сумеешь — честь тебе и хвала! Не сумеешь — повтори материал еще раз.

Мурзик (*капризничает*). Начинается! Повторенье — мать ученья. На дворе кол, на колу мочало, начинай читать книжку сначала! И сколько раз я должен учить одно и то же?

Папа Циркуль. Пока не разгадаешь все загадки Саши Ехидного и Миши Проверялкина.

Дружок, вставай! Дружок, проснись!
Садись за стол и не ленись.
Задачки станем мы решать
И на вопросы отвечать!

Робот Коля прячется за шкафом

Дано

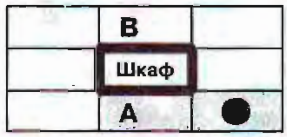
Робот Коля умеет исполнять команды:
«Шагни вправо», «Шагни вверх», «Шагни влево».
Коля (обозначенный кружком) находится в клетке А.

Надо

Спрятать Колю за шкафом, т.е. переместить его из клетки А в клетку В.

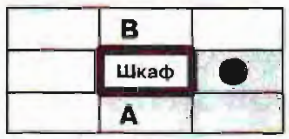
Шагни вправо

Коля шагнул вправо и очутился в правом нижнем углу



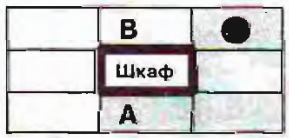
Шагни вверх

Коля шагнул вверх



Шагни вверх

Коля сделал еще один шаг вверх и очутился в правом верхнем углу



Шагни влево

Задача решена, потому что Коля пришел в клетку В, что и требовалось

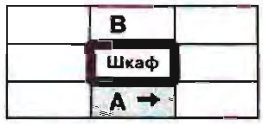
Конец

Рис. 13. Коля прячется за шкафом. Алгоритм с комментариями (сравни с рис. 11)

Робот Степа прячется за шкафом

Дано

Робот Степа умеет исполнять команды: «Шагни вперед», «Повернись налево».
Степа (обозначенный стрелкой) находится в клетке А.

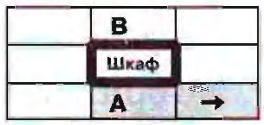


Надо

Спрятать Степу за шкафом, т. е. переместить его из клетки А в клетку В.

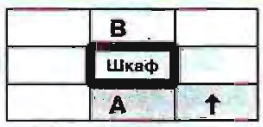
Шагни вперед

Степа шагнул вперед и очутился в правом нижнем углу



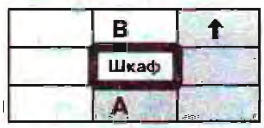
Повернись налево

Степа, стоя на месте, повернулся налево



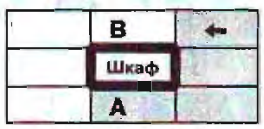
Шагни вперед
Шагни вперед

Степа сделал два шага вперед и очутился в правом верхнем углу



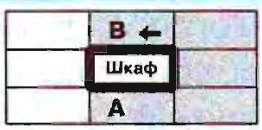
Повернись налево

Степа, стоя на месте, повернулся налево



Шагни вперед

Задача решена, потому что Степа пришел в клетку В, что и требовалось.



Конец

Рис. 14. Степа прячется за шкафом. Алгоритм с комментариями (сравни с рис. 12)

§ 12. ЗАЧЕМ НУЖНА ИКОНА «КОММЕНТАРИЙ»?

На рис. 13 и 14 много раз используется икона «Комментарий». Если убрать все комментарии, алгоритмы приобретут вид, показанный на рис. 11 и 12. Что же меняется в результате исключения комментариев? Отвечая на этот вопрос, робот Коля мог бы сказать примерно так: «Для меня и моего друга Степы не меняется ничего. Читая алгоритм, мы исполняем только команды и не обращаем на комментарии никакого внимания. Нам, роботам, все эти ваши комментарии — как мертвому припарки. Нам от них ни горячо ни холодно».

Хлястик. В таком случае для кого пишутся комментарии?

Папа Циркуль. Для людей, которые читают алгоритмы. Зачастую алгоритмы трудно понять, поэтому в сложных случаях без комментария невозможно разобраться в существе вопроса.

Хлястик. А нельзя ли привести пример?

Папа Циркуль. В молодости я работал в одном институте, который проектировал алгоритмы для компьютера, управляющего большой электростанцией. В нашей лаборатории выделялся талантливый ученый Валька Силаков. Он как раз и придумал самый главный и самый сложный алгоритм, но по вредности характера никогда не писал комментарии. Помню, начальник лаборатории все время охал: «Ой, Силаков, доведешь ты нас до беды!» Как в воду глядел. Вскоре Силаков уволился, а еще через месяц в его алгоритме обнаружилась ошибка. Кинулись мы разбираться, а ничего понять не можем. Алгоритм очень сложный, документации никакой — поди сообрази, где прокол. Попробовали Силакова найти, а того и след простыл —в другой город уехал. Ищи ветра в поле! Так, веришь или нет, мы впятером целый месяц тот проклятый алгоритм, как ребус, расшифровывали. Как послание инопланетян! Изрядно голову поломали! В общем, подложил нам Силаков большую свинью. А если бы он не ленился и писал комментарии, мы бы нашли ошибку намного раньше и не потратили впустую столько сил.

Что такое комментарии	Это различные пояснения, облегчающие понимание алгоритма
Почему исполнители не читают комментарии	Дело исполнителя — выполнять команды. А комментарий — это не команда
Для кого пишут комментарии	Для человека, который читает алгоритм и стремится его понять
Запомни	Алгоритм без комментариев — это мина замедленного действия
Вредный совет	Если хочешь подложить свинью тем, кто без тебя будет разбираться в твоем алгоритме, пиши его без комментариев

АЛГОРИТМЫ С РАЗВИЛКАМИ — ЭТО ОЧЕНЬ ИНТЕРЕСНО!

§ 13. ДАВАЙТЕ ПУТЕШЕСТВОВАТЬ ПО ДРАКОН-СХЕМЕ, КАК ПО ЖЕЛЕЗНОЙ ДОРОГЕ!

Папа Циркуль. Предлагаю сыграть в новую игру.

Мурзик. Какую?

Папа Циркуль. Будем путешествовать по алгоритму, как по железной дороге. Посмотри на рис. 15. На нем изображена дракон-схема, рассказывающая, как девочка Лена собирается в школу. А теперь вообрази, что вертикальная линия — это рельсы, а каждая икона — станция. Сверху находится станция отправления (которая называется «Утро школьницы Лены»), внизу — станция назначения под названием «Конец». Правда, есть

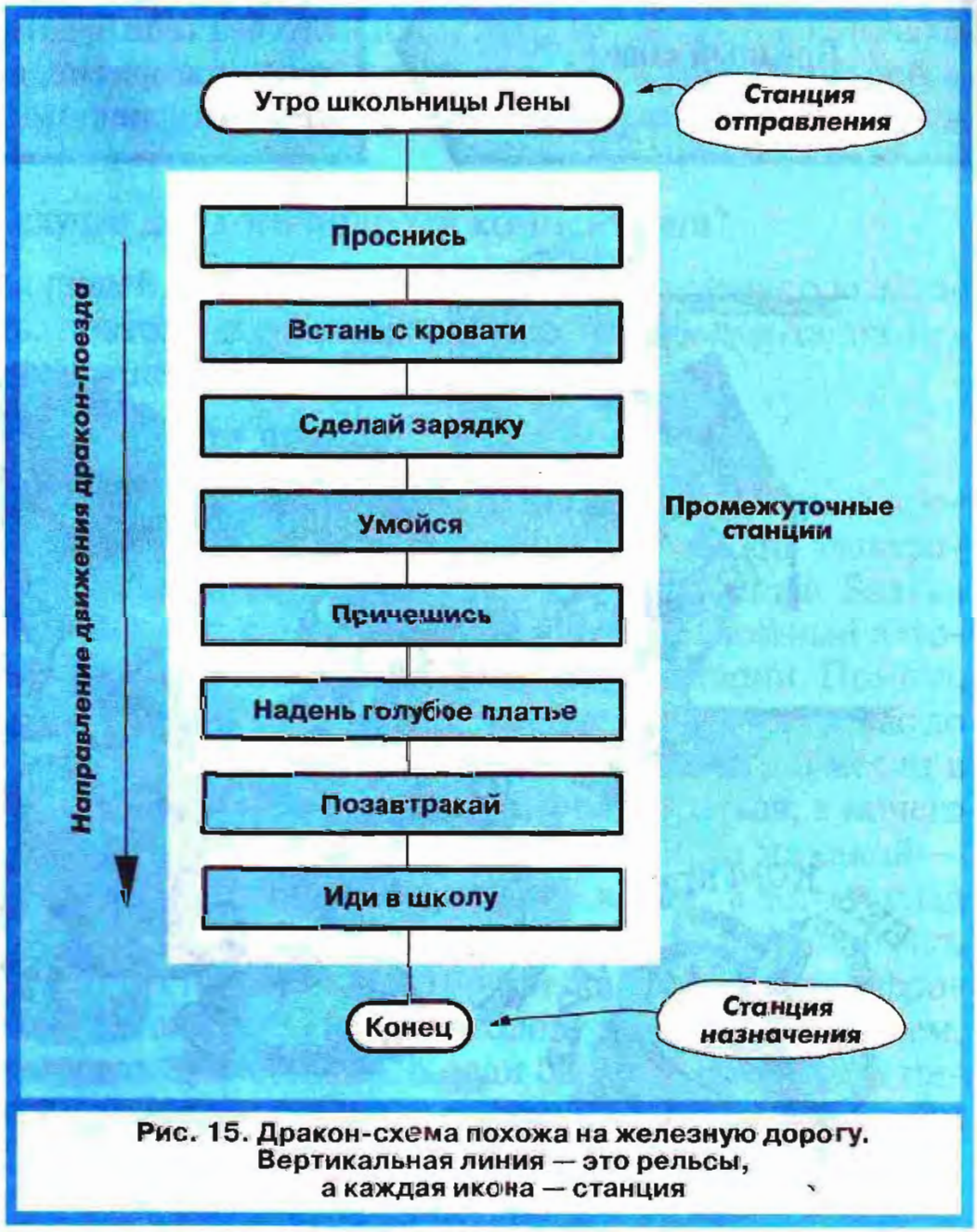


Рис. 15. Дракон-схема похожа на железную дорогу. Вертикальная линия — это рельсы, а каждая икона — станция

хитрость: по алгоритмической железной дороге поезда ходят только в одну сторону — сверху вниз: от иконы «Заголовок» к иконе «Конец».

Мурзик. А обратно нельзя?

Папа Циркуль. Обратно нельзя. Потому что бессмысленно. Хочешь убедиться — прочитай команды снизу вверх — от конца к началу.

Мурзик (*нумерует команды задом наперед и читает*). 1. Иди в школу. 2. Позавтракай... М-да... И вправду чепуха какая-то. Получается, что Лена сначала пришла в школу, а потом завтракает. Но ведь на самом деле она завтракает дома и только после этого идет в школу.

Я понял свою ошибку. Обратно действительно нельзя. Если читать снизу вверх, получается, что время движется в обратную сторону. Как в кино, когда пленку крутят назад.

Папа Циркуль. Дорога от начала до конца алгоритма называется «дракон-дорога». По этой дороге едет воображаемый поезд — дракон-поезд. Он идет от станции «Заголовок» до станции «Конец». В поезде один-единственный пассажир, который называется...

Мурзик. Исполнитель!

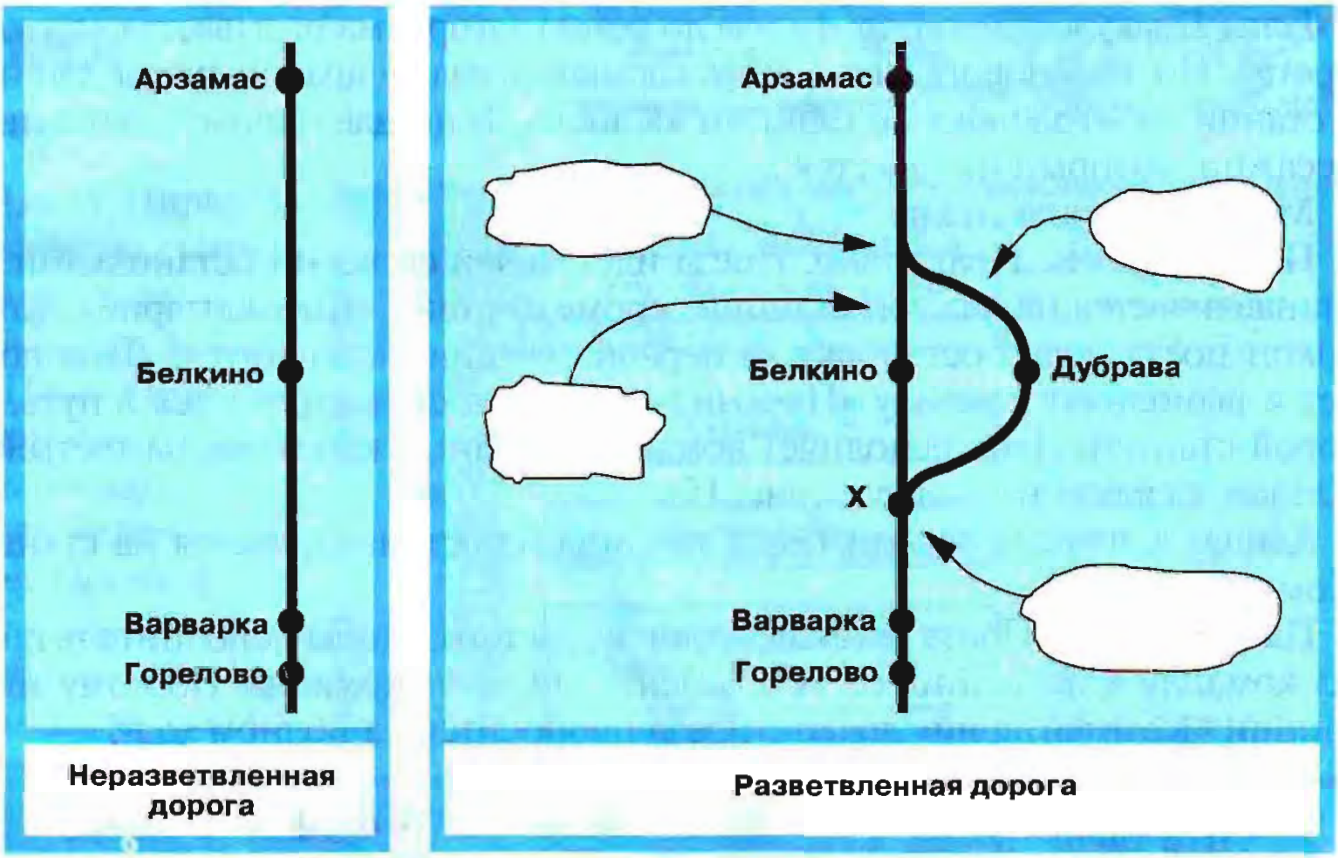
Папа Циркуль. Правильно. Поезд идет почти со всеми остановками: он останавливается на каждой станции, кроме станции «Комментарий». Когда дракон-поезд делает остановку на первой станции, исполнитель Лена получает и выполняет команду «Проснись», и поезд снова трогается в путь. На второй станции Лена исполняет команду «Встань с кровати», на третьей — «Сделай зарядку» и так далее (рис. 15).

Алина. А почему дракон-поезд никогда не останавливается на станции «Комментарий»?

Папа Циркуль. Поезд останавливается для того, чтобы исполнитель получил команду и выполнил ее. А комментарий — не команда. Поэтому мимо станции «Комментарий» дракон-поезд проносится на полном ходу.

Что такое дракон-дорога	**Это графический путь на дракон-схеме, идущий от начала алгоритма (икона «Заголовок») до его конца (икона «Конец»)**
Что такое дракон-поезд	**Это воображаемый поезд, следующий по дракон-дороге от начала до конца, при движении которого выполняются команды алгоритма**
Как показать движение дракон-поезда по дракон-схеме	**Для этого ученик обычно использует собственный палец (или конец указки), которым он прослеживает путь от начала до конца алгоритма**

§ 14. ЧТО ТАКОЕ ЛИНЕЙНЫЙ АЛГОРИТМ? ЧТО ТАКОЕ РАЗВЕТВЛЕННЫЙ АЛГОРИТМ?



а *б*

Рис. 16. Неразветвленная и разветвленная дорога

Дороги бывают неразветвленные (рис. 16*а*) и разветвленные (рис. 16*б*). Алгоритмы похожи на дороги: они тоже бывают неразветвленные (рис. 17*а*) и разветвленные (рис. 17*б*). Если в алгоритме нет ни одного разветвления, он называется линейным.

Мурзик. Почему такое название?

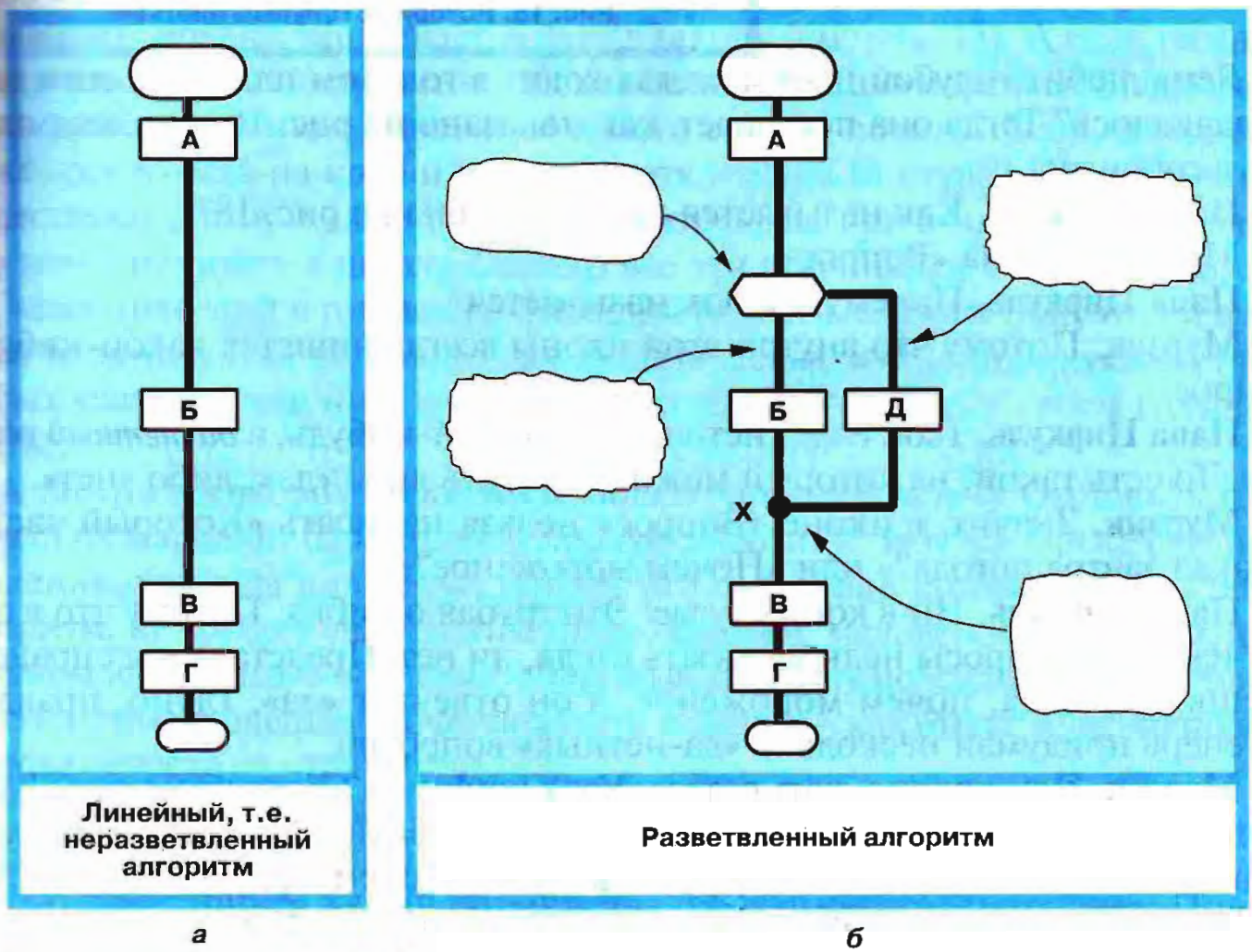
Папа Циркуль. Потому что у линейного алгоритма дракон-дорога прямая, как линейка. Она изображается как прямая вертикальная линия, которая проходит через все иконы алгоритма. Кстати, ты знаешь, как делают шашлык?

Мурзик. По телику видел. Берут такую длинную металлическую иглу — шампур — и нанизывают на нее кусочки мяса.

Папа Циркуль. А теперь посмотри на рис. 17*а*. Это линейный алгоритм. Видишь? Вертикальная линия — это шампур. Только вместо кусочков мяса на него нанизаны команды алгоритма.

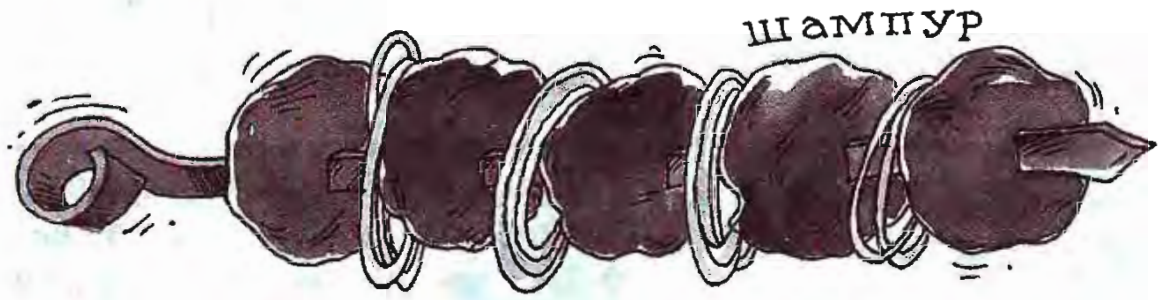
Мурзик. Но ведь дороги не только расходятся, но и сходятся, сливаются.

Папа Циркуль. Конечно. Например, на рис. 16*б* показано слияние двух дорог в точке X. В алгоритмах тоже бывают слияния. На дракон-схеме они изображаются двумя линиями, которые встречаются под прямым углом (см. точку X на рис. 17*б*).



а *б*

Рис. 17. Линейный и разветвленный алгоритмы



§ 15. ЗАЧЕМ НУЖНА ИКОНА «ВОПРОС»?

Рис. 18. История с голубым платьем

Лена любит голубой цвет и всегда ходит в голубом платье. А если оно испачкалось? Тогда она поступает, как показано на рис. 18: надевает платье в горошек.

Папа Циркуль. Как называется верхняя икона на рис. 18?

Мурзик. Икона «Вопрос».

Папа Циркуль. Почему она так называется?

Мурзик. Потому что внутри этой иконы всегда пишется какой-нибудь вопрос.

Папа Циркуль. Твой ответ неточен. Не какой-нибудь, а *да-нетный* вопрос. То есть такой, на который можно ответить либо «да», либо «нет».

Мурзик. Значит, в иконе «Вопрос» нельзя написать «Который час?», «Какая завтра погода?» или «Почем мороженое?»?

Папа Циркуль. Ни в коем случае! Это грубая ошибка. Потому что в ответ на такие вопросы нельзя сказать ни да, ни нет. Представь, ты спрашиваешь продавца, почем мороженое, а он отвечает «да». Глупо, правда? А теперь придумай несколько «да-нетных» вопросов.

Мурзик. Пожалуйста, хоть сотню. Урок кончился? Утюг сломался? Вася купил хлеб? Поезд пришел? Преступника арестовали? «Спартак» выиграл? Майка желтая? Эта лужа больше, чем та? На улице температура выше нуля?

Алина. А я так и не поняла, зачем нужна икона «Вопрос».

Папа Циркуль. Бывают ситуации, когда нужно выбрать одно действие из двух. Тут без иконы «Вопрос» никак не обойтись. При ответе «нет» выполняется одно действие, при ответе «да» — другое. Икона «Вопрос» нужна, чтобы сделать в алгоритме развилку. В этом легко убедиться, взглянув на рис. 18.

Что такое да-нетный вопрос

- **Это вопрос, на который можно ответить либо «да», либо «нет»**
- **Все другие ответы запрещены**

§ 16. КАКОЙ АЛГОРИТМ ТОЧНЕЕ ОПИСЫВАЕТ ПОВЕДЕНИЕ ШКОЛЬНИЦЫ ЛЕНЫ: ЛИНЕЙНЫЙ ИЛИ РАЗВЕТВЛЕННЫЙ?

Девочка Лена из § 13 ведет себя очень похвально. При всех условиях она делает зарядку, завтракает и идет в школу (см. рис. 15). А если пожар, наводнение или землетрясение? Если внезапная атака марсиан? Если кошке дверью прищемило хвост и ее нужно срочно лечить? Если, наконец, Лена просто села на клей и не может отклеиться от стула? Как известно, существуют сотни и даже тысячи уважительных причин, чтобы опоздать или вовсе не пойти в школу. Однако все эти причины, в том числе пожары, землетрясения и происки барабашки не оказывают на Лену ни малейшего влияния. Она действует как механическая заведенная кукла и при любых условиях как ни в чем не бывало явится в школу в своем голубом платье.

А теперь поговорим серьезно и заодно упростим задачу. Отставить мифических марсиан, барабашек и прочую экзотику! Забудем также про наводнения, ураганы и пожары, потому что цель, ради которой создан наш алгоритм, не требует учета таких, прямо скажем, маловероятных событий. Оставим лишь самые обычные, знакомые каждому повседневные проблемы. Например: очень не хочется делать зарядку — как быть? Куда-то подевалась расческа — что делать?

На рис. 15, к сожалению, не отражены подобные вещи. В результате Лена ведет себя как запрограммированный робот, с железной пунктуальностью движущийся к цели по единственно возможному маршруту. Почему так получилось? Да потому что сочинитель придумал для нее линейный алгоритм, а он не учитывает НИКАКИХ внешних условий, событий и происшествий. В самом деле, алгоритмическая железная дорога на рис. 15 изображается одной-единственной вертикальной линией, которая не имеет ни развилок, ни объездных путей.

Однако в реальной жизни линейные алгоритмы встречаются довольно редко. Гораздо чаще попадаются алгоритмы со множеством развилок. Поэтому алгоритмический рассказ «Утро школьницы Лены» в действительности содержит большое число развилок, некоторые из которых показаны на рис. 19.

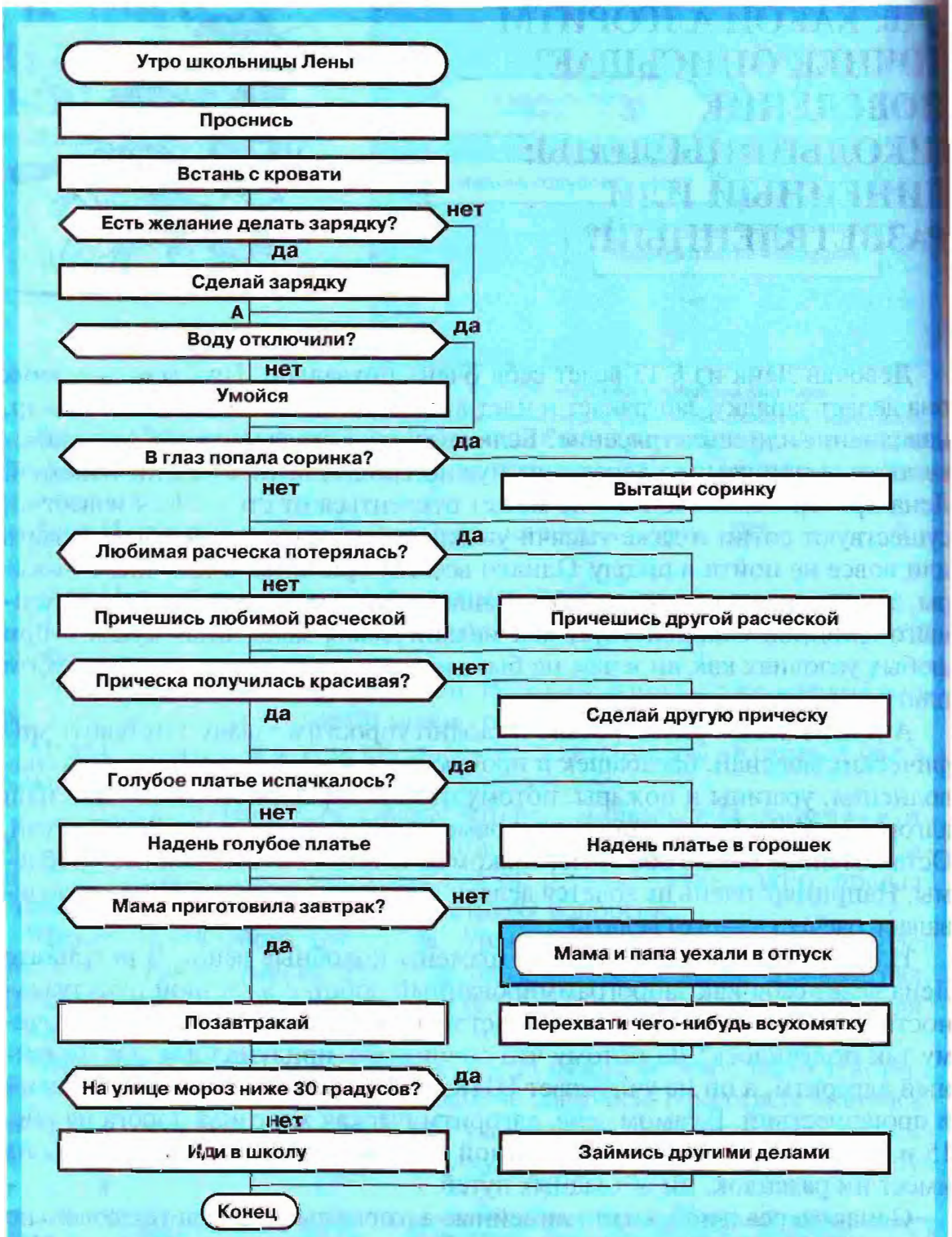


Рис. 19. Разветвленный алгоритм, полученный из линейного алгоритма на рис. 15

Задачи Миши Проверялкина

1. Исправь алгоритм на рис. 19: выброси из него развилки, которые тебе кажутся необязательными.
2. Добавь в алгоритм на рис. 19 новые развилки, позволяющие описать поведение Лены более подробно. Не бойся ошибиться. Пофантазируй.

§ 17. ЧЕМ ОТЛИЧАЕТСЯ ИКОНА «ВОПРОС» ОТ ИКОНЫ «ДЕЙСТВИЕ»?

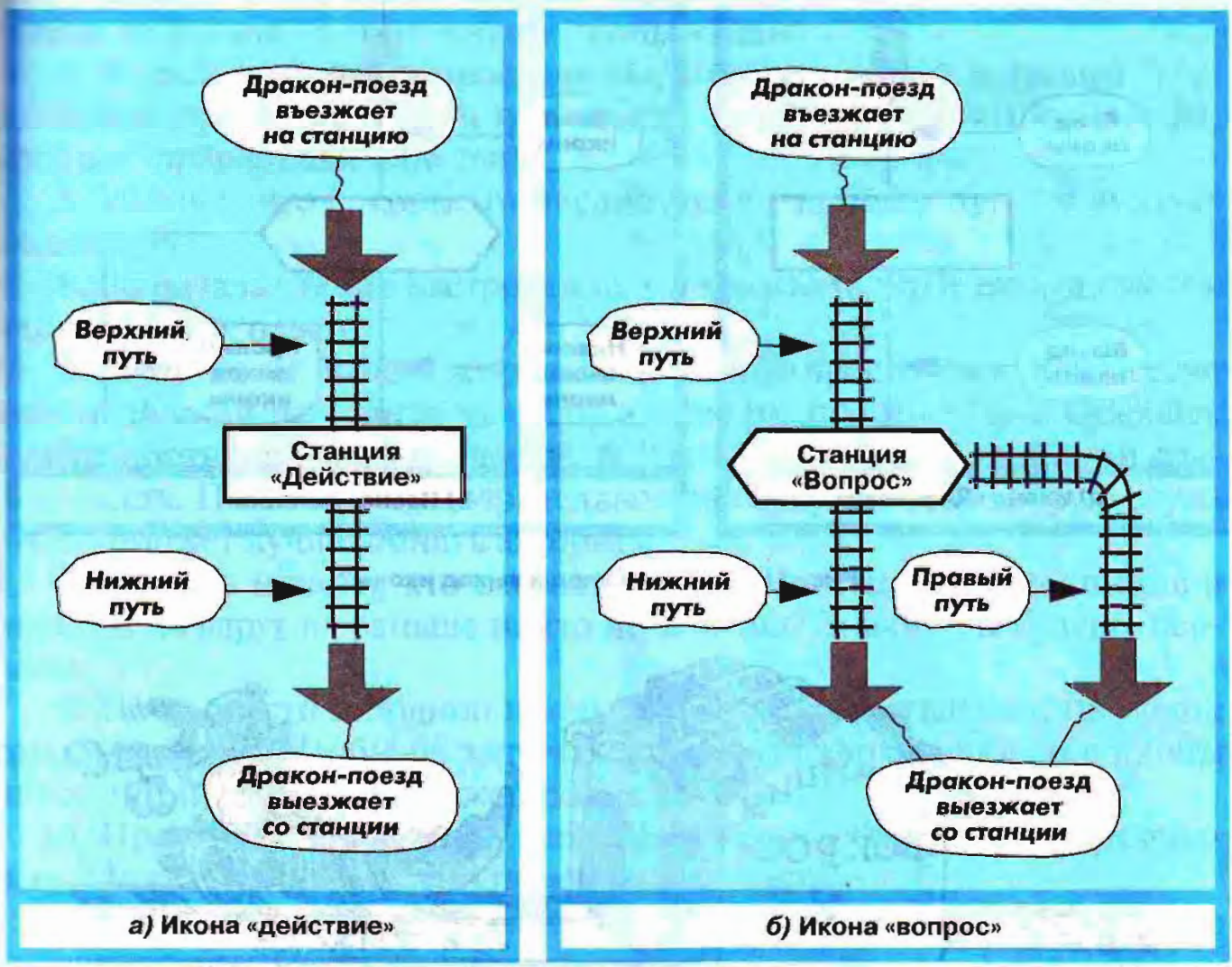
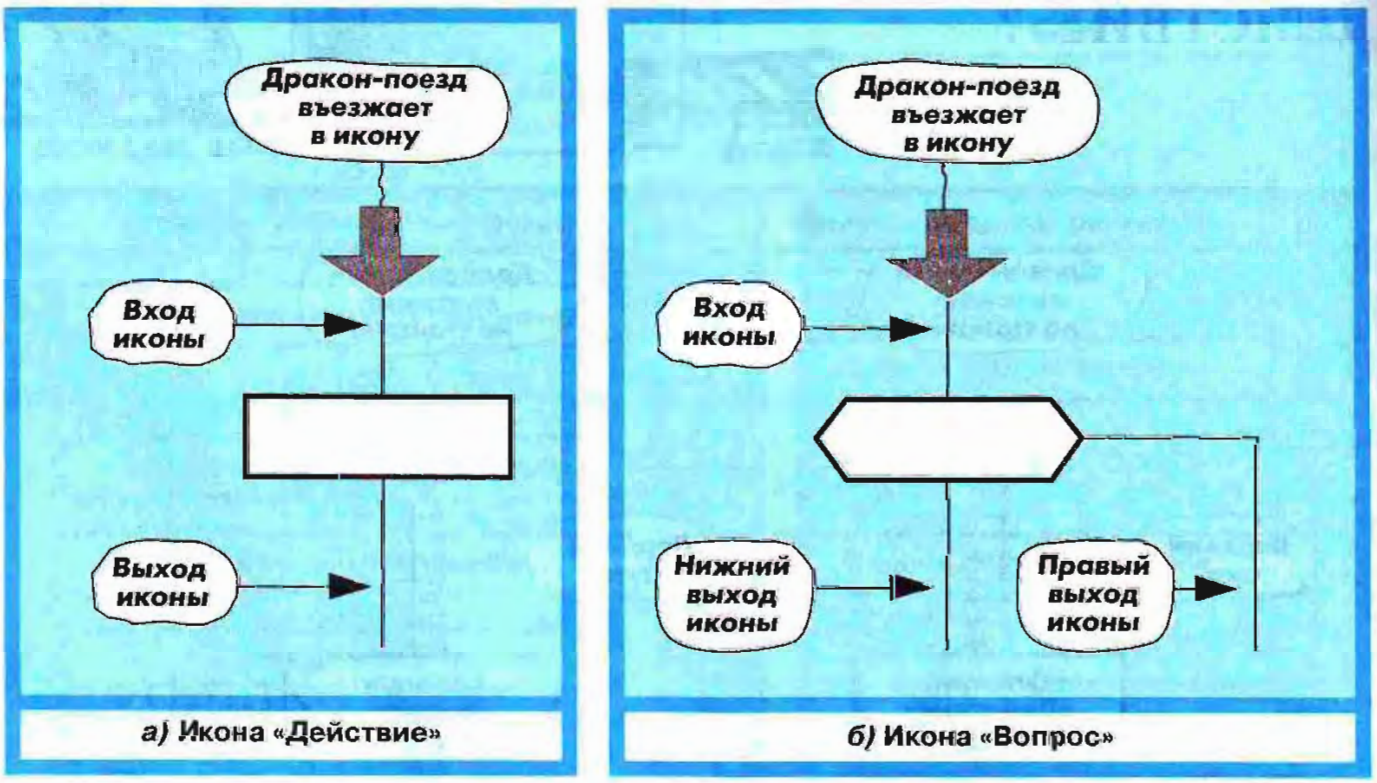


Рис. 20. Чем отличается икона «Действие» от иконы «Вопрос»?

Мы уже знаем, что икону можно сравнить с железнодорожной станцией. К станции «Действие» подходят две железнодорожные колеи: верхний путь и нижний путь (рис. 20*а*). Дракон-поезд едет через икону «Действие» сверху вниз: по верхнему пути он въезжает на станцию, по нижнему — выезжает.

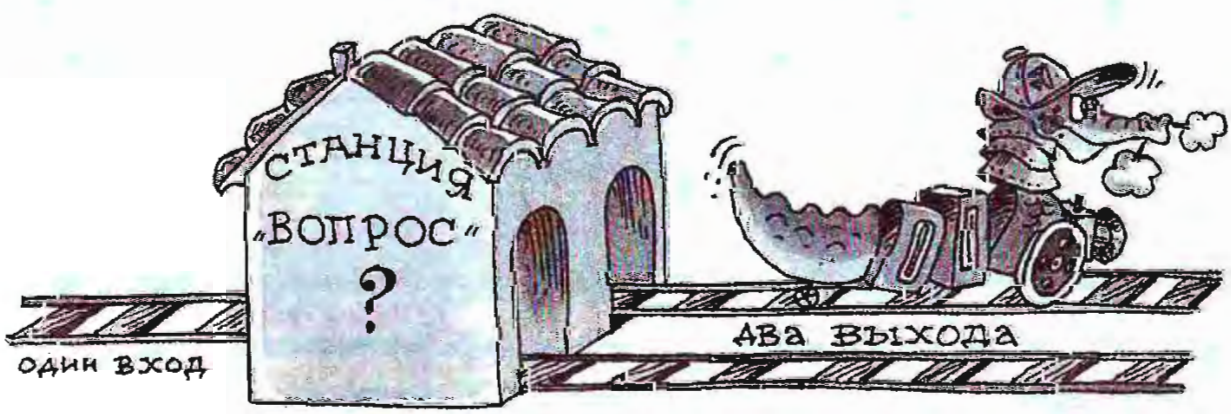
К иконе «Вопрос» подходят не две, а три колеи: верхний, нижний и правый путь (рис. 20*б*). Дракон-поезд въезжает на станцию «Вопрос» через верхний путь, а выехать может либо через нижний, либо через правый путь. Таким образом, икона «Вопрос» выполняет роль железнодорожной стрелки: если стрелочник переведет стрелку направо, дракон-поезд поедет через правый путь, если вниз — через нижний.

Верхний путь называется входом иконы, нижний и правый путь — это выходы (рис. 21). Можно сказать, что дракон-поезд въезжает в икону через вход, а выезжает через выход.



а) Икона «Действие» *б)* Икона «Вопрос»

Рис. 21. Что такое вход и выход иконы?



§ 18. КАК ЧИТАТЬ АЛГОРИТМЫ

Алгоритм читают совсем не так, как рассказ или газету. Читая обычную книгу, мы обращаем внимание лишь на то, что нам интересно, остальное можем и пропустить. Однако при чтении алгоритма надо поступать иначе.

Правила чтения алгоритма

1. Пойми, зачем нужен данный алгоритм, какова его цель.
2. Вникни в смысл каждой команды и каждого вопроса. Если что-то непонятно, разберись сам или попроси помощи.
3. Читай алгоритм в том же направлении, в каком движется дракон-поезд: от иконы «Заголовок» до иконы «Конец».
4. Встретив развилку, проследи оба выхода: нижний и правый. Убедись, что при ответе «да» и при ответе «нет» алгоритм выполняет разумные, правильные действия.
5. Убедись, что боковые пути сливаются с прямым путем в нужных точках.
6. До начала чтения настройся на то, что в алгоритме есть ошибки и ты должен их найти.
7. Никогда не читай алгоритм бездумно, «просто так», неизвестно зачем. Всякий раз, когда читаешь алгоритм, проверяй его. Старайся найти противоречия, нелепости, дефекты, упущения, ошибки и слабые места. Подобная «подозрительность» сослужит тебе добрую службу и поможет лучше понять алгоритм.
8. Не верь никому, кто скажет, что в алгоритме нет недостатков и ошибок. А вдруг их раньше никто не заметил? Значит, ты будешь первым.
9. Не пропусти ни одной иконы, ни одного ответвления, ни одной точки слияния. Чтобы не запутаться, отмечай карандашом все иконы и все линии, которые ты уже проверил.
10. Проверь, правильно ли расставлены слова «да» и «нет» в развилках. Может быть, их нужно поменять местами?
11. Заканчивая чтение, убедись, что алгоритм решает поставленную задачу.

§ 19. МУРЗИК УЧИТСЯ ЧИТАТЬ АЛГОРИТМЫ

Папа Циркуль. Мурзик, прочитай алгоритм на рис. 19.

Мурзик (*касается карандашом иконы «Заголовок»*). Алгоритм называется «Утро школьницы Лены». (*Передвигает карандаш на следующую икону*.) Первая команда: «Проснись». (*Далее Мурзик на протяжении всего рассказа непрерывно водит карандашом по линиям, всякий раз указывая на ту икону, о которой он в данный момент говорит.*) Вторая команда: «Встань с кровати».

Затем идет развилка «Есть желание делать зарядку?». При ответе «да» выполняется команда «Сделай зарядку». При ответе «нет» действия отсутствуют. Все правильно: никто не делает зарядку, если не хочется.

После этого попадаем на развилку «Воду отключили?». Если «нет», выполняется команда «Умойся». Если «да» (вода не льется), действия отсутствуют. Правильно, потому что нечем умываться.

Далее следует развилка «В глаз попала соринка?». Если «нет», действия отсутствуют, потому что все в порядке. Если «да», выполняется команда «Вытащи соринку».

Затем идет развилка «Любимая расческа потерялась?». Если «нет» (расческа на месте), выполняется команда «Причешись любимой расческой». Если «да» (расческа пропала), следует команда «Причешись другой расческой». Что ж, вполне логично. Ну и так далее.

(Подражая Мурзику, дочитай алгоритм до конца.)

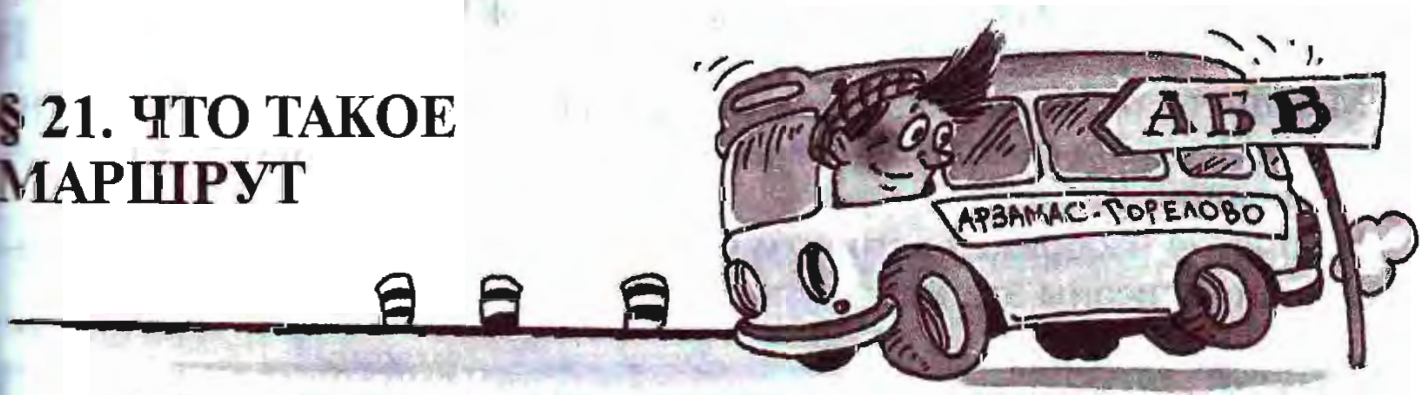
§ 20. ЧТО ТАКОЕ БУКВЕННАЯ ДРАКОН-СХЕМА?

На рис. 22*а* изображена дракон-схема «Охота на мамонта». Заменим текст внутри икон буквами. Вместо «Охота на мамонта» напишем букву А, вместо «Поймай мамонта» — букву Б и т. д. В результате получим буквенную дракон-схему на рис. 22*б*.

Рис. 22. Как преобразовать обычную дракон-схему в буквенную?

Зачем мы это сделали? Надписи в иконах алгоритма хороши тем, что позволяют понять его смысл. Однако в следующем параграфе мы должны научиться решать новую задачу: описывать маршруты в алгоритмах. Для описания маршрутов надписи неудобны: они слишком длинны. Поэтому мы и перешли от слов к буквам — к буквенным дракон-схемам.

§ 21. ЧТО ТАКОЕ МАРШРУТ



«Внимание! Автобус следует по маршруту: Арзамас — Белкино — Варварка — Горелово».

На рис. 16*а* изображена неразветвленная дорога, на ней всего один маршрут. Если же дорога раздваивается, число маршрутов увеличивается. На-

пример, на рис. 16*б* показано, что автобусы из Арзамаса в Горелово идут по двум маршрутам: один через Белкино, другой через Дубраву.

Маршрут 1. Арзамас — Белкино — Варварка — Горелово.

Маршрут 2. Арзамас — Дубрава — Варварка — Горелово.

С алгоритмами дело обстоит точно так же. Маршрут в алгоритме — это путь дракон-поезда, следующего из начала в конец алгоритма. Маршрут показывает, через какие иконы идет дракон-поезд. Чтобы описать маршрут, необходимо:

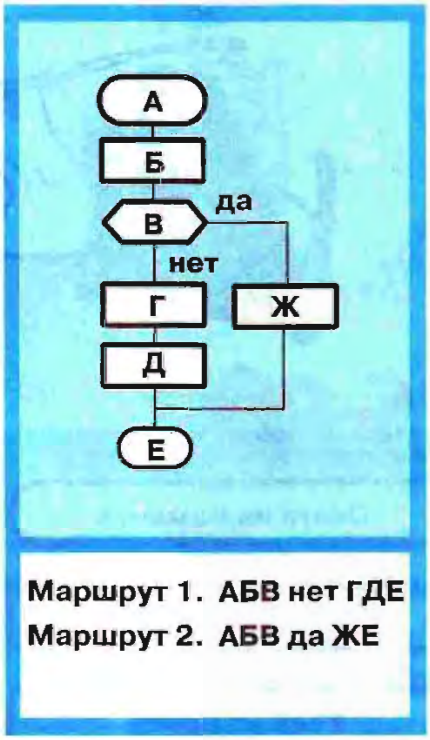


Рис. 23. Алгоритм с двумя маршрутами

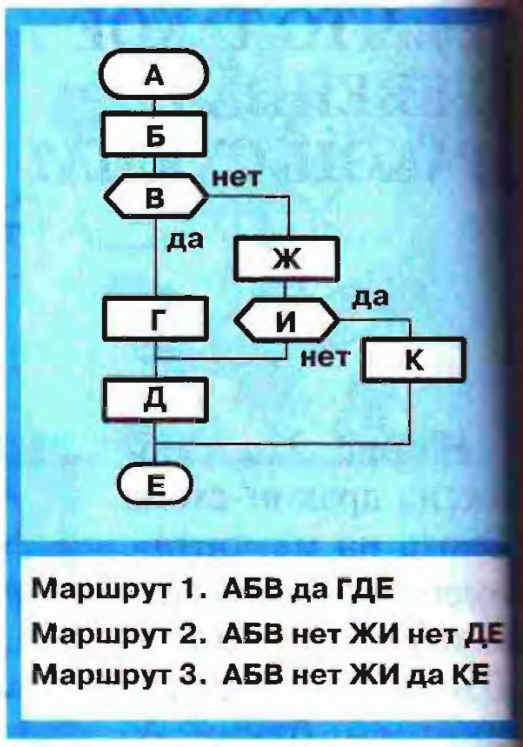


Рис. 24. Алгоритм с тремя маршрутами

- обозначить иконы буквами;
- записать последовательность букв (икон), через которые движется дракон-поезд;
- в нужных местах добавить слова «да» и «нет».

Линейный алгоритм на рис. 22*б* имеет всего один маршрут, который записывается так: АБВГД. На рис. 23 представлен алгоритм с двумя маршрутами:

Маршрут 1 — **АБВ нет ГДЕ**

Маршрут 2 — **АБВ да ЖЕ**

Маршрут 1 описывает движение сверху вниз по вертикали. Если же на развилке В дракон-поезд пойдет не вниз, а вправо (через икону Ж), получится маршрут 2.

На рис. 24 изображен алгоритм с тремя маршрутами. Проверьте, правильно ли записаны эти маршруты.

Что такое маршрут алгоритма

Это путь дракон-поезда, следующего по дракон-схеме от иконы «Заголовок» до иконы «Конец». Маршрут показывает, через какие иконы идет дракон-поезд

§ 22. ОТКУДА И КУДА ЕДЕТ ДРАКОН-ПОЕЗД?

У рыбы только одна голова и только один хвост. У алгоритма только одно начало и только один конец. Началом алгоритма является икона «Заголовок», концом — икона «Конец». Хотя у алгоритма может быть много маршрутов, все они имеют одно и то же начало — икону «Заголовок» и заканчиваются в одном и том же месте — в иконе «Конец».

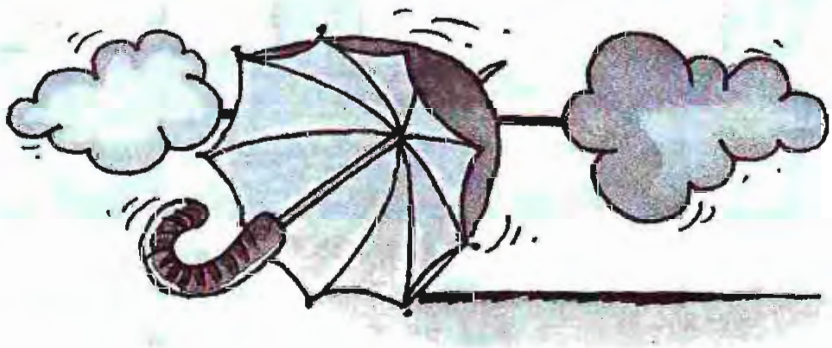
Правило

В алгоритмах все маршруты идут из начала в конец (из иконы «Заголовок» в икону «Конец»)

Задачи Миши Проверялкина

1. Сколько маршрутов на рис. 1, 4, 6, 7, 19?
2. Составь буквенную схему для рис. 7. Напиши формулы всех маршрутов.

§ 23. ОШИБКА, КОТОРАЯ НАЗЫВАЕТСЯ «ВИСЯЧИЙ ХВОСТ»



Мурзик. Наш поэт Хлястик сочинил алгоритмическое стихотворение:

Если тучи облепили горизонт,
Непременно захватите плащ и зонт.
Ну а ежели на улице жара,
Мы на пляж помчимся с криками «Ура!» (рис. 25*а*).

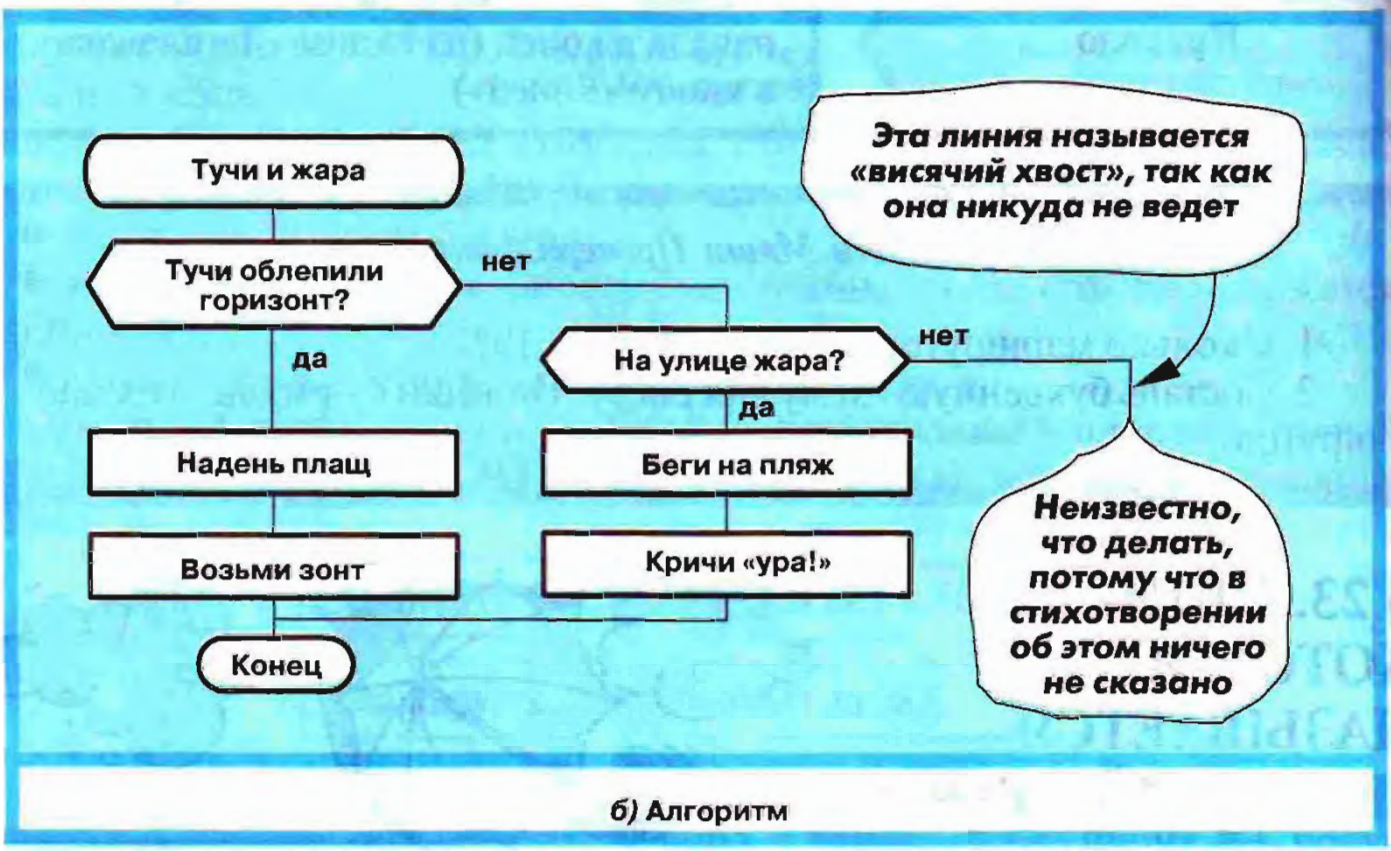
Папа Циркуль. Алгоритмическое, говоришь? А нет ли в этом алгоритме ошибок?

Мурзик. Разумеется, нет.

Можно ли превратить стихотворение в алгоритм?

Если тучи облепили горизонт,
Непременно захватите плащ и зонт.
Ну а ежели на улице жара,
Мы на пляж помчимся с криками «ура!»

а) Стихотворение



б) Алгоритм

Рис. 25. Алгоритмическая ошибка, которая называется «висячий хвост»

Папа Циркуль. Ты уверен?

Мурзик. Абсолютно уверен. Прекрасный алгоритм. Комар носа не подточит.

Папа Циркуль. Ну что ж, если это прекрасный алгоритм, без ошибок, попробуй начертить его дракон-схему.

Мурзик (*начинает рисовать схему на рис. 25б и вдруг кричит*). Ой! А почему эта линия оборвалась и повисла в воздухе? Здесь какая-то ошибка! Куда мне ее вести? Ничего не понимаю!

Папа Циркуль. Что ж, давай разберемся. На каком маршруте ты споткнулся?

Мурзик. Вот, пожалуйста (рис. 25*б*). Из иконы «Заголовок» идем в икону «Вопрос». Тучи облепили горизонт? Отвечаем «нет» и выходим направо. Попадаем во вторую икону «Вопрос». На улице жара? Отвечаем «нет» и снова выходим направо. А дальше-то куда идти? Получается, туч нет, сияет солнце, но и жары нет. Это может быть, например, зимой: температура минус десять, снегу по колено, все катаются на лыжах. А может быть и осенью: температура плюс десять, в лесу тьма грибников, полные корзины опят. Что же мне писать на дракон-схеме — «Катайся на лыжах» или «Иди за грибами»? В стихотворении Хлястика об этом ничего не сказано.

Папа Циркуль. Вот ты сам себе и ответил. Если не сказано — значит, это не алгоритм. Словесное описание только тогда можно назвать алгоритмом, если в нем сказано все, что нужно. Алгоритм должен четко и однозначно описывать действия ПРИ ЛЮБЫХ СОЧЕТАНИЯХ УСЛОВИЙ. А что получилось в стихотворении Хлястика? Что делать при сочетании условий «туч нет» и «жары нет»? Неизвестно. Поэтому ты и начинаешь гадать и фантазировать. То ли лыжи, то ли снег, то ли осень, то ли нет. Стихотворение Хлястика действительно немножко похоже на алгоритм, но это алгоритм с ошибкой. Ошибка заключается в том, что один из маршрутов обрывается и линия висит в воздухе, как хвост. Эта ошибка так и называется — «висячий хвост».

Что такое «висячий хвост»

Это ошибка, при которой один из маршрутов обрывается и не доходит до конца алгоритма

Запомни

В дракон-схеме не должно быть «висячих хвостов»

§ 24. МОЖНО ЛИ ОПИСЫВАТЬ АЛГОРИТМЫ СЛОВАМИ?

Мурзик. А почему нет? Конечно, можно. Вот, пожалуйста.

АЛГОРИТМ, ОПИСАННЫЙ СЛОВАМИ

Если я быстро сделаю уроки, пойду к Вовке. Если Вовка дома, побежим на футбол. Если билетов нет, перелезем через забор. Если нас поймают, пойдем в кино. Если фильм неинтересный, убежим с середины сеанса. Если в кино встретим Борьку, пойдем к нему в гости. Если Борькиной мамы нет дома, включим магнитофон на полную громкость. Вот будет здорово! Правда, если на шум прибегут соседи, придется удирать. Все. Конец алгоритма.

Папа Циркуль. Ох, Мурзик, ты снова сел в лужу. Но самое печальное, ты делаешь одни и те же ошибки. Смотри сюда — я начертил твой так называемый алгоритм (рис. 26). Полюбуйся на него! Видишь, сколько висячих хвостов. Целых восемь!

Мурзик. Ума не приложу, откуда они берутся. Ведь алгоритм, описанный словами, выглядит таким стройным и логичным. Когда читаешь его, кажется, что все правильно, все точно. А потом почему-то оказывается, что слова обманчивы и под ними прячутся замаскированные ошибки. Может быть, здесь есть какая-то тайна?

Папа Циркуль. Обычный язык, на котором мы разговариваем, имеет много достоинств. Однако он плохо приспособлен для описания алгоритмов. Слова действительно обманчивы. Они скрывают много алгоритмических тайн и нередко вводят читателя в заблуждение. Поэтому ученые пришли к выводу: для записи алгоритмов нужны особые, *алгоритмические* языки.

Ты убедился сам — как только мы переходим от обычного языка к алгоритмическому языку ДРАКОН, тайное становится явным и «висячие хво-

ошибки» начинают вылезать из всех щелей. Замена обычного языка на алгоритмический полезна тем, что срывает с ошибок шапку-невидимку, помогает их устранить и тем самым избавляет нас от многих неприятностей.

Мурзик. В таком случае предлагаю раз и навсегда запретить писать алгоритмы на обычном языке. А нарушителям отрубать голову.

Папа Циркуль. Тогда бы ты сам уже давно ходил без головы. Нет, Мурзик, запрещать ничего не надо. Но тот, кто пытается описывать алгоритмы словами, должен знать, что идет по тонкому льду и совершенно напрасно рискует. Скорее всего, в конце работы он обнаружит в своих алгоритмах немало неприятных сюрпризов. И потратит впустую много сил, труда и времени на исправление ошибок. Этого можно избежать, если с самого начала писать алгоритмы на каком-либо алгоритмическом языке, например, на языке ДРАКОН.

Рис. 26. Алгоритм, в котором восемь ошибок «висячий хвост»

Совет

Не пиши алгоритмы на обычном языке — это чревато ошибками. Для разработки алгоритмов используй алгоритмический язык, например ДРАКОН

§ 25. РОБОТ КОЛЯ РЕШИЛ УТОПИТЬСЯ ОТ НЕСЧАСТНОЙ ЛЮБВИ

— Алина, дорогая! Я люблю тебя больше жизни. Выходи за меня замуж! — закричал робот Коля.

— За тебя? Замуж? Да ты с ума сошел! — Алина расхохоталась и презрительно фыркнула.

— О, Боже! Она никогда меня не полюбит! Зачем тогда жить?

И бедный Коля решил утопиться. Где? В бездонном океане, который окружает Шахматную Страну со всех сторон (рис. 27). Но утопиться не так-то просто, потому что на границе Шахматной Страны установлена глухая стена, которая отделяет ее от океана. Однако Коля знает, что в стене есть дыра. Чтобы Коля смог найти дыру, в его репертуар включен вопрос:

«Справа дыра?»

Если Коля найдет дыру и сделает шаг, он упадет в океан.

Рис. 27. Шахматная Страна — это остров в океане

А теперь задача. Известно, что Коля находится на клетке K, а дыра сделана в правой стене одной из клеток K, L, M, в какой именно — неизвестно. Требуется написать алгоритм, позволяющий Коле прыгнуть в океан (не бойся, он не утонет, потому что сделан из материала, который легче воды).

Решение задачи представлено на рис. 28.

Вопросы Саши Ехидного

1. Сколько команд выполнит Коля согласно алгоритму на рис. 28, если дыра находится в клетке K? Назови эти команды.
2. Сколько команд выполнит Коля согласно алгоритму на рис. 28, если дыра находится в клетке L? Назови эти команды.
3. Сколько команд выполнит Коля согласно алгоритму на рис. 28, если дыра находится в клетке M? Назови эти команды.
4. Известно, что на первой развилке (рис. 28) дракон-поезд идет направо, а на второй — вниз. В какой клетке находится дыра?
5. Робот Коля умеет выполнять команды «Шагни вверх», «Шагни вниз», «Шагни влево», «Шагни вправо» и понимать вопросы: «Сверху дыра?», «Снизу дыра?», «Слева дыра?» и «Справа дыра?». Его исходная позиция — клетка K (рис. 28). Известно, что дыра находится либо в правой стене (в одной из клеток K, L, M), либо в нижней стене (в одной из клеток C, F, M). Напиши алгоритм, позволяющий Коле прыгнуть в океан.
6. Реши задачу 5 при условии, что исходная Колина позиция — клетка A.
7. Реши задачу 5 для случая, когда Коля вначале находится в клетке E, а дыра в стене может быть в любом месте.

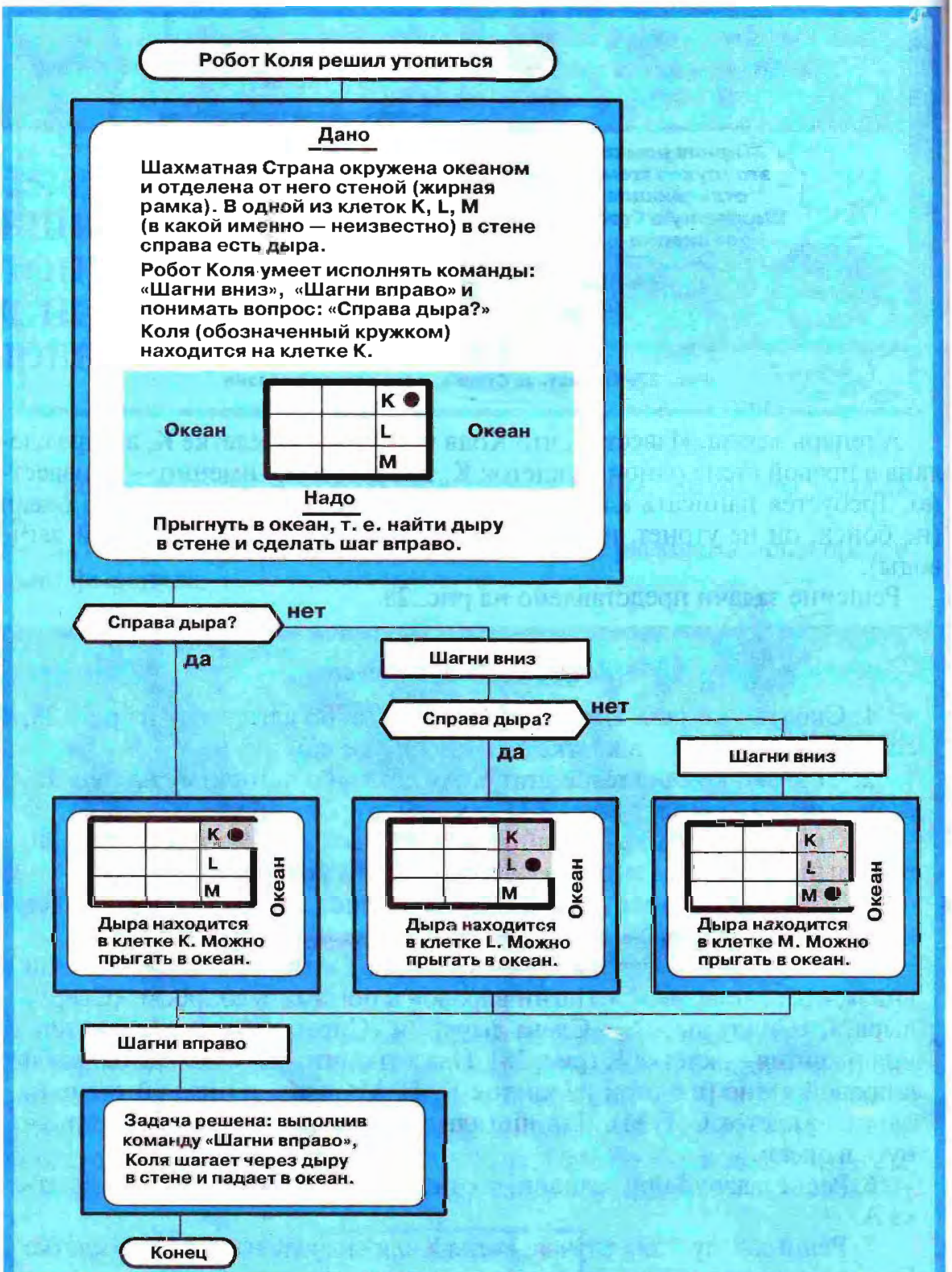


Рис. 28. Алгоритм «Робот Коля решил утопиться»

§ 26. КАК УСТРОЕН РОБОТ? ЧТО У НЕГО ВНУТРИ?

Папа Циркуль. В этом пузырьке волшебный эликсир, который пила Алиса, чтобы уменьшиться в размерах и превратиться в дюймовочку. Выпив его, ты станешь маленьким, как булавочная головка, и сможешь совершить прогулку по внутренностям робота.

Мурзик пьет эликсир. Свет гаснет, звучит тихая музыка. Из темноты появляется странного вида незнакомец.

— Добро пожаловать, Мурзик. меня зовут Мозг. Кто я такой? Сейчас узнаешь. Для начала я объясню, куда ты попал. Ты находишься внутри робота по имени Коля.

Мурзик. Как это внутри? Ведь Коля — всего лишь маленький черный кружок, нарисованный на бумаге.

Мозг. Вовсе нет. Любой робот, и Коля в том числе — это электромеханический самодвижущийся прибор.

Мурзик. А как он устроен?

Мозг. В нем три части: Мозг (это я), Глаз и Мускул (рис. 29). Сейчас мы покажем тебе, как мы работаем, когда робот выполняет алгоритм. Какой алгоритм ты хотел бы посмотреть?

Мурзик. Про то, как Коля решил утопиться.

Мозг. Внимание! По просьбе Мурзика исполняется алгоритм на рис. 28. Прошу всех членов экипажа занять свои места. Вы помните свои обязанности? Для бестолковых повторяю. Глаз, когда я спрошу тебя: «Справа дыра?» — что ты должен делать?

Глаз. Я должен выглянуть из окошка и внимательно посмотреть. Если увижу справа дыру в стене, отвечаю «да», в противном случае — «нет».

Мозг. Молодец, Глаз. Мускул, а ты у нас зачем? Какова твоя роль?

Мускул. Я — это мотор робота, его мышцы и ноги. Я делаю всю физическую работу. Именно я шагаю из клетки в клетку. Когда ты, Мозг, прикажешь мне «Шагни вниз» или «Шагни вправо», я точно выполню приказ. При этом робот переместится туда, куда надо.

Мурзик. Не понимаю. Кто же все таки шагает: Мускул или робот?

Мозг. Сейчас поймешь. Предположим, в алгоритме есть команда «Шагни вниз». Эту команду читаю я, Мозг. Затем я передаю ее Мускулу, как показано на рис. 29, и говорю: «Мускул, шагни вниз». Выполняет мой приказ именно Мускул. Он-то и передвигает робота в нужное место.

Мурзик. Понятно. Поехали дальше.

Мозг. В алгоритме на рис. 28 три маршрута. Выбирай, Мурзик, по какому из них мы должны пройти через алгоритм?

Мурзик. По тому, где дыра в стене находится в клетке М.

Мозг. Все ясно. А теперь — приготовиться! Внимание! Начали! (*Мозг достает рис. 28 и начинает читать алгоритм.*) В начале алгоритма вопрос «Справа дыра?» Глаз, посмотри, есть справа дыра?

Глаз. Нет.

Мозг. Мускул, шагни вниз.

(Мускул шагает вниз и робот Коля перемещается из клетки К в клетку L).

Мозг. Глаз, справа дыра?

Глаз. Нет.

Мозг. Мускул, шагни вниз.

(Мускул шагает вниз и робот Коля перемещается из клетки L в клетку М).

Мозг. Глаз, справа дыра?

Глаз. Да.

Мозг. Мускул, шагни вправо.

(Мускул шагает вправо сквозь дыру в стене и робот Коля стремительно падает в океан).

Папа Циркуль. Спасайся, Мурзик! Выпей эликсир из другого пузырька!

Мурзик (*пьет эликсир и в тот же миг возвращается на прежнее место*). Ну и натерпелся я страху. Это же надо! Чуть в океан не свалился! Зато теперь я знаю, как устроены роботы и как они выполняют алгоритм.

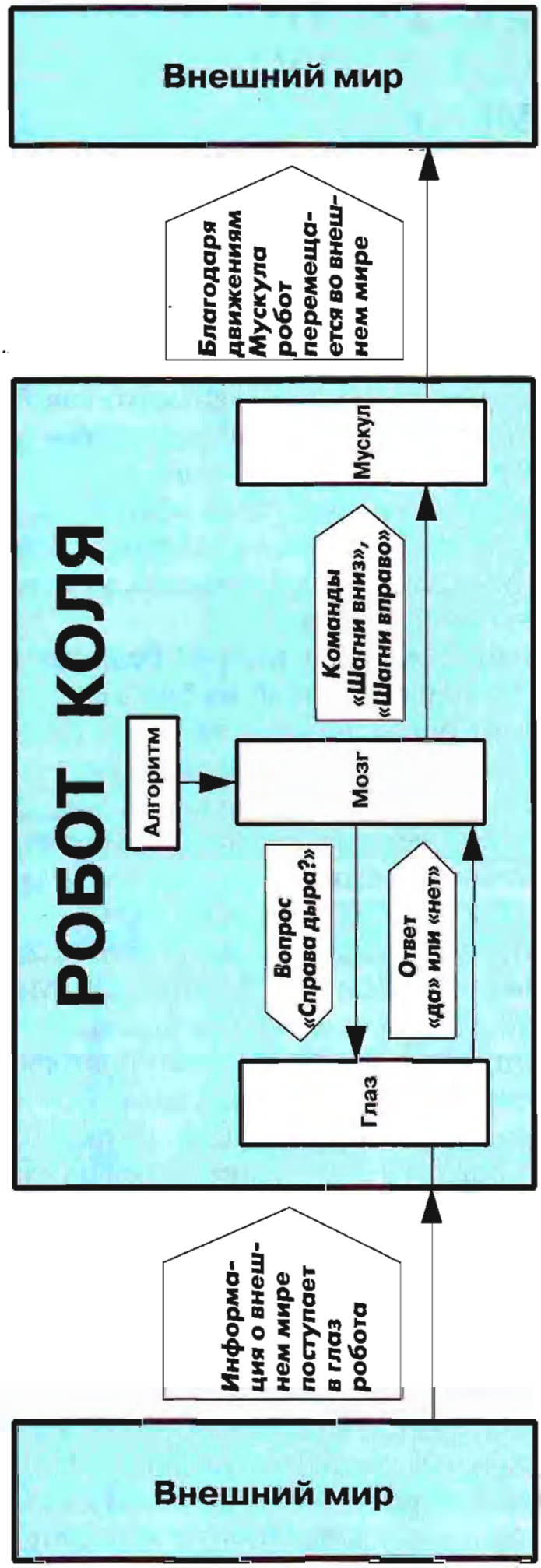


Рис. 29. Как устроен робот Коля?

УЧИСЬ РИСОВАТЬ ПОНЯТНЫЕ АЛГОРИТМЫ

§ 27. ОШИБКИ В АЛГОРИТМАХ — УЖАСНОЕ БЕДСТВИЕ

Мурзик. Что самое главное в алгоритме?

Папа Циркуль. Главное — чтобы алгоритм решал поставленную задачу. Чтобы он был правильный. Чтобы в нем не было ошибок. Ведь алгоритм с ошибками не позволяет решить задачу. Такой алгоритм никому не нужен.

Мурзик. А по-моему, ошибки — это мелочь. Ведь их всегда можно исправить! Стоит ли из-за пустяков шум подымать?

Папа Циркуль. Ты не прав. Ошибки в алгоритмах — настоящее бедствие. Из-за них космические ракеты «сходят с ума» и летят мимо цели, ломаются спутники, разбиваются самолеты, гибнут люди, взрываются заводы, портится продукция, начинается хаос, нарушается жизнь общества.

На исправление ошибок уходят огромные деньги. Поэтому нужно с самого начала тщательно проверять алгоритмы. Чем раньше мы обнаружим ошибку, тем дешевле ее исправить.

Мурзик. За чем же дело стало? Предлагаю ввести закон: написал алгоритм — сразу проверь! А кто нарушит — казнить!

Папа Циркуль. Твой закон не поможет, и вот почему. Искать ошибки в алгоритмах не так-то просто, это дело кропотливое. Чтобы хорошенько проверить алгоритм, нужно глубоко его понять. К сожалению, многие алгоритмы плохо приспособлены для быстрой и надежной проверки, для легкого и быстрого понимания.

Мурзик. Как же быть?

Папа Циркуль. Мы должны научиться рисовать алгоритмы, удобные для понимания и проверки.

Мурзик. Разве это возможно?

Папа Циркуль. Да. Такие алгоритмы называют *эргономичными*.

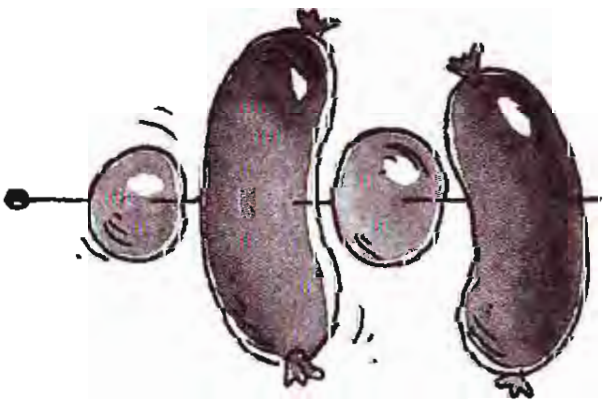
Хлястик. Я не понял. Объясните подробнее, зачем они нужны?

Папа Циркуль. Эргономичные алгоритмы намного лучше, яснее и нагляднее, чем обычные. Если алгоритм непонятный, в нем трудно заметить затаившуюся ошибку. И наоборот, чем понятнее алгоритм, тем легче найти дефект. Поэтому более понятные, эргономичные алгоритмы — мощное средство для борьбы с ошибками. Кроме того, эргономичные алгоритмы удобнее для изучения, их проще объяснить другому человеку.

Что такое эргономичный алгоритм	**Это алгоритм, легкий для понимания, удобный для проверки и поиска ошибок**
Запомни	● **Ошибки в алгоритмах — это плохо** ● **Чтобы облегчить поиск ошибок, нужно сделать алгоритм ясным и доходчивым** ● **Сделать алгоритм ясным и доходчивым — значит сделать его эргономичным**

Мурзик. По каким правилам рисуют эргономичные алгоритмы?

Папа Циркуль. Скоро узнаешь. А пока давай немножко отдохнем и ответим на очень «вкусный» вопрос: что такое шашлык по-карски?



§ 28. ЧТО ЛУЧШЕ: АЛГОРИТМ ИЛИ ШАШЛЫК?

а *б*

Рис. 30. Алгоритм немножко похож на шашлык по-карски

Шашлык по-карски делают очень просто: берут кусок баранины и бараньи почки и нанизывают на шампур так, чтобы кусок мяса оказался между почками (рис. 30*а*).

А теперь посмотри на рис. 30*б* — не правда ли, «сходство» между шашлыком и алгоритмом сразу бросается в глаза? Это забавное сходство позволяет легко запомнить одно из правил рисования дракон-схем, которое называется «правило шампура». Оно гласит: *выход иконы «заголовок» и вход иконы «конец» должны находиться на одной вертикали*. Мы уже знаем (§ 14), что по шампуру проходит прямой путь алгоритма. Все обходные пути рисуют справа от шампура.

На рис. 31 показана типичная ошибка: дракон-схема без шампура.

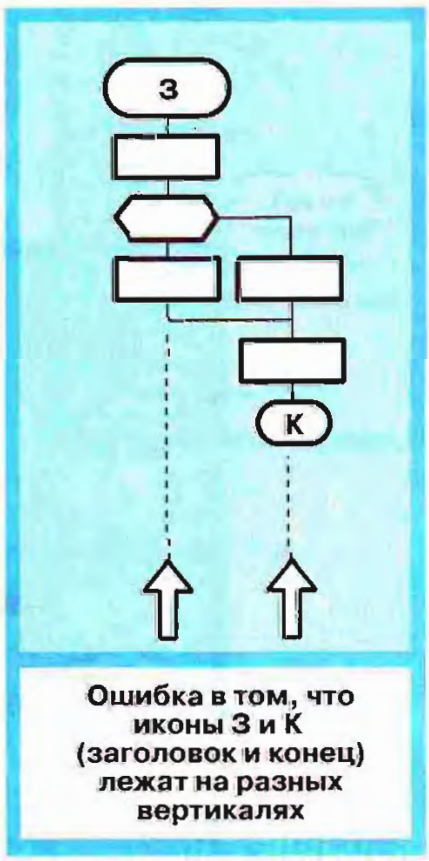


Рис. 31. Неправильно нарисованный алгоритм. Ошибка, которая называется «Нет шампура»

Мурзик. Как это без шампура?

Папа Циркуль. *Шампур* — вертикаль, проходящая через иконы «Заголовок» и «Конец». Однако на рис. 31 такая вертикаль отсутствует, следовательно, шампура в этой схеме нет. Дракон-схема без шампура является запрещенной — кто ее нарисует, получит жирную двойку.

Что такое шампур?

- **Это вертикальная линия, соединяющая икону «Заголовок» и икону «Конец»**
- **По ней проходит прямой путь алгоритма.**

Правило шампура

Выход иконы «Заголовок» и вход иконы «Конец» должны лежать на одной вертикали

§ 29. ЧТО ТАКОЕ ПЛЕЧО РАЗВИЛКИ?

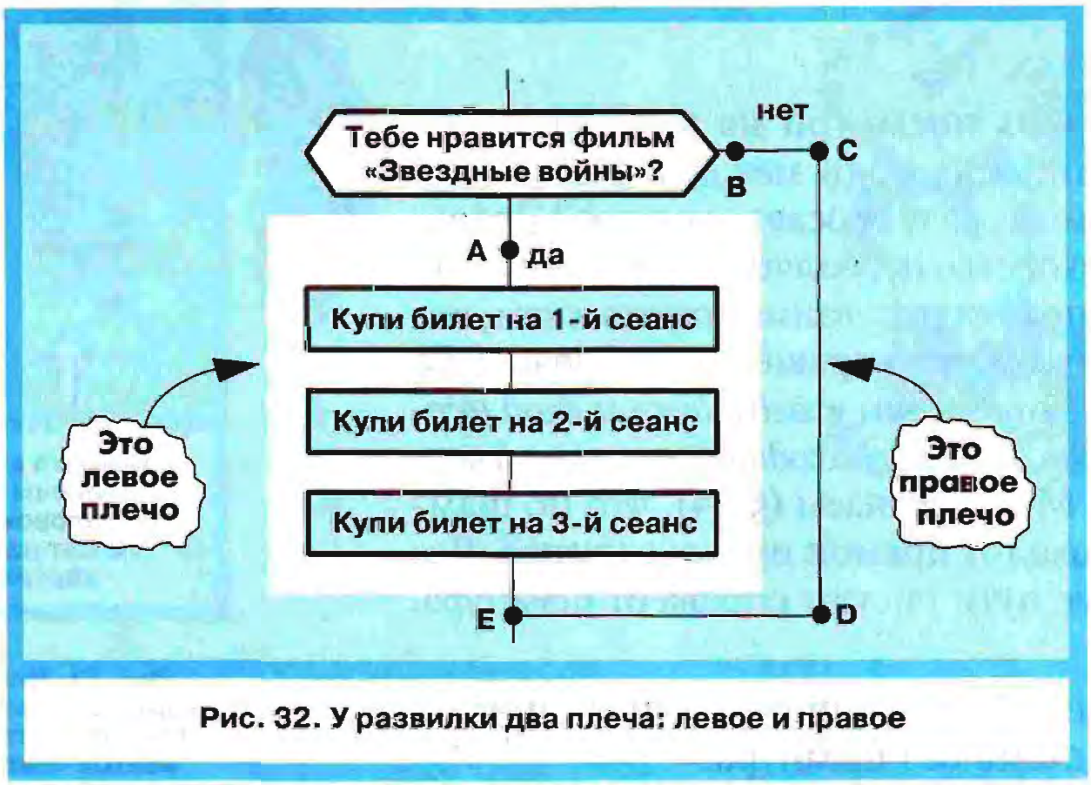


Рис. 32. У развилки два плеча: левое и правое

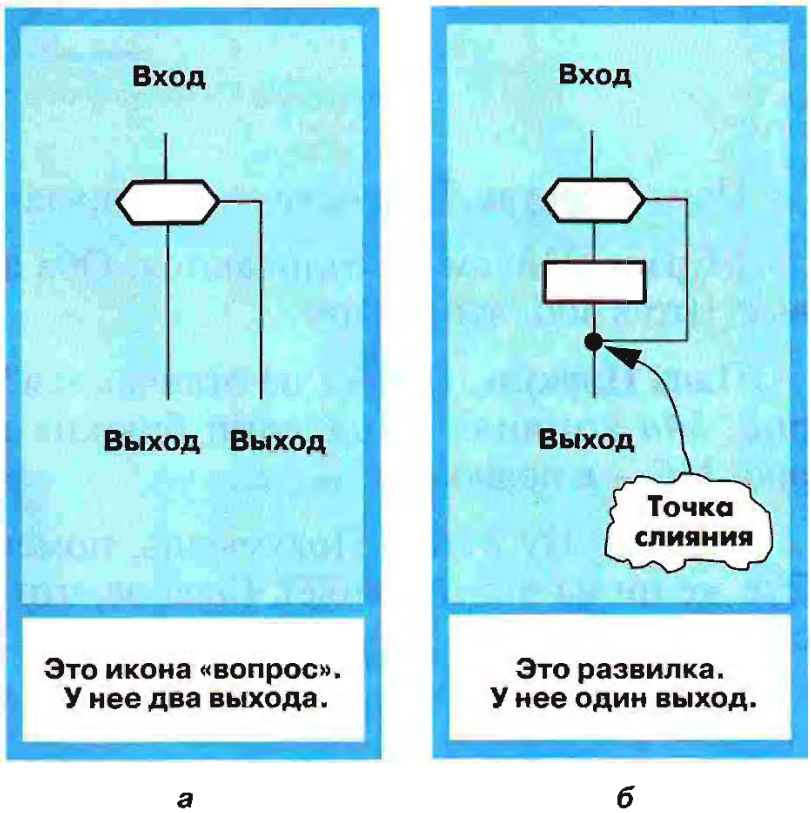
Папа Циркуль. Мурзик, запомни, пожалуйста, очень легкое правило: *у развилки два плеча, левое и правое.*

Суть дела ясна из примера на рис. 32.

Левое плечо — это маршрут от нижнего выхода иконы «Вопрос» (точка A) до *точки слияния* E. Оно содержит ответ «да», три иконы и соединительные линии.

Правое плечо — это путь от правого выхода до точки слияния. На рис. 32 оно содержит ответ «нет» и ломаную линию BCDE.

§ 30. ЧЕМ ОТЛИЧАЕТСЯ РАЗВИЛКА ОТ ИКОНЫ «ВОПРОС»?



а *б*

Рис. 33. Чем отличается развилка от иконы «Вопрос»?

Мурзик. Чтобы сделать в алгоритме разветвление, нужно изобразить икону «Вопрос». Так?

Папа Циркуль. Так.

Мурзик. Значит, икона «Вопрос» и развилка — одно и то же?

Папа Циркуль. Нет, это разные вещи.

Мурзик. А в чем же разница?

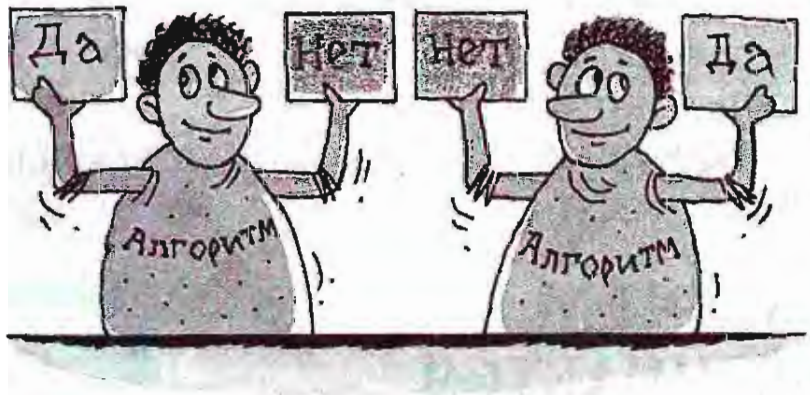
Папа Циркуль. Если ты нарисуешь икону «Вопрос» и этим ограничишься, у тебя получится ошибка «висячий хвост» (рис. 33*а*). Чтобы убрать хвосты, надо пририсовать к иконе «Вопрос», плечи и точку слияния, т. е. превратить ее в *развилку,* как на рис. 33*б*.

Что такое развилка

Это часть дракон-схемы, в которую входят четыре элемента:

- **икона «Вопрос»;**
- **левое плечо;**
- **правое плечо;**
- **точка слияния.**

§ 31. ЧТО ТАКОЕ РОКИРОВКА?



Папа Циркуль. Чем отличаются дракон-схемы на рис. 34?

Мурзик. Ничем не отличаются. Оба алгоритма совершенно одинаковые. Что в лоб, что по лбу.

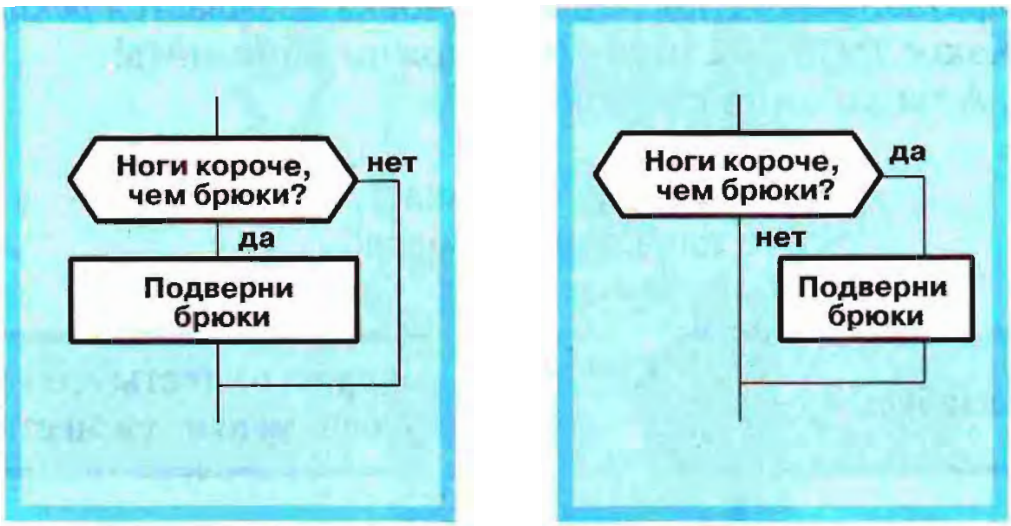
Папа Циркуль. Как же не отличаются? Посмотри-ка внимательнее. На рис. 34*а* команда «Подверни брюки» находится в левом плече, а на рис. 34*б* — в правом.

Мурзик. Ну и что? Подумаешь, поменяли местами плечи у развилки! Это же ни на что не влияет. Главное, что смысл алгоритма остался тот же самый.

Папа Циркуль. А можешь ты эту мысль изложить в виде правила?

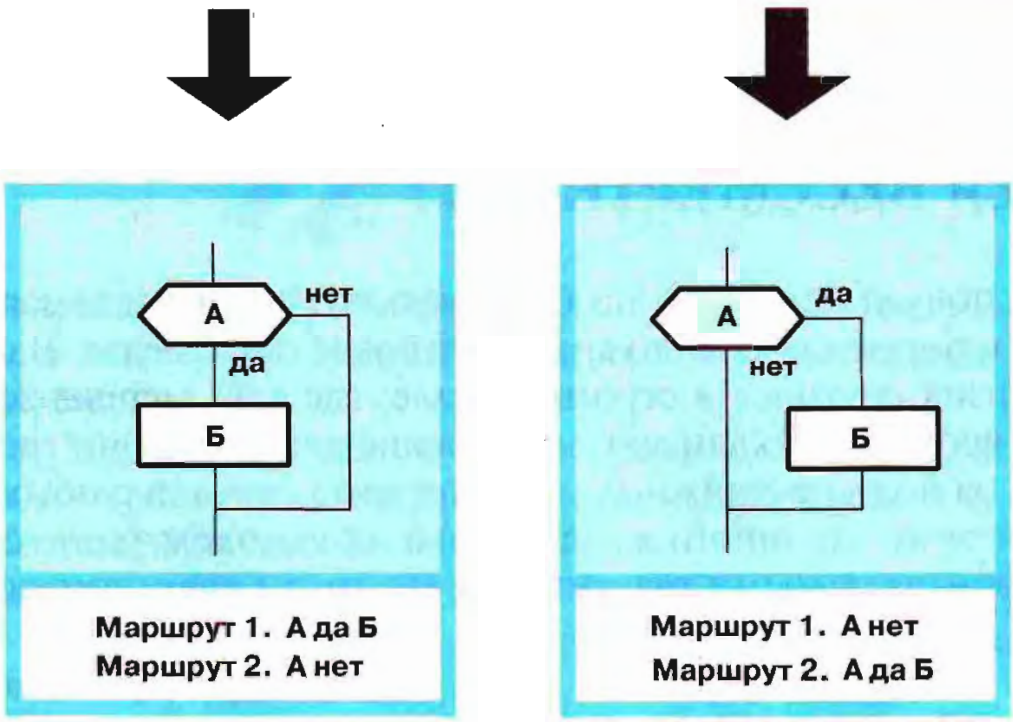
Мурзик. Конечно, могу.

Поменяйте две штанины —
Не изменятся штаны.
Поменяй местами плечи —
Алгоритму хоть бы хны.



а *б*

Рис. 34. Пример равносильных алгоритмов



а *б*

Рис. 35. Буквенные дракон-схемы, полученные из смысловых дракон-схем на рис. 34

Папа Циркуль. Нет, дружок, на твоих стихах мы далеко не уедем. Здесь нужны строгие правила.

Мурзик. Пожалуйста, вот вам и правило: *если у развилки переставить плечи, смысл алгоритма не изменится.*

Папа Циркуль. Верно. Такая перестановка называется *рокировка.*

Алина. Какое трудное слово! Невозможно запомнить!

Мурзик. А ты запомни стишок:

Что такое рокировка?
Это плеч перестановка!

Не забудь

При рокировке ответы «да» и «нет» обязательно меняются местами

Задание Миши Проверялкина

1. Произведи рокировку в алгоритме на рис. 1.
2. Произведи рокировку в алгоритме на рис. 6.
3. Произведи рокировку левой нижней развилки на рис. 7.

§ 32. В ГОСТЯХ У КАЩЕЯ БЕССМЕРТНОГО

Мурзик почуял неладное, но было уже поздно. Неведомая сила подхватила его и потащила куда-то вниз, в глубокое подземелье. Наконец гром утих, и Мурзик очутился в огромном зале, где едва мерцал зеленоватый свет. Везде яростно квакали математические лягушки. Они громоздились повсюду — на полу, на книжных полках и даже свисали с потолка.

Посреди этого странного лягушатника на высоком троне сидел седой старик с пронзительными глазами. На его груди светилась золотая табличка:

Председатель математических наук академик Кащей Бессмертный

Время от времени из его ноздрей вырывались огонь и клубы дыма. Кащей был явно недоволен. Наконец он произнес дребезжащим, металлическим голосом:

— Это ты, Мурзик? Хорош, нечего сказать!

У Мурзика перехватило дыхание. Дрожа, как осиновый лист, он пролепетал: «А в чем де-де-дело?»

Кащей метнул грозный взгляд, выпустил сноп огня, и его бесхвостые красавицы кинулись врассыпную. Вскоре, однако он успокоился и довольно миролюбиво пробурчал:

— Ты попал в дурную компанию. Идя на поводу у этой скверной девчонки — ей, видите ли, не нравятся лягушки —легкомысленный автор нашей книжки выбросил из курса информатики все математические примеры. Это же курам на смех! Я, как старый математический волк, не могу с этим согласиться. Поэтому здесь, в глухом подземелье, я организовал подпольную математическую школу. Надеюсь, ты любишь математику?

«Разве можно любить такую гадость?» — хотел было крикнуть Мурзик, но лягушки заквакали так громко, что он против своей воли воскликнул:

— Да-да, конечно! Я ее просто обожаю!

— Вот и отлично! — обрадовался Кащей. — Мы немедленно начинаем математический урок. Ты готов?

«Ну и влип же я в историю!» — тоскливо подумал Мурзик, но вслух бойко отчеканил:

— Готов, господин академик!

§ 33. КАЩЕЙ БЕССМЕРТНЫЙ И РАВНОСИЛЬНЫЕ АЛГОРИТМЫ

Только для тех, кто любит математику. Остальные должны сказать: «Чур меня!» — и пропустить этот параграф

Кащей Бессмертный. Начнем с повторения. Ты помнишь, что в алгоритмах всегда есть маршруты? В одних алгоритмах их мало, в других — много. Например, на рис. 22 — один маршрут, на рис. 23 — два, на рис. 24 — три. Верно?

Мурзик. Верно.

Кащей Бессмертный. Два алгоритма называются равносильными, если они имеют один и тот же набор маршрутов.

Мурзик. Что это значит?

Кащей Бессмертный. Сейчас поймешь. На рис. 34*а* и *б* показаны два смысловых алгоритма. Для каждого из них нарисуй буквенную схему и определи набор маршрутов.

Мурзик. Готово! Ответ показан на рис. 35*а* и *б*.

Кащей Бессмертный. Очень хорошо! Сравни наборы маршрутов и скажи: они одинаковые или разные?

Мурзик (*рисует две таблички*):

Набор маршрутов для рис. 35*а*	
1	А да Б
2	А нет

Набор маршрутов для рис. 35*б*	
1	А нет
2	А да Б

Из табличек видно, что два алгоритма на рис. 35 имеют в точности одинаковый набор маршрутов. Следовательно, эти алгоритмы равносильны.

Кащей Бессмертный. Молодец! А теперь вспомни, что такое рокировка?

Мурзик. Это преобразование алгоритма, при котором плечи у развилки меняются местами.

Кащей Бессмертный. Преобразование алгоритма X в алгоритм Y называется равносильным, если алгоритмы X и Y равносильны.

Мурзик. Ура! Я понял! Рокировка — это равносильное преобразование алгоритма. Рисунки 34 и 35 подтверждают это!

Кащей Бессмертный. Ты, я вижу, смышленый. Запомни: равносильные преобразования алгоритмов — очень полезная вещь.

В следующих параграфах будет показано, что равносильные преобразования позволяют решить важную задачу: превратить плохую (неэргономичную) дракон-схему в хорошую (эргономичную).

Что такое набор маршрутов

- **Это множество маршрутов данного алгоритма**
- **Каждый маршрут пишут в виде формулы**

Что такое равносильные алгоритмы

Это алгоритмы, имеющие одинаковый набор маршрутов

Как доказать, что алгоритмы X и Y равносильны

Для этого достаточно:

- **определить набор маршрутов алгоритма X**
- **определить набор маршрутов алгоритма Y**
- **убедиться, что два набора маршрутов совпадают**

Первая теорема Кащея Бессмертного

Равносильные алгоритмы имеют один и тот же смысл (это значит, что при одинаковых исходных данных они дают одинаковый результат)

Вторая теорема Кащея Бессмертного

При равносильных преобразованиях смысл алгоритма не меняется

§ 34. ЕСТЬ ЛИ В АЛГОРИТМЕ ЦАРСКАЯ ДОРОГА?

Папа Циркуль (*подозрительно нюхает воздух*). Почему от тебя пахнет лягушками? Что-нибудь случилось?

Мурзик. Ничего особенного. Просто я был в гостях у одного чудака, который живет в подземном лягушатнике.

Папа Циркуль. Знаю я этого чудака. Ты с ним поосторожней. Ну да ладно, забудем об этом. Давай-ка лучше сыграем в игру, которая называется «Царская дорога». Хочешь?

Мурзик. Конечно.

Папа Циркуль. Значит, так. Царь любит только хорошее и полезное, а плохое терпеть не может. Найди царскую дорогу на рис. 19.

Мурзик. В развилке «Есть желание делать зарядку?» царская дорога идет через «да», потому что делать зарядку полезно и хорошо. В иконе «Воду отключили?» она идет через «нет», так как с водой хорошо, а без воды плохо. В развилке «В глаз попала соринка?» также идем через «нет»: соринка — это больно и плохо, а без соринки, наоборот, хорошо. В иконе «Любимая расческа потерялась?» царский путь проходит через «нет»: чего хорошего, если расческа пропала. В развилке «Прическа получилась красивая?» идем через «да»: красиво — это хорошо. Далее: «Голубое платье испачкалось?» — идем через «нет», так как пачкать вещи плохо. «Мама приготовила завтрак?» — царский маршрут идет через «да», потому что нормально поесть — это хорошо, а перехватить всухомятку — плохо. «На улице мороз ниже 30 градусов?» — идем через «нет», так как посещать школу хорошо, а пропускать — плохо.

Папа Циркуль. Все верно. А теперь чуть-чуть изменим правила. Вместо «царский маршрут» будем говорить «главный маршрут».

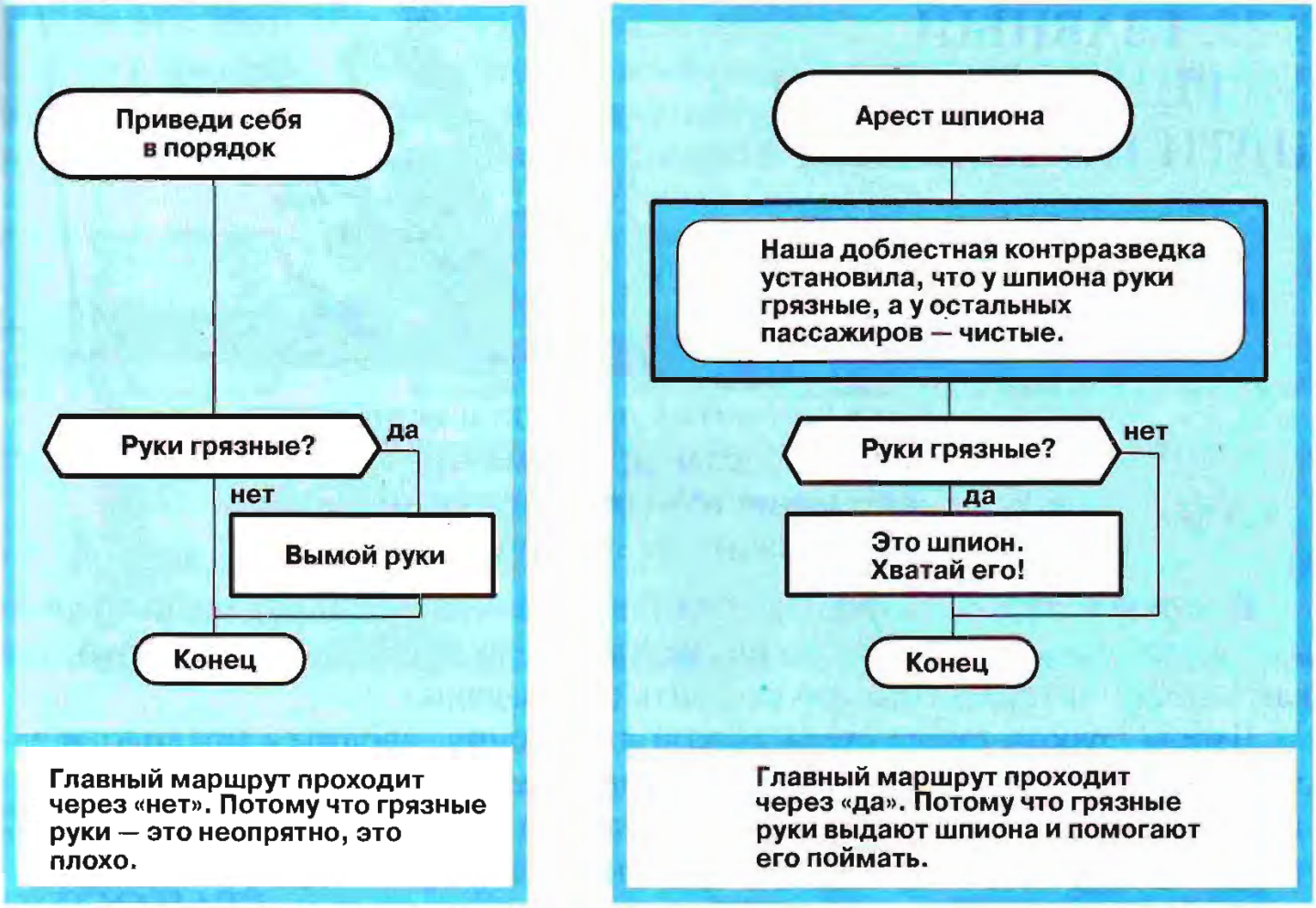
Мурзик. Договорились.

Папа Циркуль. На рис. 36*а* есть вопрос: «Руки грязные?» Где проходит главный маршрут: через «да» или через «нет»?

Мурзик. Через «нет».

Папа Циркуль. Почему?

Мурзик. Потому что чистые руки — это хорошо, а грязные — плохо. Главный маршрут идет там, где хорошо.



а *б*

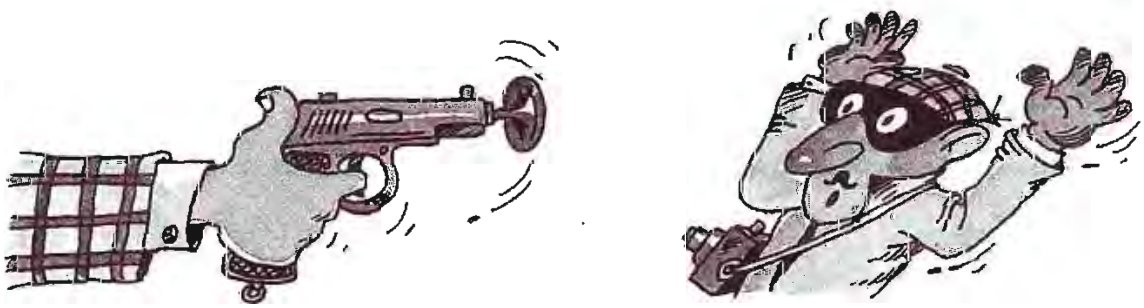
Рис. 36. Маршрут называется главным, если он быстрее ведет к цели

Папа Циркуль. Ладно. А что ты скажешь про рис. 36 *б*?

Мурзик. То же самое.

Папа Циркуль. А вот здесь ты ошибся. *Главный маршрут* — тот, что быстрее ведет к цели. В чем цель алгоритма на рис. 36*б*? В том, чтобы поймать шпиона. Если шпион пойман — это успех, это хорошо. Значит, в данном случае грязные руки — это хорошо, так как они выдают шпиона с головой. И наоборот, чистые руки — это плохо, так как они означают, что перед нами обычный пассажир, а шпион опять ускользнул.

Твоя ошибка в том, что ты рассуждаешь абстрактно: хорошо или плохо. А надо спрашивать по-другому: помогает это условие достижению цели алгоритма или мешает? Если помогает — хорошо. Если мешает — вот тогда плохо.



§ 35. ГЛАВНЫЙ МАРШРУТ ДОЛЖЕН ИДТИ ПО ШАМПУРУ

Все в алгоритме понятно и ясно,
Если он сделан эргономично.
Эргономично — это прекрасно!
Эргономично — значит отлично!

В неумело нарисованных алгоритмах главный маршрут очень трудно найти: словно маскируясь, он петляет то влево, то вправо. Это плохо, так как мешает читателю быстро ухватить суть задачи.

Чтобы дракон-схема стала легкой для чтения, вводится правило: *в эргономичном алгоритме главный маршрут должен идти по шампуру*. Это значит, что царская дорога должна быть прямая как стрела. На рис. 37*а* в качестве примера показана плохая дракон-схема, где это правило не выполняется.

Плохая дракон-схема. Нарушено правило: «Главный маршрут должен идти по шампуру»

а

Хорошая (эргономичная) дракон-схема. Главный маршрут идет по шампуру

б

Рис. 37. Как пустить главный маршрут по шампуру? Поменяйте местами плечи у развилки

Мурзик. Не понял. Почему не выполняется?

Папа Циркуль. Потому что главный маршрут (жирная линия) не прямой, а изогнутый. Чтобы исправить ошибку, нужно сделать рокировку и переставить плечи у развилки, как показано на рис. *37б*.

Правило

- **Главный маршрут должен идти по шампуру**
- **Если он не идет по шампуру, надо произвести рокировку**

Задание Миши Проверялкина

1. Докажи, что алгоритмы на рис. 37*а* и *б* равносильны.
Подсказка. Прочитай §33 и обрати особое внимание на правила в рамке.

§ 36. ПОБОЧНЫЕ МАРШРУТЫ НЕЛЬЗЯ РИСОВАТЬ КАК ПОПАЛО

«О ужас! Я, кажется, потерял деньги!»

Кому случалось делать подобное открытие, знает, что степень огорчения зависит от потерянной суммы.

Рассмотрим пять возможных ситуаций и дадим им оценку.

	Ситуация	Оценка
1	Я ничего не потерял	Хорошо
2	Я потерял рубль	Плохо
3	Я потерял пять рублей	Очень плохо
4	Я потерял десять рублей	Совсем плохо
5	Я потерял всю получку	Хуже некуда

На рис. 38 показана дракон-схема, описывающая эту грустную историю. На схеме пять вертикалей. По каждой вертикали идет свой маршрут.

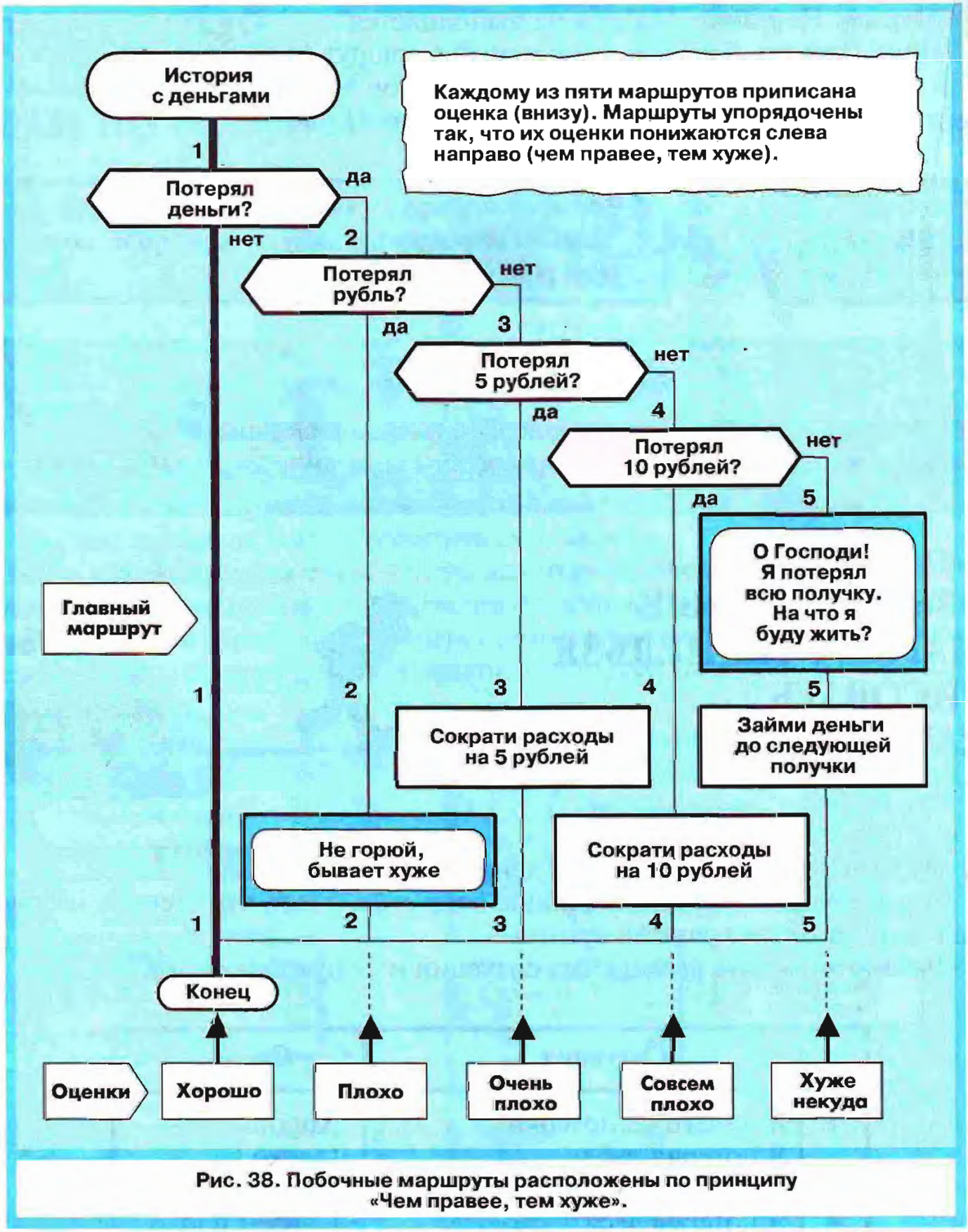


Рис. 38. Побочные маршруты расположены по принципу «Чем правее, тем хуже».

Самая первая, крайняя слева вертикаль — это шампур. По ней идет главный маршрут, имеющий оценку «хорошо», потому что все деньги целы. Чуть правее находится вторая вертикаль (потерян рубль) с оценкой «плохо». Еще правее идет третья вертикаль (потеряны 5 рублей) с оценкой «очень плохо». И так далее. Крайняя справа — пятая вертикаль описывает

самую плохую ситуацию, когда потеряна вся получка. Таким образом, четыре побочных маршрута, идущие по вертикалям 2, 3, 4, 5, расположены не случайно, а со смыслом. Они выстроены слева направо по принципу: «Чем правее, тем хуже».

Что такое побочный маршрут	**Любой маршрут разветвленного алгоритма за исключением главного**
Запомни	● **Все побочные маршруты находятся правее шампура.** ● **Располагать их левее шампура запрещается**

§ 37. ЧТО ЛУЧШЕ: ПОРЯДОК ИЛИ ПУТАНИЦА?

Папа Циркуль. Отгадай загадку. В некотором царстве, в некотором государстве были два города. Один строился без плана, вкривь и вкось: тесные улочки и переулки сплелись в змеиный клубок. Зато другой был образцовым: красивые площади, широкие проспекты, всюду простор и порядок. В каком городе легче найти дорогу?

Мурзик. Конечно, во втором.

Папа Циркуль. А теперь представь, что ты должен изучить незнакомый алгоритм. Если в нем, как в первом городе, нет порядка, он превращается в запутанный лабиринт, в котором ничего нельзя понять.

Если же он, как второй город, построен по хорошему плану, ситуация в корне меняется. Про такой алгоритм говорят: все ясно, как на ладони!

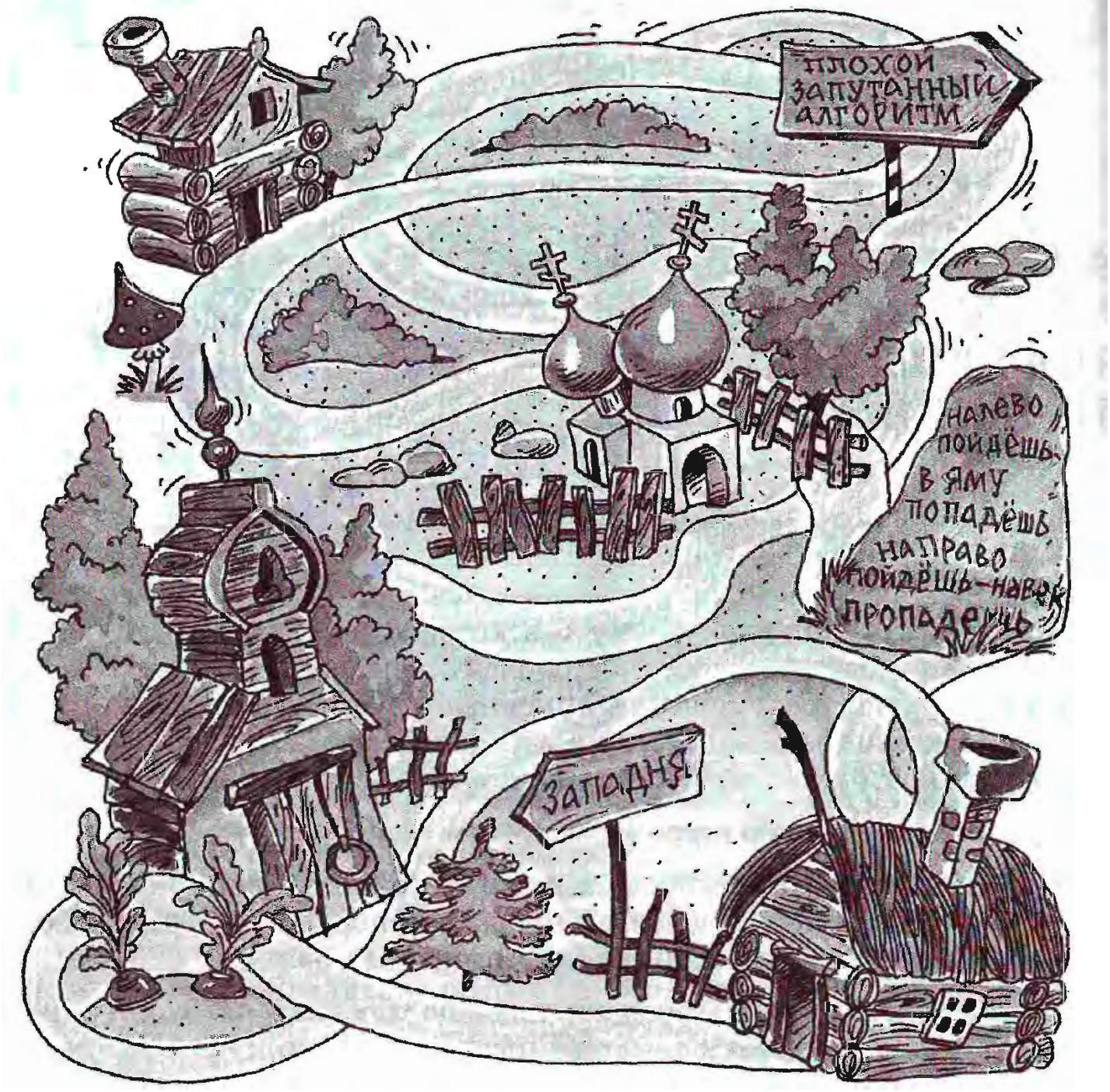
Чтобы убедиться в этом, посмотри еще разок на рис. 38 и проведи взглядом по всем вертикалям слева направо. Ты обнаружишь не хаос, а строгий порядок. Потому что маршруты нарисованы не как попало, а по правилам.

Первое правило

- **В эргономичном алгоритме крайняя левая вертикаль (шампур) считается самым почетным местом, царской дорогой**
- **Главный маршрут должен идти по царской дороге (т.е. по шампуру).**

Второе правило

Побочные маршруты должны быть упорядочены слева направо



§ 38. КАК ОТЛИЧИТЬ ХОРОШУЮ ДРАКОН-СХЕМУ ОТ ПЛОХОЙ?

Папа Циркуль. Сейчас, Мурзик, я устрою тебе экзамен. На рис. 39 представлены четыре схемы, на которых описана история с чернильницей. Обрати внимание: на всех схемах изображен один и тот же алгоритм. Однако схемы выглядят по-разному. Попробуй сообразить, какие из них начерчены правильно, а какие нет.

Мурзик. Сначала нужно найти главный маршрут. В развилке «Чернильница упала?» главный маршрут идет через «нет». Потому что когда вещи падают — это плохо, а когда не падают — хорошо. Значит, две схемы — на рис. 39*а* и *б* — нарисованы неверно.

Папа Циркуль. Объясни почему.

Мурзик. Согласно правилу главный маршрут (жирная линия) должен идти по шампуру. А он вместо этого петляет по задворкам, как заяц.

Папа Циркуль. Молодец! Давай дальше.

Мурзик. Теперь проверим правило побочных маршрутов. На рис. 39 их два. Один описывает ситуацию, когда чернильница упала, но уцелела; во втором случае она разбилась. Первой ситуации выставим оценку «плохо», второй — «очень плохо». На рис. 39*в* маршруты не упорядочены — значит, схема нарисована неверно.

Папа Циркуль. Почему?

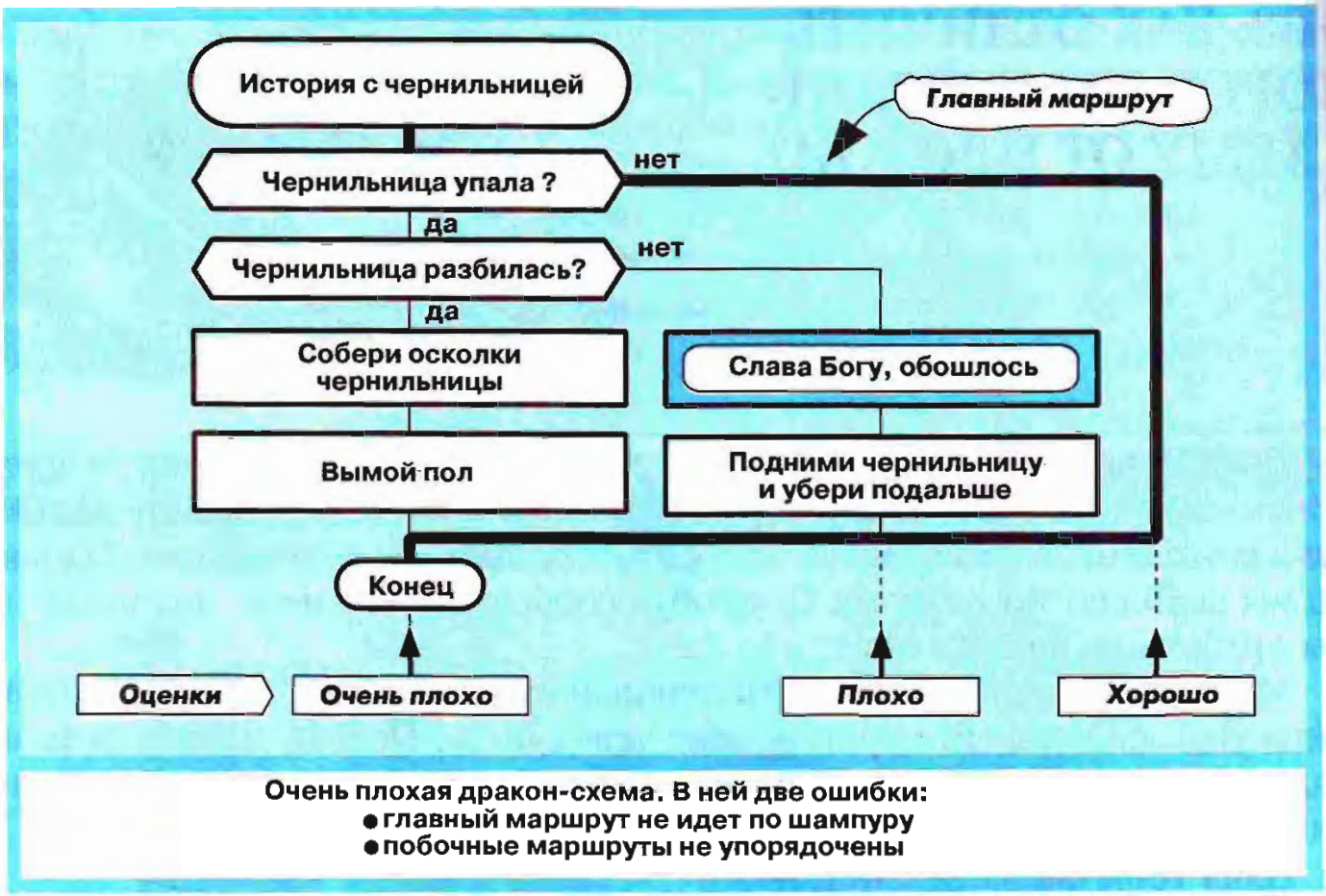
Мурзик. Потому что самый плохой маршрут (с оценкой «очень плохо») должен быть крайним справа. А он по ошибке затесался в середину.

Папа Циркуль. Верно.

Мурзик. Таким образом, правильно нарисована только одна, самая последняя схема (рис. 39*г*). В ней нет ни одной ошибки: главный маршрут идет по шампуру, и все маршруты упорядочены по принципу «Чем правее, тем хуже».

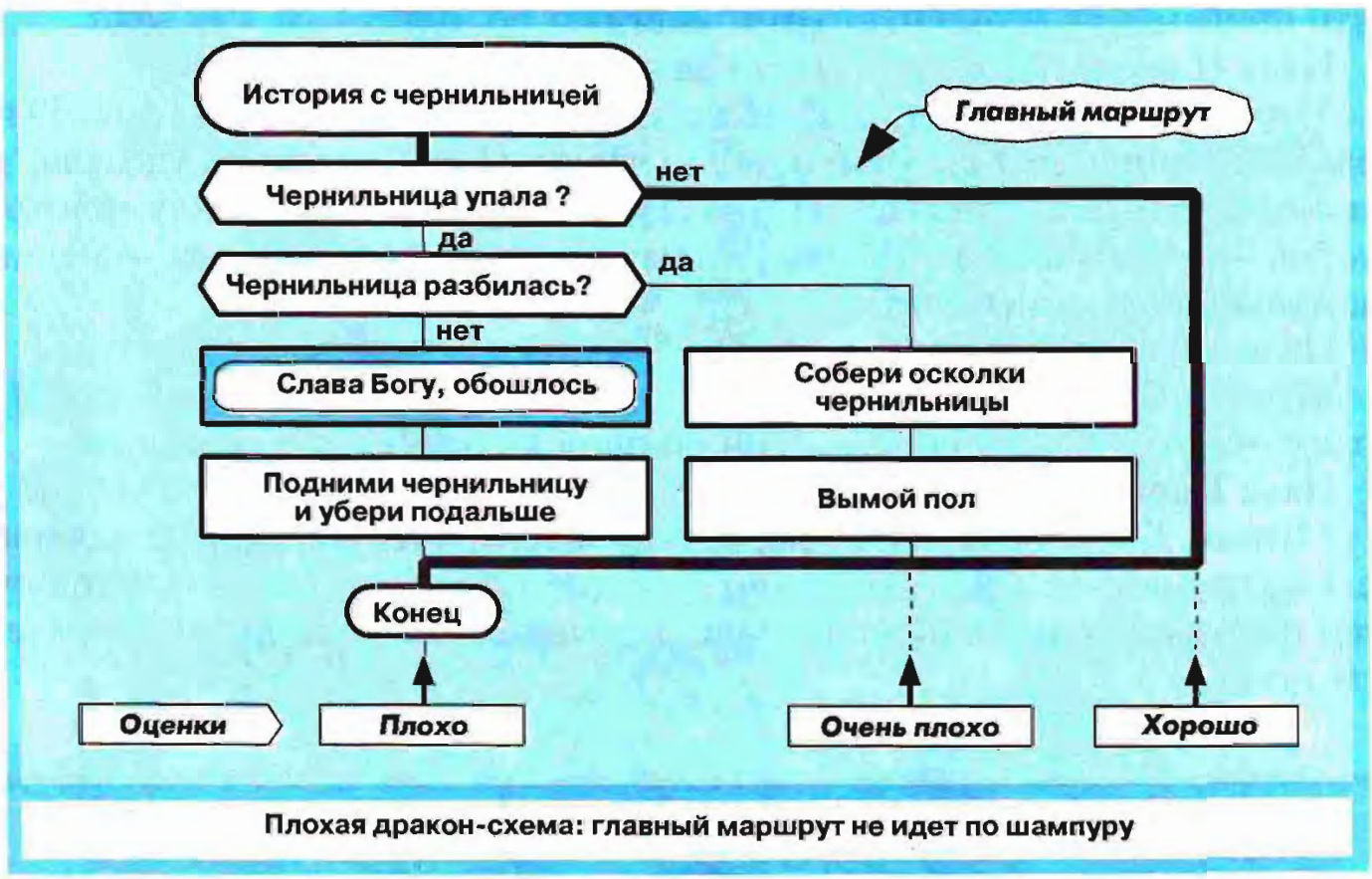
Задание Миши Проверялкина

1. Докажи, что четыре алгоритма на рис. 39 *а*, *б*, *в*, *г* равносильны.



Очень плохая дракон-схема. В ней две ошибки:
- главный маршрут не идет по шампуру
- побочные маршруты не упорядочены

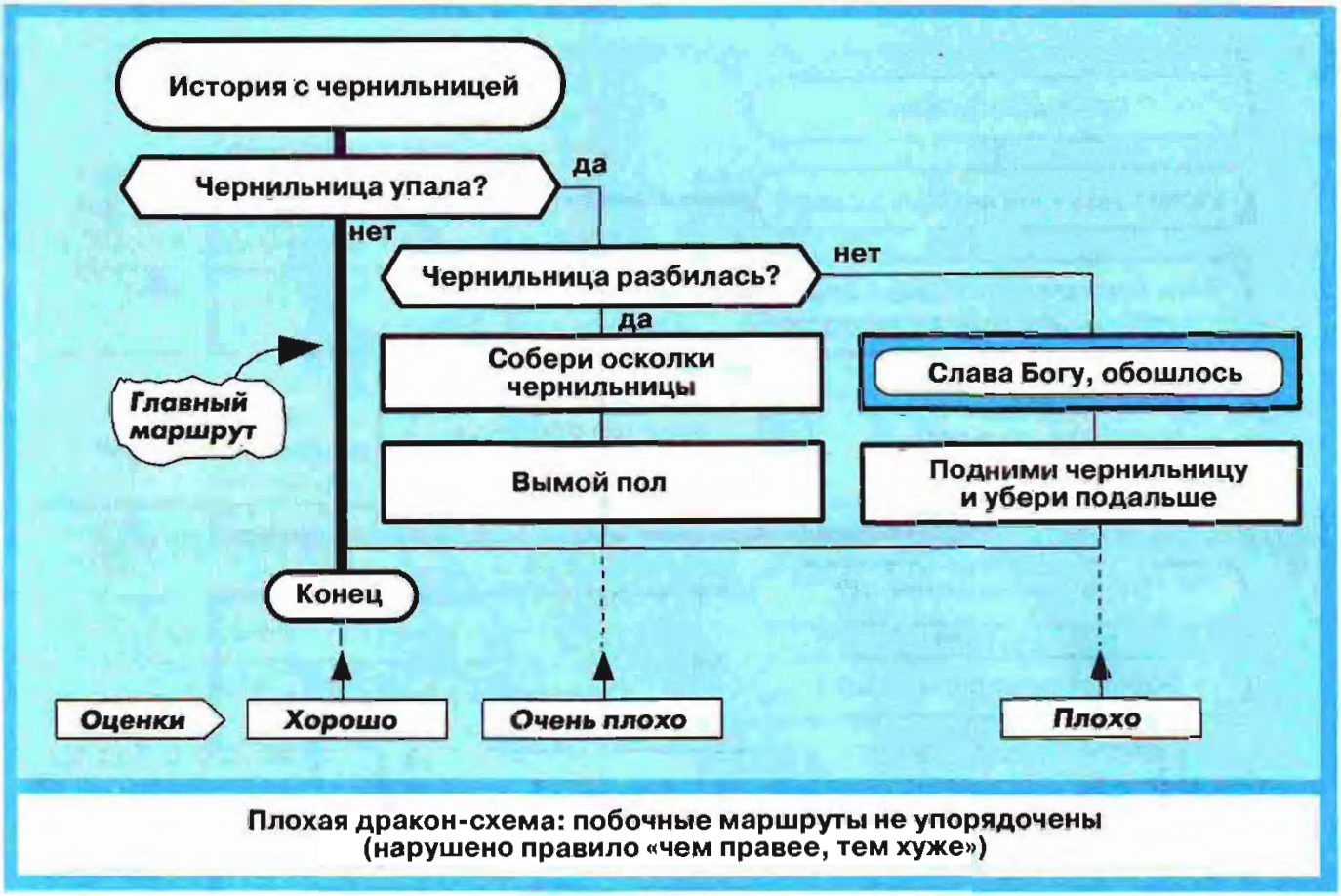
а



Плохая дракон-схема: главный маршрут не идет по шампуру

б

Рис. 39. Чтобы сделать алгоритм эргономичным, необходимо:
- **пустить главный маршрут по шампуру**
- **упорядочить побочные маршруты**



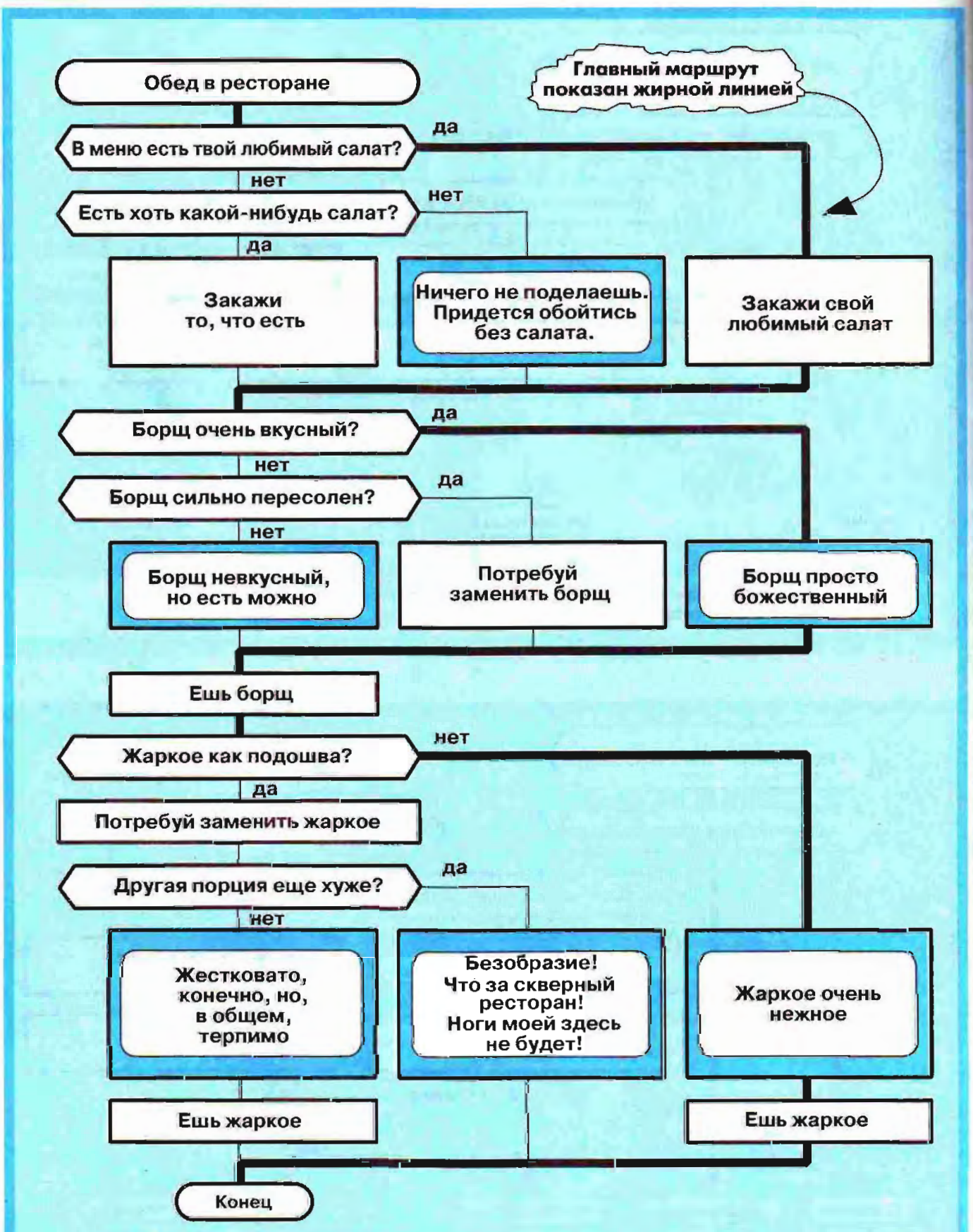
Плохая дракон-схема: побочные маршруты не упорядочены (нарушено правило «чем правее, тем хуже»)

в

Хорошая (эргономичная) дракон-схема:
- главный маршрут идет по шампуру
- побочные маршруты упорядочены

г

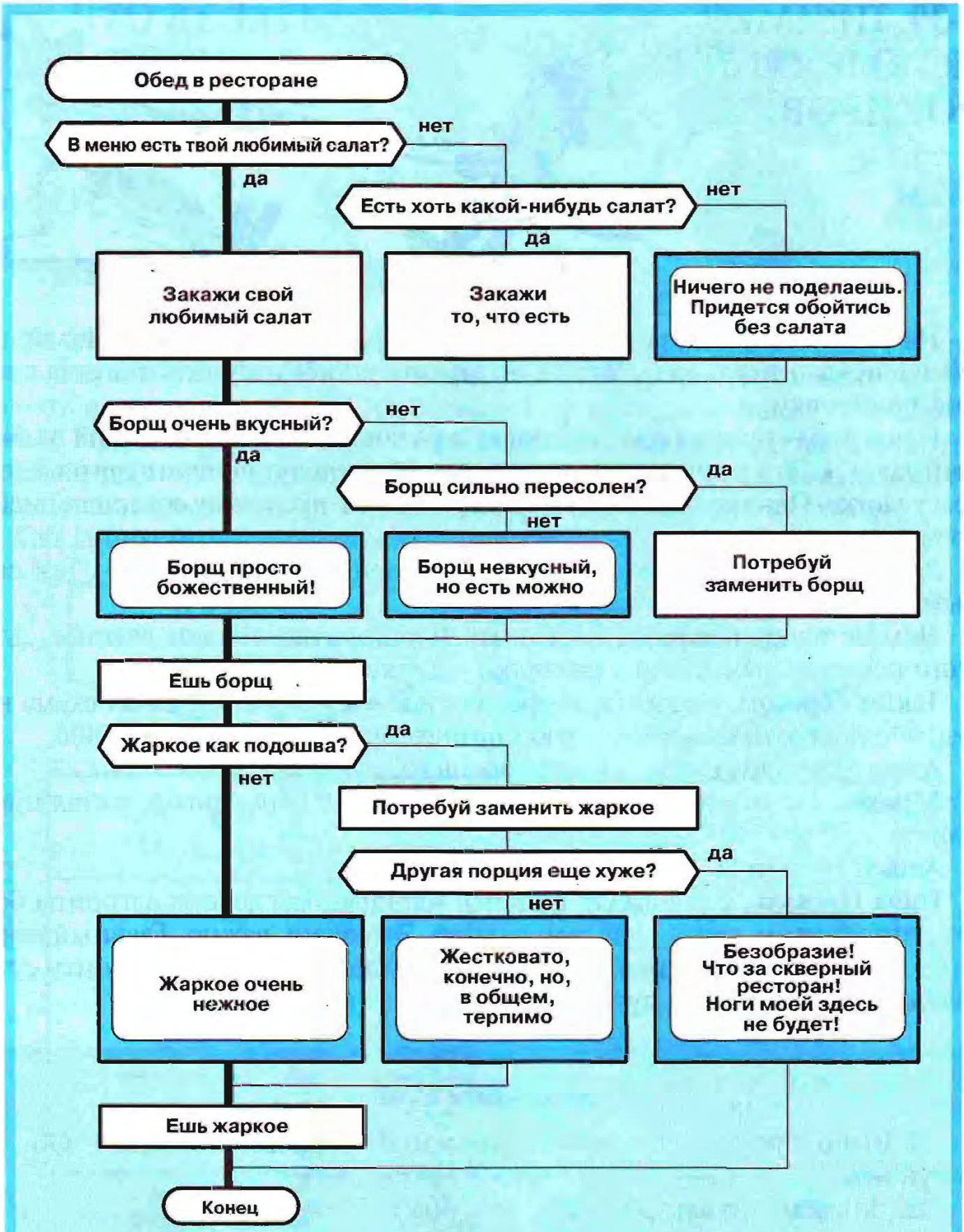
Рис. 39. Продолжение



Плохая дракон-схема. Ошибка в том, что нарушено правило «Главный маршрут должен идти по шампуру». Поэтому главный маршрут (жирная линия) все время петляет и делает зигзаги. Его трудно проследить взглядом.

а

Рис. 40. Плохую дракон-схему *(а)* всегда можно преобразовать в хорошую *(б)*



Хорошая дракон-схема.
Главный маршрут идет по шампуру.
Его легко проследить взглядом.

б

Рис. 40. Продолжение

§ 39. ПРИМЕР ИСПОЛЬЗОВАНИЯ РОКИРОВКИ

На рис. 40*а* показана неэргономичная дракон-схема (главный маршрут нарисован неудачно). Чтобы исправить ошибку, нужно три раза сделать рокировку.

Первый раз применяем рокировку в развилке «В меню есть ваш любимый салат?». Это позволяет пустить главный маршрут по шампуру на верхнем участке. Однако внизу главный маршрут по-прежнему совершает неоправданные зигзаги.

Затем делаем рокировку в развилке «Борщ очень вкусный?». Тем самым улучшаем среднюю часть схемы.

Нам осталось выпрямить главный маршрут на нижнем участке. Для этого переставляем плечи у развилки «Жаркое как подошва?».

Таким образом, в результате трех рокировок неэргономичная схема на рис. 40*а* превратилась в хорошую (эргономичную) схему на рис. 40*б*.

Алина. Хорошую? А что в ней хорошего?

Мурзик. Ты что, не видишь: главный маршрут был кривой, а стал прямой.

Алина. Ну и что?

Папа Циркуль. Выпрямляя главный маршрут, мы делаем алгоритм более наглядным и легким для понимания. Это очень важно. Главный маршрут — путеводная нить алгоритма, позволяющая быстрее уяснить суть дела и не заблудиться в путанице развилок.

Вопросы Саши Ехидного

1. Выполняется ли на рис. 40*б* правило «Чем правее, тем хуже»? Обоснуй ответ.
2. Докажи, что алгоритмы на рис. 40*а* и *б* равносильны.

§ 40. ЧТО ДЕЛАТЬ, ЕСЛИ ПРАВИЛО «ЧЕМ ПРАВЕЕ, ТЕМ ХУЖЕ» НЕ РАБОТАЕТ?

Из физики известно: если начальная скорость ракеты меньше 7,8 километров в секунду (км/с), она непременно упадет на Землю. Если же разогнать ее чуть сильнее, но не больше 11,2 км/с, она станет спутником Земли. При дальнейшем увеличении скорости ракета превратится в спутник Солнца. А если скорость превысит 16,4 км/с, ракета навсегда простится с Солнечной системой и умчится в безбрежные просторы космоса.

Сказанное можно подытожить с помощью таблицы.

Скорость ракеты (*км/с*)	Куда полетит ракета?
$V < 7{,}8$	Ракета упадет на Землю
$7{,}8 \leqslant V < 11{,}2$	Ракета станет спутником Земли
$11{,}2 \leqslant V < 16{,}4$	Ракета станет спутником Солнца
$V \geqslant 16{,}4$	Ракета покинет Солнечную систему

Алгоритм для этой задачи показан на рис. 41.

Мы знаем, что каждой вертикальной линии на дракон-схеме соответствует свой маршрут, причем вертикали следует рисовать не хаотично, а упорядоченно. До сих пор мы пользовались правилом «Чем правее, тем хуже». Однако здесь оно не имеет смысла. Поэтому на рис. 41 выбрано другое правило: «Чем правее, тем больше скорость ракеты».

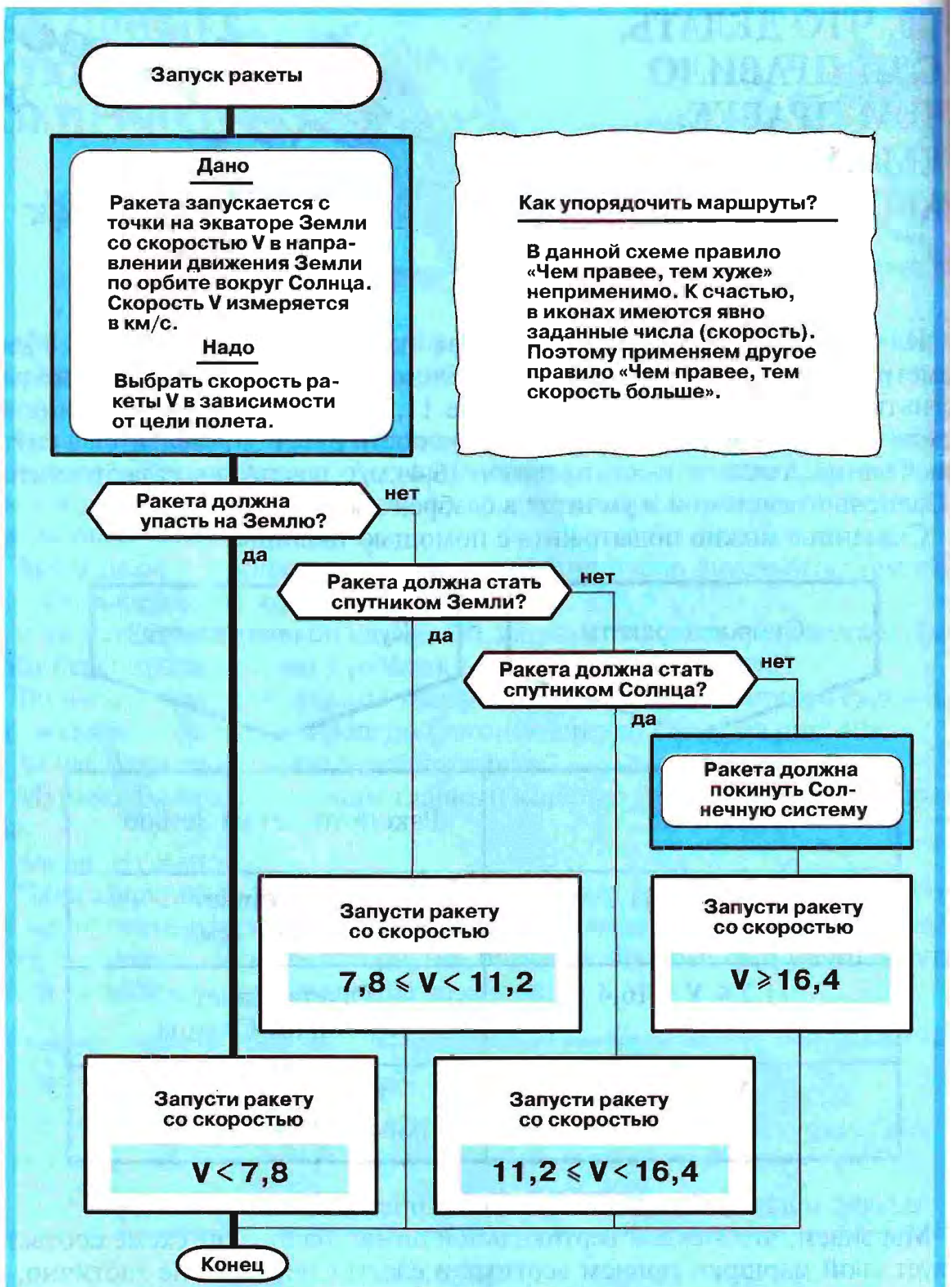


Рис. 41. Маршруты упорядочены слева направо согласно правилу «Чем правее, тем скорость больше».

Мурзик. Как это понимать?

Папа Циркуль. Проведи взглядом по схеме слева направо. Ты увидишь, что от маршрута к маршруту скорость неуклонно возрастает. Первый слева маршрут самый медленный. На втором маршруте скорость больше. На третьем — еще больше. Наконец, четвертый (самый правый) маршрут описывает ситуацию, когда ракета с огромной скоростью улетает за пределы Солнечной системы.

§ 41. ЧТО ТАКОЕ ОБЪЕДИНЕНИЕ?

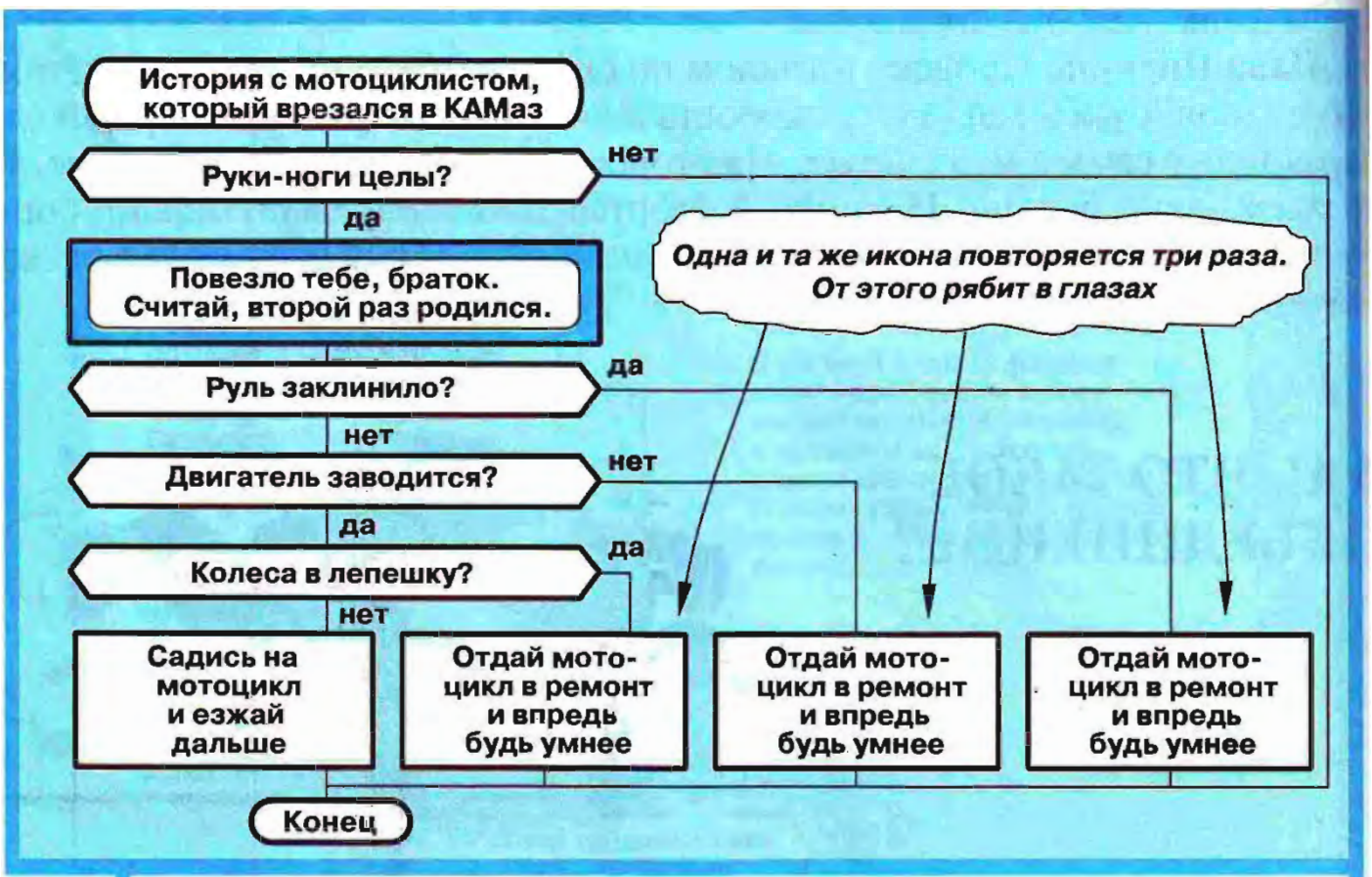
Иногда бывает, что в дракон-схеме иконы повторяются. Например, на рис. 42*а* икона «Отдай мотоцикл в ремонт и впредь будь умнее» встречается три раза. Это плохо: навязчивые повторения раздражают читателя, которому приходится тратить лишнее время и несколько раз читать одно и то же. К счастью, некоторые повторы можно устранить. Такая возможность появляется, если одинаковые иконы находятся рядом, причем их выходы соединены между собой (рис. 42*а*). В этом случае действует правило: «Повторы запрещены». Устранение повторов производится с помощью вертикальной линии (рис. 42*б*) и называется *вертикальным объединением*.

Мурзик. Почему такое название — «объединение»?

Папа Циркуль. Потому что несколько одинаковых икон объединяются в одну. Заметь, как это делается: *сначала входы икон объединяются вертикальной линией, после чего удаляются все одинаковые иконы, кроме крайней левой.*

Рассмотрим задачу на рис. 43*а*, где также имеются повторы, подлежащие устранению. Сначала устраним повторение иконы «Съешь кашу» и получим схему на рис. 43*б*. В этом случае для исключения повторов используется не вертикальная, а горизонтальная линия (рис. 43*б*). Соответствующая операция называется *горизонтальным объединением*.

Затем исключим повторение развилки «Есть можно?». Окончательный результат показан на рис. 43*в*. Легко заметить, что схема на рис. 43*в* более удобна и занимает меньше места, чем исходная схема на рис. 43*а*. Самое главное, она стала проще и намного понятнее.



а Плохая дракон-схема. Нарушено правило «Повторы запрещены»

б Хорошая (эргономичная) дракон-схема. Повторы исключены с помощью операции «Вертикальное объединение».

Рис. 42. Операция «Вертикальное объединение» позволяет устранить ненужные повторы и превратить плохую (неэргономичную) дракон-схему в хорошую (эргономичную)

Что такое повтор — **Ситуация, когда на дракон-схеме одинаковые иконы повторяются несколько раз**

Запомни — **Повторы — это плохо. Их желательно устранить**

Задания Миши Проверялкина

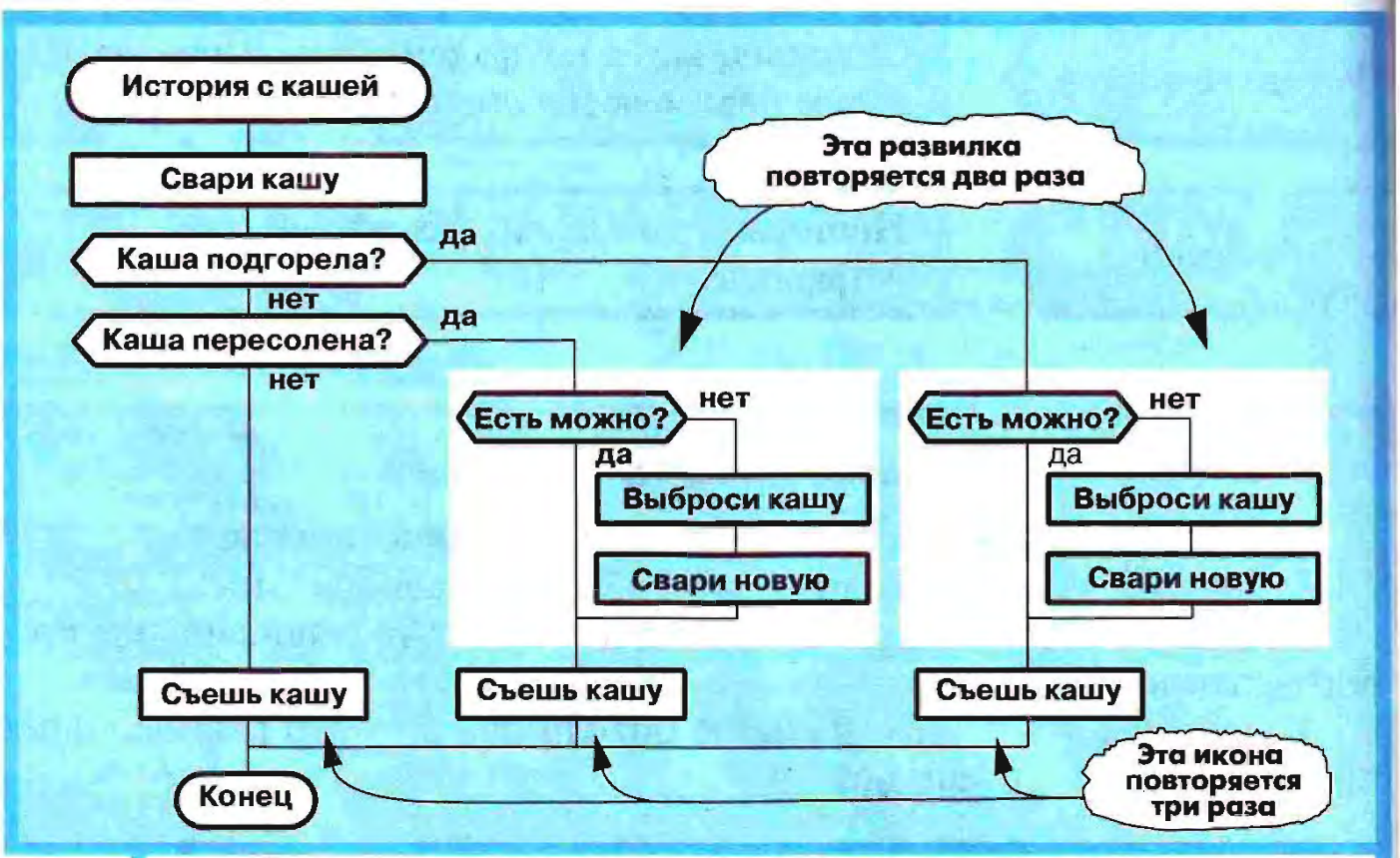
1. Докажи, что алгоритмы на рис. 42 *а* и *б* равносильны.
2. Докажи, что алгоритмы на рис. 43 *а*, *б*, *в* равносильны.
3. Докажи, что вертикальное объединение — это равносильное преобразование алгоритмов.
4. Докажи, что горизонтальное объединение — это равносильное преобразование алгоритмов.

§ 42. КАКАЯ ОШИБКА ПОДСТЕРЕГАЕТ НАС ПРИ ОБЪЕДИНЕНИИ?

Папа Циркуль. Устрани повторы на рис. 44*а*.

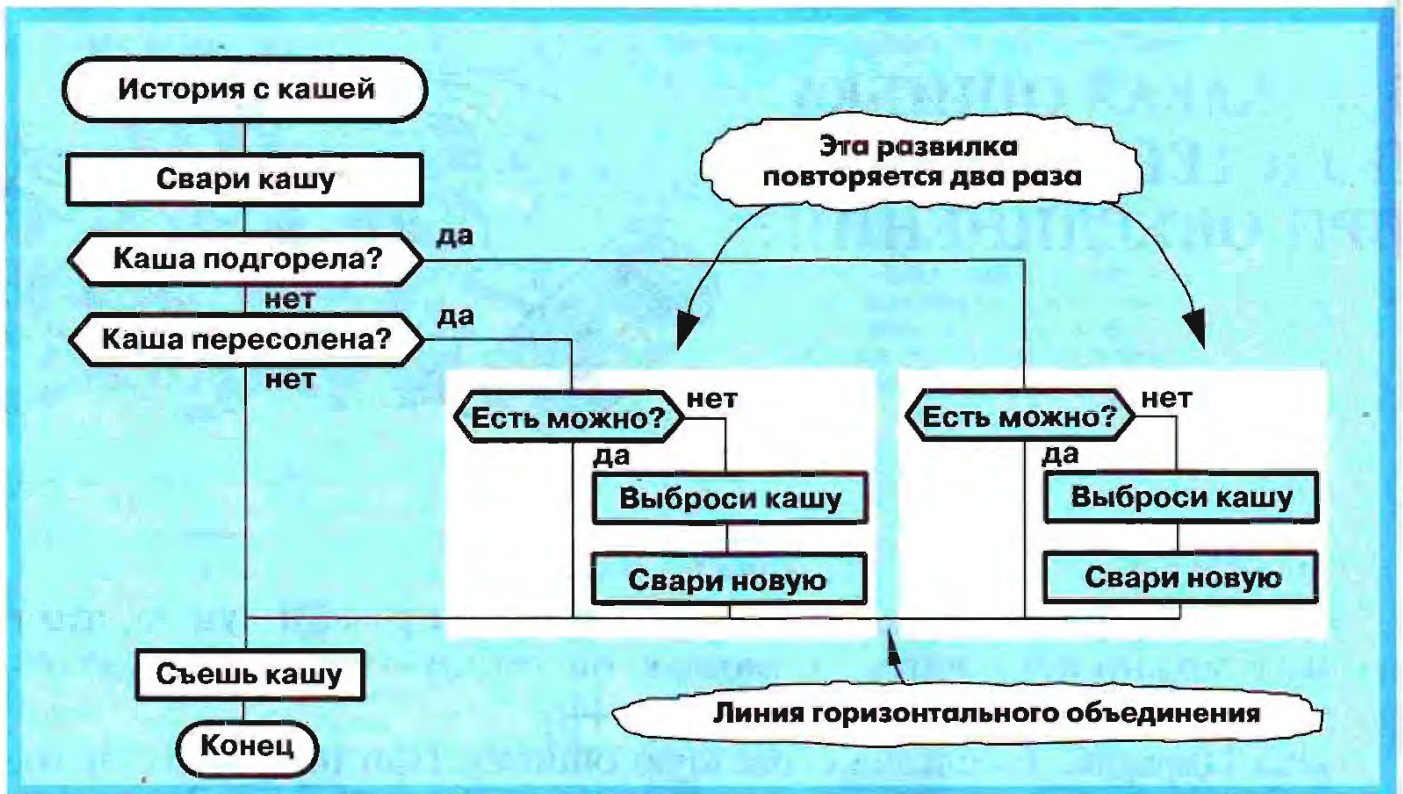
Мурзик. На рис. 44*а* икона F повторяется три раза. Я думаю, что две иконы F можно исключить с помощью операции «горизонтальное объединение». Результат я изобразил на рис. 44*б*.

Папа Циркуль. Ты сделал серьезную ошибку. При рисовании дракон-схем действует правило: *пересечения запрещены*. А у тебя две линии пересекаются в точке X. Запомни: разрешается объединять не любые одинаковые иконы, а только *соседние*. Обрати внимание: на рис. 44*а* из трех одинаковых икон F только две правые являются соседними. А третья (крайняя левая) отделена от них иконой H. Поэтому она не может участвовать в объединении. Правильный ответ показан на рис. 44*в*.



а

Очень плохая дракон-схема. Правило «Повторы запрещены» нарушено в двух местах



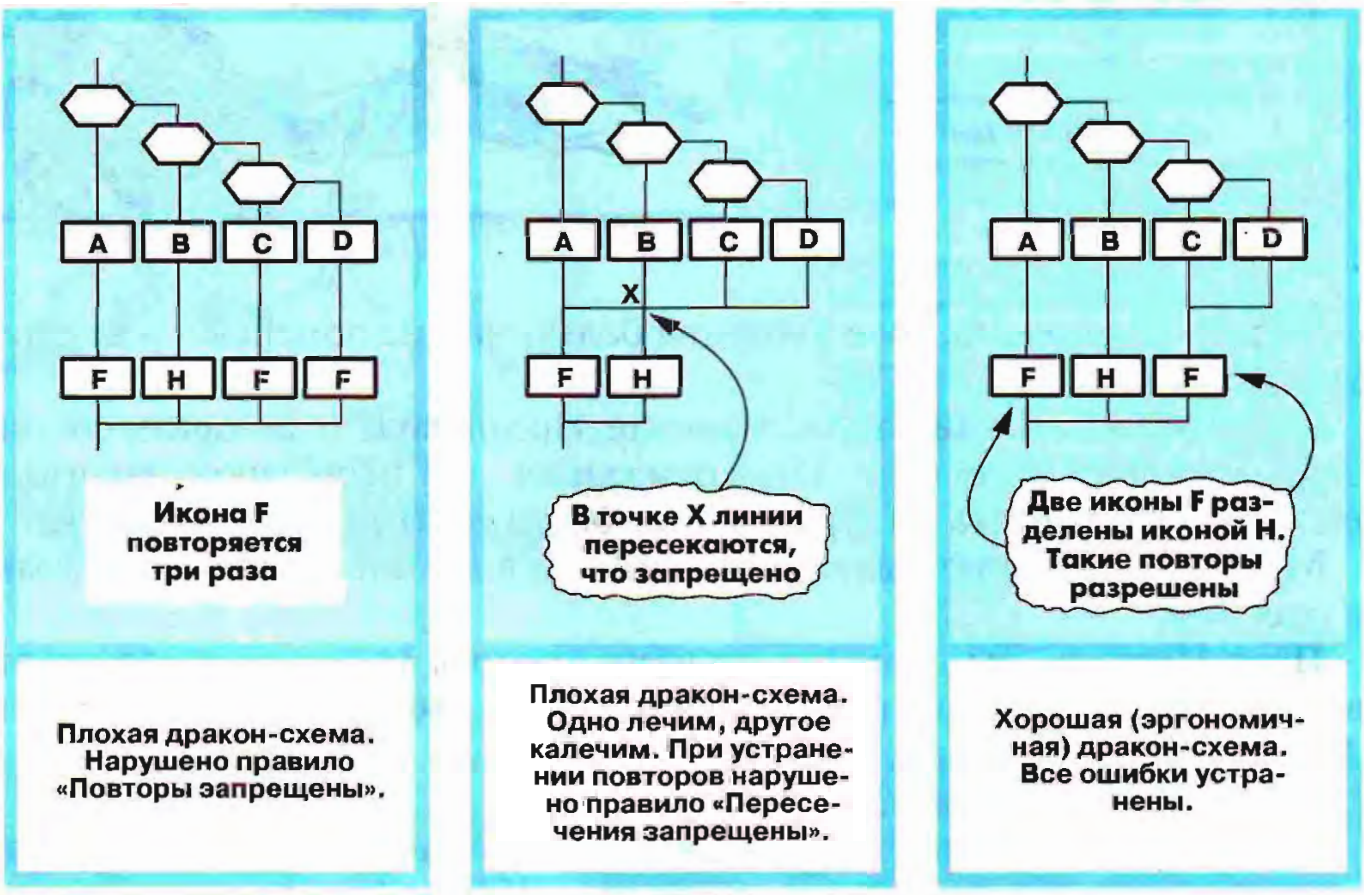
б

Плохая дракон-схема. Нарушено правило «Повторы запрещены». Данная схема получена из схемы а в результате горизонтального объединения трех икон «Съешь кашу».

Рис. 43. Последовательное применение операций «Горизонтальное объединение» и «Вертикальное объединение» позволяет устранить ненужные повторы и превратить очень плохую (неэргономичную) дракон-схему в хорошую (эргономичную)

в Хорошая (эргономичная) дракон-схема. Данная схема получена из схемы *б* в результате вертикального объединения двух развилок «Есть можно?»

Рис. 43. Продолжение



а Плохая дракон-схема. Нарушено правило «Повторы запрещены».

б Плохая дракон-схема. Одно лечим, другое калечим. При устранении повторов нарушено правило «Пересечения запрещены».

в Хорошая (эргономичная) дракон-схема. Все ошибки устранены.

Рис. 44. Устраняя повторы, следи, чтобы не появились пересечения

Правило	**В дракон-схеме пересечения линий запрещены**
Указание	**Выполняя вертикальное и горизонтальное объединение, следи, чтобы в схеме не появились пересечения**

Задание Миши Проверялкина

1. Докажи, что алгоритмы на рис. 44 *а*, *б*, *в* равносильны.

§ 43. ЗАЧЕМ НУЖНА ИКОНА «ВСТАВКА»?

Мурзик. Что делать, если алгоритм большой и не помещается на листе бумаги?

Папа Циркуль. Давай подумаем вместе. Предположим, алгоритм состоит из четырнадцати икон (рис. 45), а бумажный лист небольшой, так что на нем можно разместить по вертикали не более десяти икон. Как быть?

Мурзик. Предлагаю расположить иконы в два столбца, как показано на рис. 46 *а*.

Папа Циркуль. Ты допустил ошибку, которая называется «движение вверх». Запомни: дракон-поезд не может ехать вверх. А у тебя на участке АВ он едет именно вверх.

Мурзик. Что же делать?

Папа Циркуль. Надо использовать прием под названием «Разрежь великана». Чтобы алгоритм-великан на рис. 45 поместился на маленьком листе, надо разрезать его на части. Делается это так.

Выбросим из алгоритма несколько икон, а вместо них вставим икону-заместитель, которая называется *вставка* (рис. 46*б*). Вставка нужна, чтобы напомнить об изъятых иконах. Вставка занимает мало места — намного меньше, чем выброшенные иконы, поэтому алгоритм становится короче.

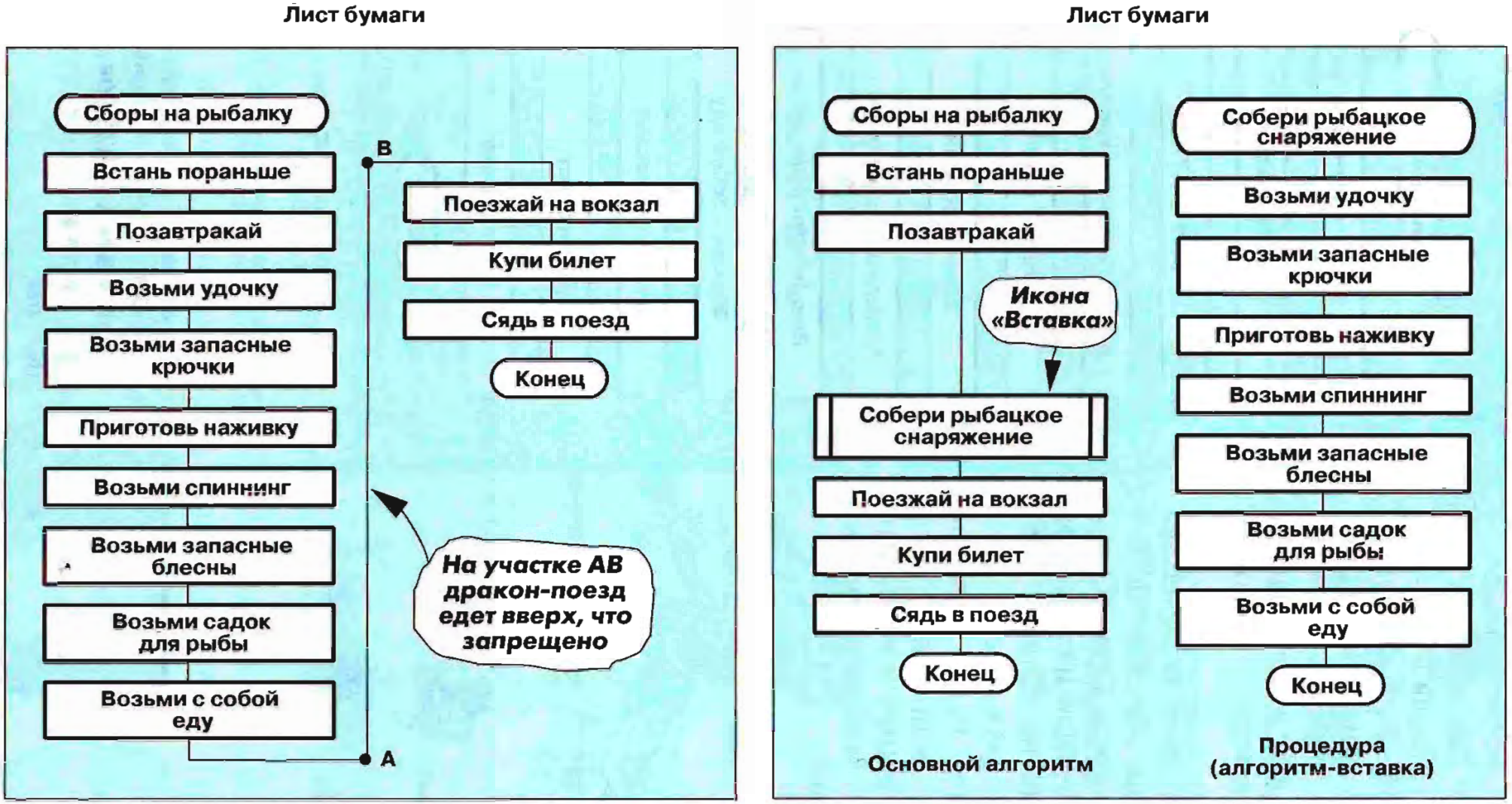
Разумеется, выброшенные иконы не пропадают, они образуют новый алгоритм — *процедуру*. Обрати внимание, икона «Вставка» и процедура имеют одинаковое название — «Собери рыбацкое снаряжение».

Таким образом, использование приема «Разрежь великана» приводит к тому, что длинный исходный алгоритм (рис. 45) разбивается на два коротких (рис. 46*б*).

Рис. 45. Алгоритм «Сборы на рыбалку»

Что такое икона «Вставка»

Знак, показывающий, что в данном месте из алгоритма изъяли кусок, который нарисовали в другом месте в виде процедуры



Плохая дракон-схема. В ней есть ошибка, которая называется «Движение вверх»

Хорошая дракон-схема. Для исправления ошибки исходный алгоритм разделен на две части:
- основной алгоритм;
- процедура

а *б*

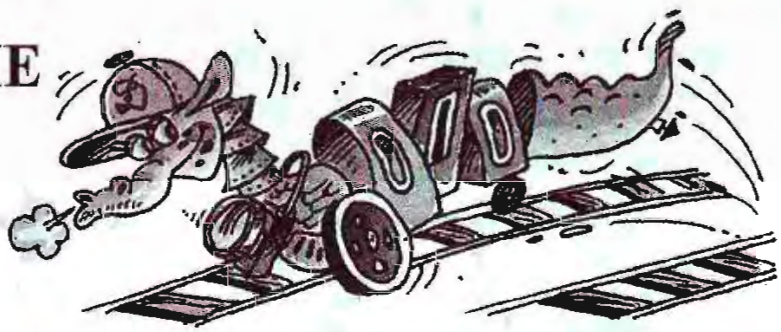
Рис. 46. Как исправить ошибку «Движение вверх»?

Что такое процедура → **Это часть исходного алгоритма, которая изъята из него и заменена иконой «Вставка»**

Задачи Миши Проверялкина

1. Разбей алгоритм на рис. 45 на две части так, чтобы икона «Возьми с собой еду» оказалась в основном алгоритме. Нарисуй основной алгоритм и процедуру.
2. Разбей алгоритм на рис. 45 на три части. Три последние иконы на рис. 45 преврати в процедуру. Нарисуй основной алгоритм и две процедуры.
3. Разбей алгоритм на рис. 15 по смыслу на части. Нарисуй основной алгоритм и процедуру.
4. Разбей алгоритмы на рис. 2 и 3 на части. Используй одинаковые процедуры в обоих алгоритмах. Нарисуй основные алгоритмы и процедуры.
5. В алгоритме на рис. 6 замени развилку на вставку и убери комментарий. Нарисуй основной алгоритм и процедуру.
6. В алгоритме на рис. 19 замени развилки на вставки. Нарисуй основной алгоритм и процедуры.
7. Разбей алгоритм на рис. 40 на части. Нарисуй основной алгоритм и процедуры.

§ 44. АКРОБАТИЧЕСКИЕ ПРЫЖКИ ДРАКОН-ПОЕЗДА, ИЛИ КАК ВЗАИМОДЕЙСТВУЮТ АЛГОРИТМЫ?



Итак, мы разделили исходный алгоритм (рис. 45) на две части (рис. 46*б*). Эти части (основной алгоритм и процедура) работают совместно и решают общую задачу. При этом они взаимодействуют между собой.

Как происходит это взаимодействие? Рассмотрим маршрут движения дракон-поезда, показанный на рис. 47. Дракон-поезд отправляется со станции «Сборы на рыбалку» и сначала движется по основному алгоритму. Однако, добравшись до иконы-вставки «Собери рыбацкое снаряжение», он прекращает движение вперед и начинает вести себя необычно. Образно говоря, он сходит с рельсов, совершает прыжок и перелетает на другую

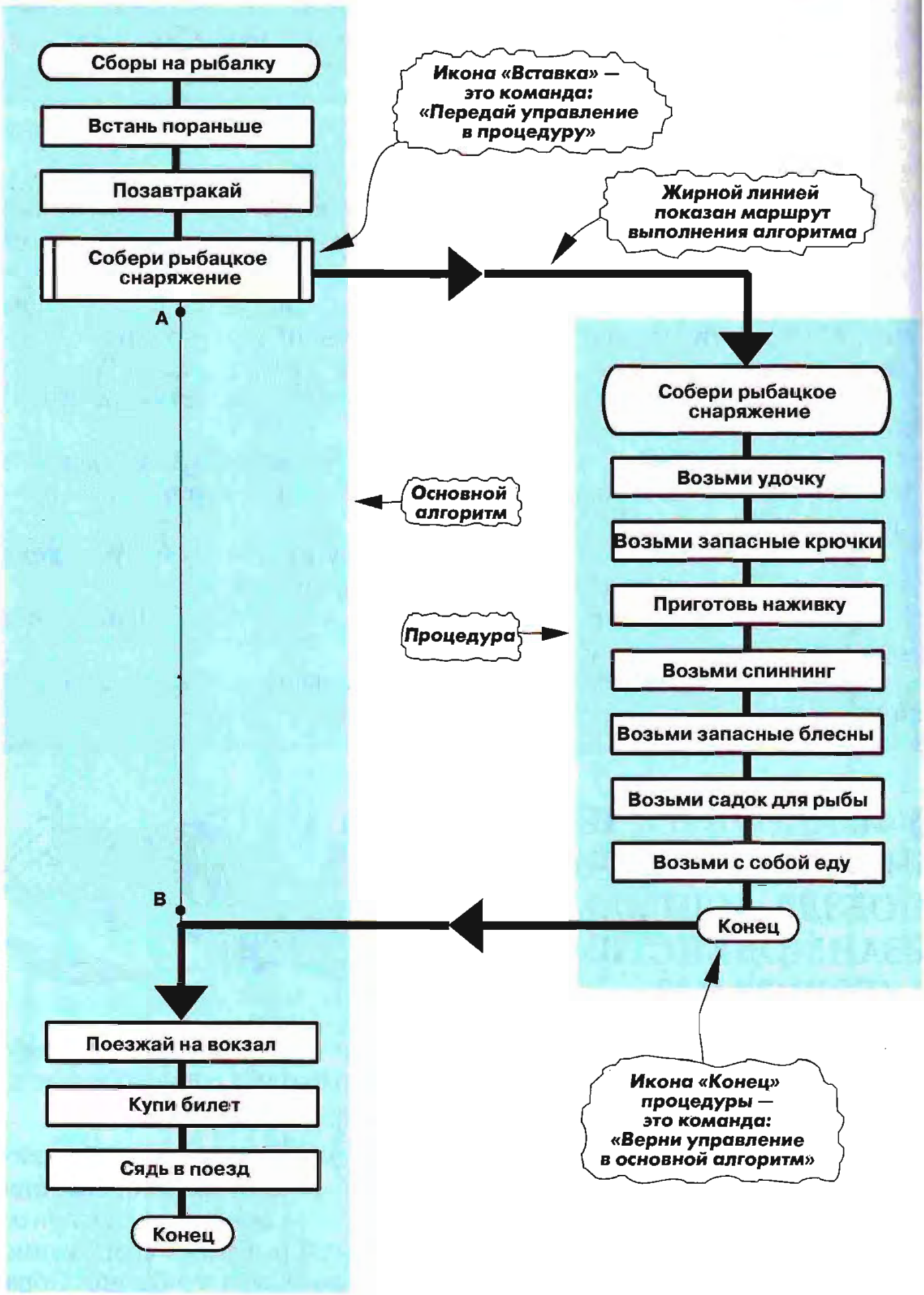


Рис. 47. Как взаимодействуют основной алгоритм и процедура?

железную дорогу — на станцию с точно таким же названием «Собери рыбацкое снаряжение» (рис. 47).

Мурзик. Почему?

Папа Циркуль. Потому что икона-вставка — это команда: «Передай управление в процедуру».

Мурзик. Что это значит?

Папа Циркуль. Это приказ исполнителю (роботу): брось основной алгоритм и начни выполнять процедуру.

Мурзик. Понятно. А что дальше?

Папа Циркуль. Подчиняясь этой команде, робот-исполнитель прекращает читать основной алгоритм и начинает выполнять процедуру. Иными словами, дракон-поезд, приземлившись в начале процедуры, встает на рельсы и едет через станции «Возьми удочку», «Возьми запасные крючки» и т. д. Однако, добравшись до конца процедуры, он снова совершает прыжок и возвращается в основной алгоритм.

Мурзик. Почему?

Папа Циркуль. Потому что икона «Конец» процедуры — это команда: «Верни управление в основной алгоритм».

Мурзик. Что это значит?

Папа Циркуль. Это приказ роботу: брось процедуру и продолжай выполнять основной алгоритм. Подчиняясь команде, робот прекращает читать процедуру, возвращается в основной алгоритм и, как ни в чем не бывало продолжает исполнять его с той точки, на которой остановился, а именно — с точки В (рис. 47).

Мурзик. А как ведет себя дракон-поезд?

Папа Циркуль. Встав на рельсы в точке В, дракон-поезд идет через станции «Поезжай на вокзал», «Купи билет» и т. д. и попадает на станцию «Конец». На этом совместная работа двух алгоритмов заканчивается.

§ 45. КАК ДВА АЛГОРИТМА УПРАВЛЯЮТ ОДНИМ РОБОТОМ?

Алина. Нельзя ли рассказать про взаимодействие алгоритмов как-нибудь попроще?

Папа Циркуль. Пожалуйста, объясняю на пальцах. Основной алгоритм — это командир, процедура — это солдат. Командир и солдат управляют роботом поочередно: когда командир отдает команды, солдат молчит. И наоборот.

Мурзик. Почему?

Папа Циркуль. Потому что если они будут галдеть и перебивать друг друга, получится путаница.

Мурзик. Какая путаница?

Папа Циркуль. Представь, что ты слуга двух господ. Один говорит в левое ухо: беги туда! А другой кричит в правое: нет, стой на месте! Что ты будешь делать?

Мурзик. Я скажу: господа, я не могу слушать сразу двоих. Говорите по очереди!

Папа Циркуль. Вот именно. Робот тоже слуга двух господ. Алгоритм-командир и алгоритм-солдат — вот два его начальника. Чтобы не было неприятностей, командир и солдат должны управлять роботом по очереди.

Мурзик. А как это сделать?

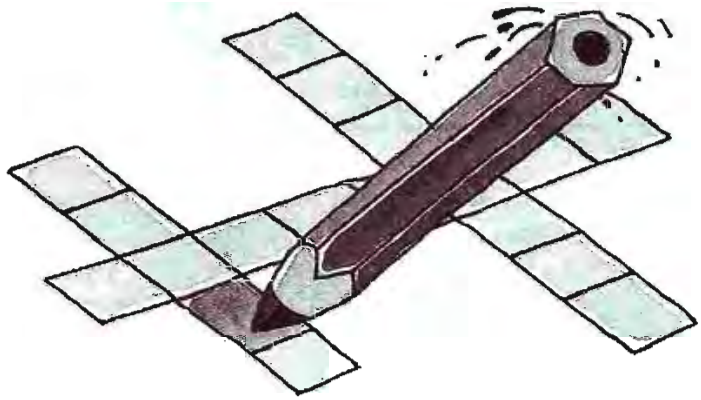
Папа Циркуль. Очень просто. Сначала командир говорит: «Беру управление на себя!» У солдата в это время рот на замке, а командир приказывает: «Встань пораньше! Позавтракай!» (рис. 47) Затем командир объявляет: «Передаю управление солдату!» С этого момента управлять роботом начинает солдат (процедура), а командир молчит. Когда солдат кончит работу, он докладывает: «Возвращаю управление командиру». После чего солдат умолкает и выходит из игры, а командир снова получает бразды правления и отдает последние команды: «Поезжай на вокзал! Купи билет! Сядь в поезд!»

Основной алгоритм и процедура должны действовать слаженно и согласованно. Как только кто-то из них *кончает* работу, он тут же подает другому сигнал, побуждающий того *начать* работу. В качестве такого сигнала используется икона «Вставка» (при прямой передаче управления) и икона «Конец» (при обратной передаче управления).

Могут ли основной алгоритм и процедура работать одновременно

Нет, они работают только по очереди

§ 46. ЧТО ТАКОЕ АЛГОРИТМИЧЕСКИЙ КРОССВОРД?



Обычный кроссворд — это загадка, в которой надо угадать слова. Алгоритмический кроссворд — тоже загадка, но совсем другая. Посмотри на рис. 48, 49, 50. На первых двух представлены алгоритмы, на третьем — слепыш. Рисунки 48 и 49 — это образец или, лучше сказать, подсказка для

АЛГОРИТМИЧЕСКИЙ КРОССВОРД №1

Рис. 48

Рис. 49

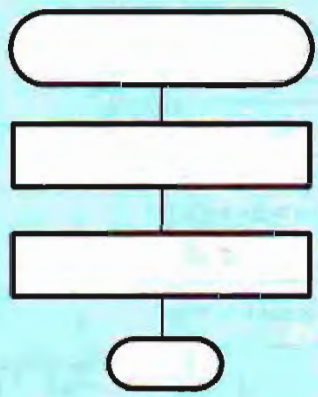


Рис. 50

Заполни рисунок 50 по аналогии

заполнения слепыша. Решить алгоритмический кроссворд — значит заполнить слепыш так, чтобы получился осмысленный, логичный алгоритм.

Задачи Миши Проверялкина

1. Реши алгоритмический кроссворд на рис. 50 и 53.
2. Реши алгоритмический кроссворд на рис. 56, 59, 62.
3. Придумай алгоритмический кроссворд из десяти икон и дай два примера его заполнения.

АЛГОРИТМИЧЕСКИЙ КРОССВОРД №2

Рис. 51

Рис. 52

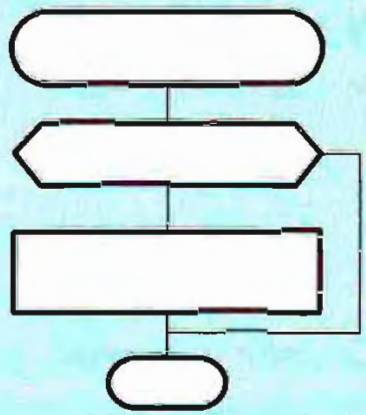


Рис. 53

Заполни рисунок 53 по аналогии

АЛГОРИТМИЧЕСКИЙ КРОССВОРД №3

Рис. 54

Рис. 55

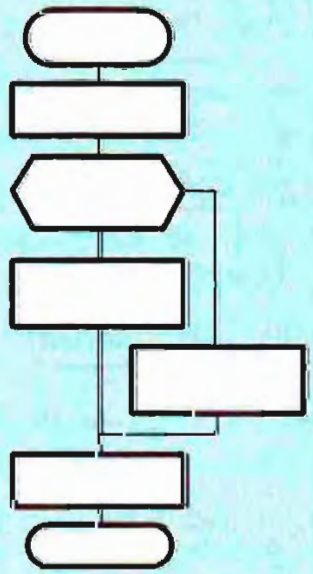


Рис. 56

Заполни рисунок 56 по аналогии

АЛГОРИТМИЧЕСКИЙ КРОССВОРД №4

Рис. 57

Рис. 58

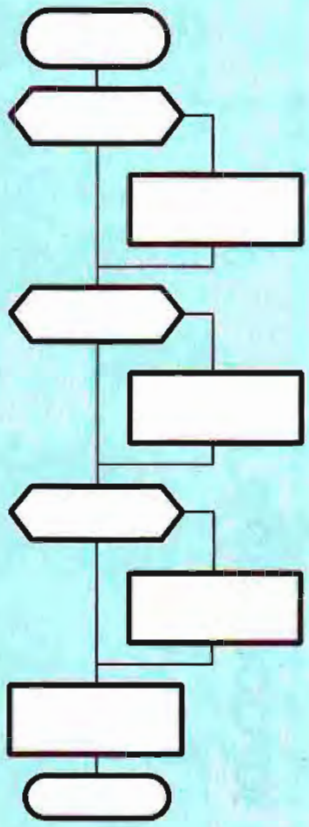
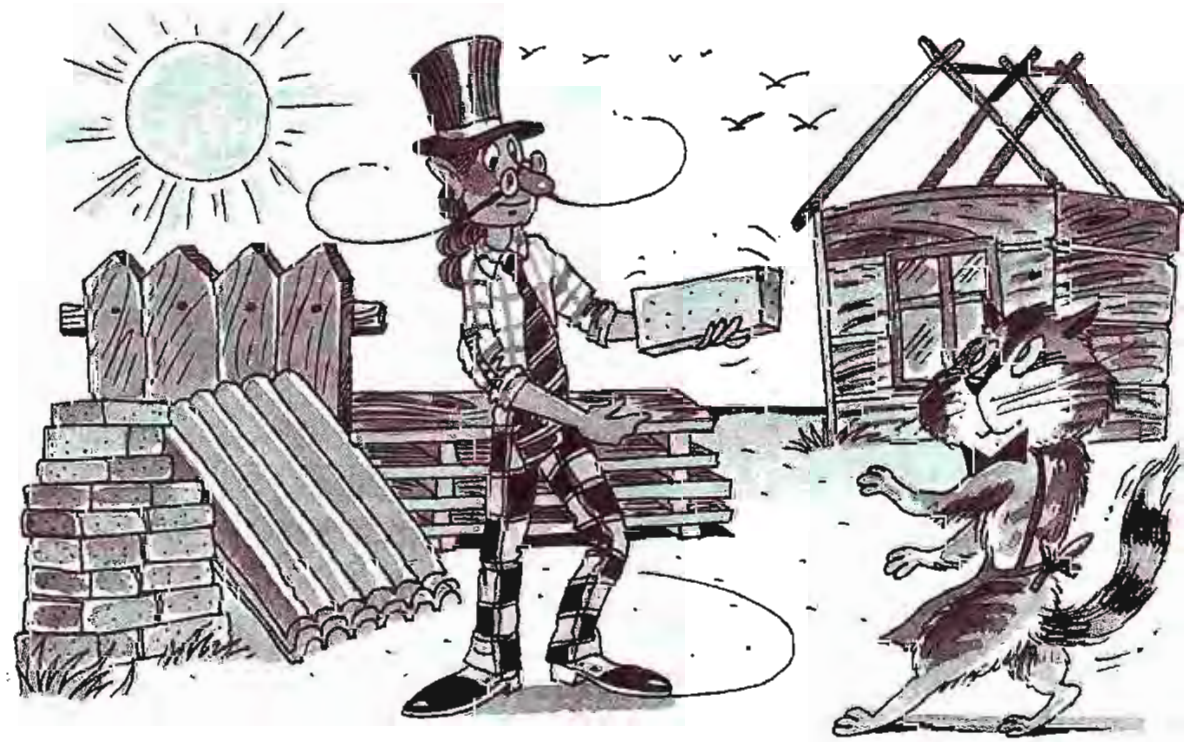


Рис. 59

Заполни рисунок 59 по аналогии



АЛГОРИТМИЧЕСКИЙ КРОССВОРД №5

Рис. 60

Рис. 61

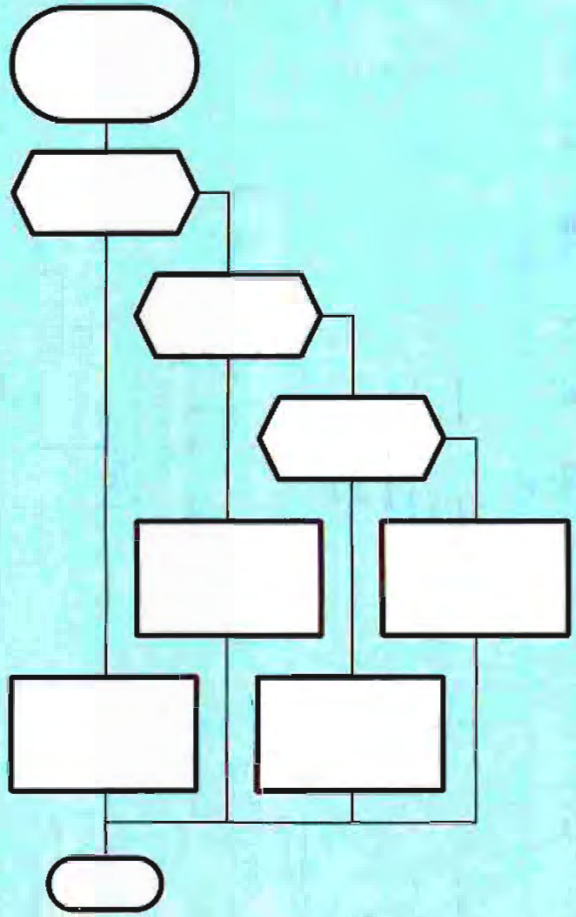


Рис. 62

Заполни рисунок 62 по аналогии

Глава четвертая

ПОЧЕМУ ЗМЕЙ ГОРЫНЫЧ БОИТСЯ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ?

§ 47. ЧТО ТАКОЕ ЦИКЛ?

Предположим, у Змея Горыныча три головы. Чтобы победить Змея, Илья Муромец — назовем так нашего робота — должен отсечь все головы, т. е. три раза исполнить команду «Отруби голову» (рис. 63). Если у Змея пять голов, эту команду придется написать пять раз (рис. 64). А если у Змея сто или тысяча голов? Эдак никакой бумаги не хватит!

Чтобы не писать слишком много одинаковых команд, нужно использовать цикл. *Цикл* — это повторяющееся исполнение одних и тех же команд. Фокус в том, что команда в алгоритме записывается один раз, а исполняется много раз — столько, сколько нужно.

Как работает цикл? Предположим, у Змея только одна голова, которую мы победили с помощью команды 1 (рис. 65). В этом случае на вопрос 2: «У Змея Горыныча остались живые головы?» — отвечаем «нет» и алгоритм заканчивается.

Если же у Змея есть уцелевшие головы, отвечаем «да» и, проходя через стрелку, снова попадаем на вход иконы 1. После чего рубим следующую голову. Еще раз задаем вопрос 2. Если у Змея по-прежнему остались головы, отвечаем «да» и вновь возвращаемся на вход иконы 1.

Таким образом, исполнение команд 1 и 2 может продолжаться и сто, и двести, и тысячу раз — до тех пор, пока наш алгоритм (управляющий роботом Ильей Муромцем) не отсечет у Змея все головы до последней.

Змей Горыныч и циклические алгоритмы

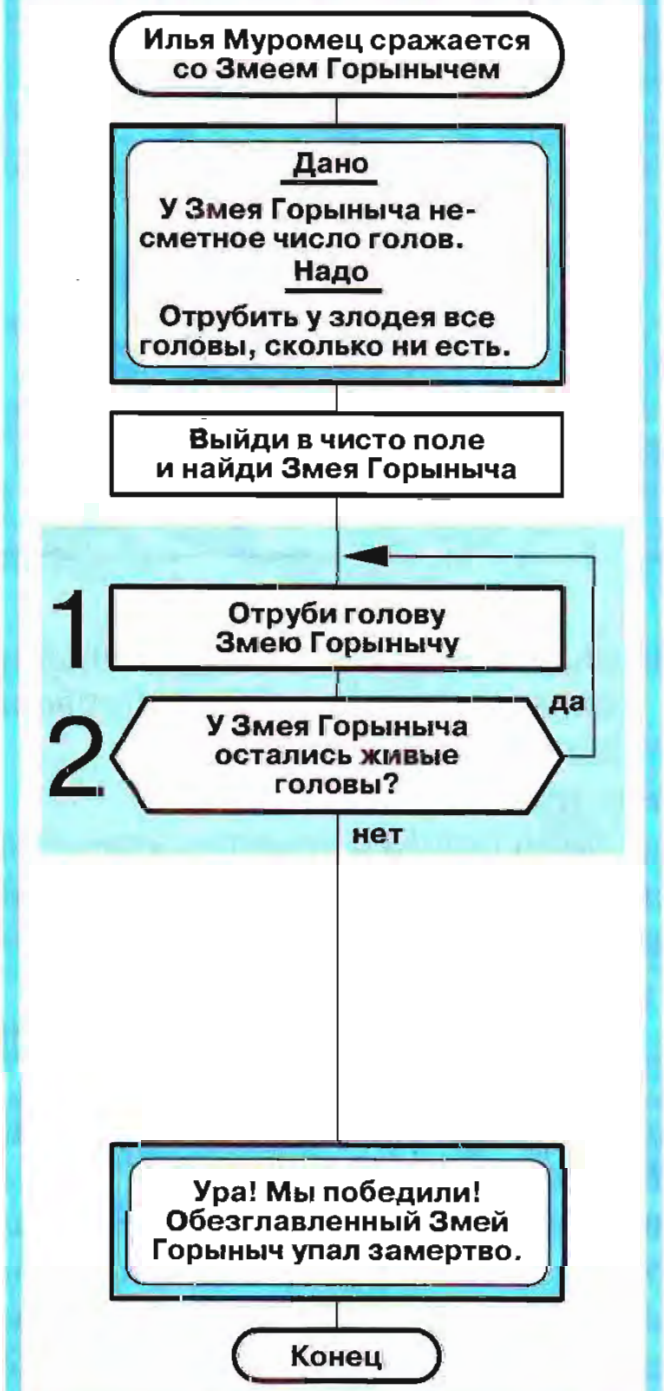
Это нециклический алгоритм

Рис. 63. Плохое (очень длинное) решение. Чтобы отрубить три головы, приходится писать три команды «Отруби голову»

Это нециклический алгоритм

Рис. 64. Плохое (очень длинное) решение. Чтобы отрубить пять голов, приходится писать пять команд «Отруби голову». А если у Змея сто голов?

Змей Горыныч
и циклические алгоритмы



Это циклический алгоритм

Рис. 65. Хорошее (короткое) решение: в алгоритме используется цикл. Циклический алгоритм позволяет отрубить любое число голов. При этом нужна всего одна команда «Отруби голову»

Это ЦИКЛ. Команды цикла (1 и 2) повторяются многократно. Они работают, пока выполняется УСЛОВИЕ ПРОДОЛЖЕНИЯ ЦИКЛА (см. внизу). Цикл прекращает работу, когда выполняется УСЛОВИЕ ОКОНЧАНИЯ ЦИКЛА (см. внизу).

УСЛОВИЕ ПРОДОЛЖЕНИЯ ЦИКЛА

У Змея Горыныча остались живые головы? = да

УСЛОВИЕ ОКОНЧАНИЯ ЦИКЛА

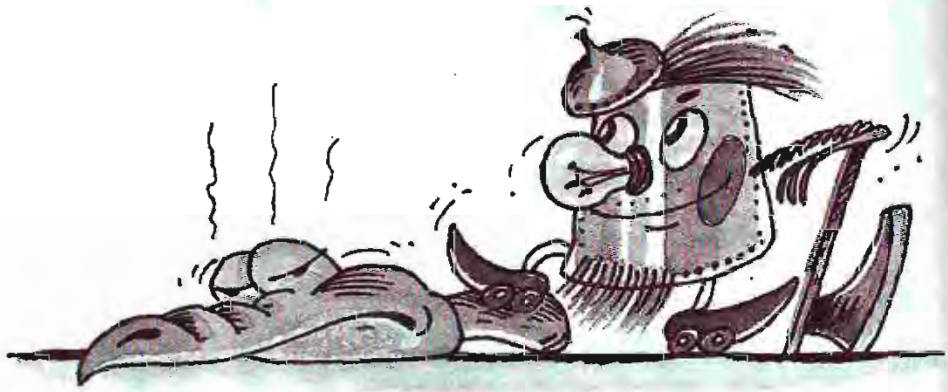
У Змея Горыныча остались живые головы? = нет

Цикл содержит условие цикла, тело цикла и цепь обратной связи ABCD

Рис. 66. Цикл, вырезанный (для наглядности) из рис. 65

Что такое цикл	**Группа команд, которые в алгоритме записываются один раз, а выполняются много раз — столько, сколько нужно**
Что такое циклический алгоритм	**Это алгоритм, в котором есть хотя бы один цикл**

§ 48. ЧТО ТАКОЕ УСЛОВИЕ?



Папа Циркуль. Мурзик, как ты думаешь, зачем нужна икона «Вопрос»?

Мурзик. Чтобы проверить условие. Например, на рис. 65 мы проверяем условие: «У Змея Горыныча остались живые головы?»

Папа Циркуль. А что значит *проверить условие*?

Мурзик. Это значит ответить либо да, либо нет. Мы говорим «да», если условие выполняется, т. е. если у Змея есть живые головы. Мы говорим «нет», если условие не выполняется (все головы уже лежат на траве).

Папа Циркуль. Что такое *условие*?

Мурзик. Это переменная величина, которая может принимать два значения: «да» или «нет».

Папа Циркуль. Молодец, все правильно. Самое интересное, что при выполнении цикла условие обычно изменяет свое значение.

Например, до начала работы алгоритма на рис. 65 условие имело значение «да» (у Змея есть голова). Однако через некоторое время, после того как Илья Муромец помахал своим боевым топором, условие изменилось и приобрело значение «нет» (потому что все змеиные головы уже валяются на земле). Это обстоятельство (изменение условия в процессе повторения цикла) имеет важное значение. Его можно использовать как сигнал, который говорит: «Хватит! Работа закончена!»

Мурзик. Ага, я догадался! По этому сигналу команда «Вопрос» производит переключение «железнодорожной стрелки»: правый путь закрывается, а нижний открывается (рис. 65). В итоге мы выходим из цикла и работа алгоритма заканчивается.

Что такое условие	**Это переменная величина, записанная в иконе «вопрос», которая принимает два значения: *да* или *нет***
Что делает команда «Вопрос»	**Проверяет условие и в зависимости от его значения (*да* или *нет*) выбирает один из двух путей исполнения алгоритма**

§ 49. ИЗ КАКИХ ЧАСТЕЙ СОСТОИТ ЦИКЛ?

На рис. 65 цикл состоит из двух команд:

Команда 1	Отруби голову Змею Горынычу
Команда 2	У Змея Горыныча остались живые головы?

Кроме того, в состав цикла входит цепь обратной связи ABCD (рис. 66), которая заканчивается стрелкой. Стрелка соединяется под прямым углом со входом иконы «Отруби голову Змею Горынычу».

Чтобы организовать цикл, правый выход иконы «Вопрос» нужно загнуть вверх и воткнуть стрелку в нужное место — например, на вход иконы «Действие».

Цикл состоит из команды «Вопрос» и последовательности действий, которая называется *телом цикла*. На рис. 66 тело цикла состоит из команды: «Отруби голову Змею Горынычу».

Назови составные части цикла	● **Команда «вопрос»** ● **Тело цикла** ● **Цепь обратной связи**
Что такое тело цикла	**Это команды, входящие в состав цикла, за исключением команды «Вопрос»**
Как организовать цикл в алгоритме	**Правый выход иконы «Вопрос» нужно загнуть вверх и воткнуть стрелку в нужное место, например, на вход иконы «Действие»**
Что такое цепь обратной связи	**Это путь, начинающийся у правого выхода иконы «Вопрос» и заканчивающийся стрелкой, присоединенной ко входу цикла**
Запомни	**Правило «Движение вверх запрещено» не распространяется на цепь обратной связи**

§ 50. ПРИМЕР: ЦИКЛ СО СДОБНЫМИ ПЛЮШКАМИ

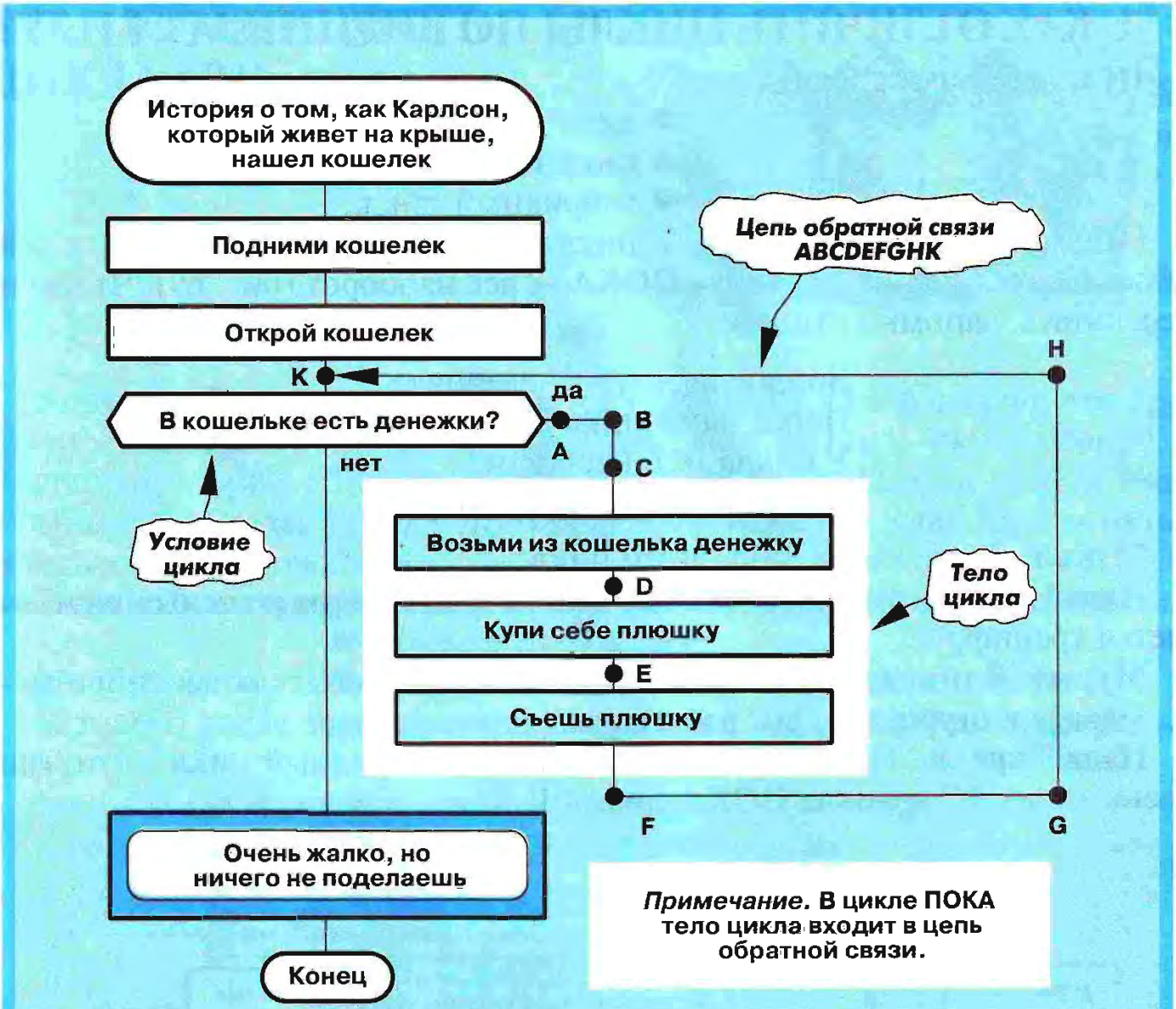
Однажды Карлсон, который живет на крыше, нашел кошелек и открыл его (рис. 67). А что случилось дальше? Здесь возможны варианты.

Вариант 1. Кошелек оказался пустым, поэтому Карлсону не удалось купить плюшку.

Вариант 2. В кошельке всего одна денежка, так что Карлсон смог купить только одну плюшку.

Вариант 3. В кошельке целая куча денег, поэтому Карлсон купил гору плюшек и наелся до отвала.

Таким образом, цикл на рис. 67 может либо ни разу не выполняться, либо выполняться один раз, либо много раз (два и более). Такой цикл имеет специальное название: «Цикл ПОКА».



УСЛОВИЕ ПРОДОЛЖЕНИЯ ЦИКЛА

В кошельке есть денежки?	=	да

УСЛОВИЕ ОКОНЧАНИЯ ЦИКЛА

В кошельке есть денежки?	=	нет

Рис. 67. Пример цикла ПОКА

§ 51. КАК ОТЛИЧИТЬ ЦИКЛЫ ПО ВНЕШНЕМУ ВИДУ?

Циклы бывают трех типов:

- цикл ДО,
- цикл ПОКА,
- гибридный цикл.

Отличить их очень просто. У цикла ДО вопрос рисуют внизу, а действие вверху (рис. 68). У цикла ПОКА — все наоборот (рис. 69). Чтобы не перепутать, запомни стишок.

Вопрос вверху — наверняка
Перед нами цикл ПОКА.
У цикла ДО, наоборот,
Вопрос внизу сидит, как крот.

Мурзик. А что такое гибридный цикл?

Папа Циркуль. Гибрид — это помесь. Гибрид лимона и апельсина называется грейпфрут.

Мурзик. Я тоже слышал: один агроном вывел замечательные гибриды — помедыни и огурбузы. Они растут прямо на асфальте.

Папа Циркуль. Не валяй дурака. В общем, гибридный цикл — это «помесь» цикла ДО и цикла ПОКА (рис. 70).

Рис. 68. Цикл ДО

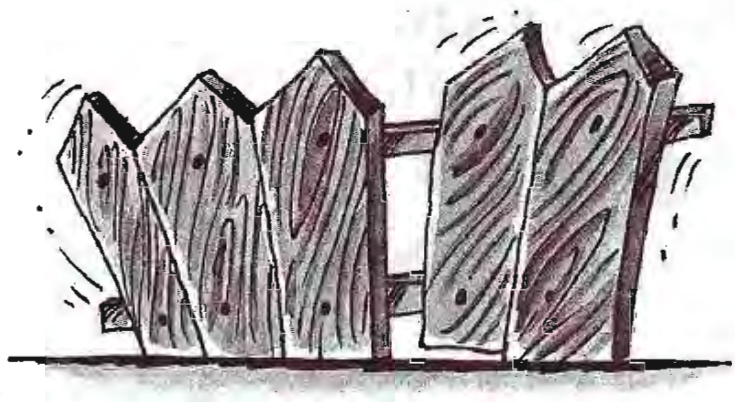
Рис. 69. Цикл ПОКА

Рис. 70. Гибридный цикл

Вопросы Саши Ехидного

1. Как называются циклы на рис. 65 и на рис. 67? Ответ обоснуй.
2. Как называется цикл на рис. 71? Ответ обоснуй.

§ 52. ОСОБЕННОСТЬ ЦИКЛА «ДО»



В цикле ДО действие выполняется *до* вопроса. Это означает, что дракон-поезд сначала проезжает икону «Действие», потом — икону «Вопрос». Например, на рис. 65 цикл начинается с действия «Отруби голову Змею Горынычу». И только после этого задается вопрос: «У Змея Горыныча остались живые головы?»

Такая же картина на рис. 71. В начале цикла выполняются два действия:

- Покрась одну доску
- Шагни вправо на ширину доски

И только после этого проверяется условие: «Все доски покрашены?»

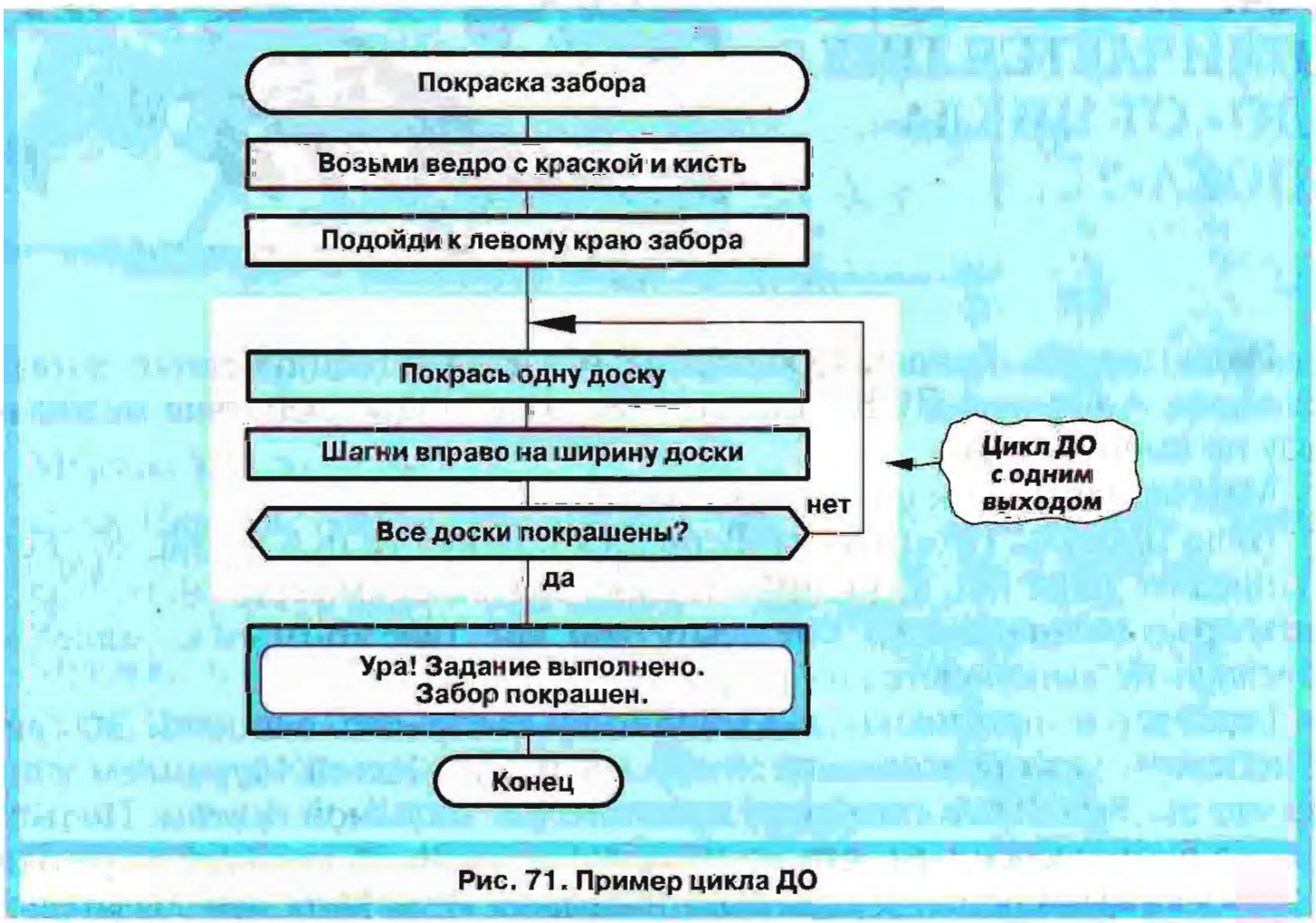


Рис. 71. Пример цикла ДО

§ 53. ОСОБЕННОСТЬ ЦИКЛА «ПОКА»

В цикле ПОКА действие либо вообще не выполняется, либо выполняется *после* вопроса. Поэтому дракон-поезд сначала обязательно едет через икону «Вопрос», а затем, если ответ благоприятный, — через икону «Действие». Например, на рис. 67 цикл начинается с вопроса: «В кошельке есть денежки?» И только после этого (при ответе «да») выполняются действия, образующие тело цикла:

- Возьми из кошелька денежку
- Купи себе плюшку
- Съешь плюшку

§ 54. ЧЕМ ОТЛИЧАЕТСЯ ЦИКЛ «ДО» ОТ ЦИКЛА «ПОКА»?

Папа Циркуль. В цикле ДО действие обязательно выполняется, хотя бы один раз. А в цикле ПОКА при некоторых условиях действие может ни разу не выполняться.

Мурзик. Нельзя ли объяснить подробнее?

Папа Циркуль. Пожалуйста. Вернемся к циклу ПОКА на рис. 67. Если в кошельке денег нет, из иконы «Вопрос» мы выходим через «нет», и алгоритм сразу заканчивается. Следовательно, действие «Возьми из кошелька денежку» не выполняется ни разу.

Рассмотрим противоположный пример, связанный с циклом ДО (рис. 65). Невозможно представить, чтобы на битву с Ильей Муромцем изначально прибыл Змей-инвалид, у которого нет ни одной головы. Поэтому можно быть уверенным, что до начала сражения по крайней мере одна голова у Змея непременно есть. Следовательно, Илья Муромец обязательно отсечет Змею хотя бы одну голову. Этот пример подтверждает, что в цикле ДО действие выполняется, как минимум, один раз.

§ 55. КАК СПАСТИ КАРЛСОНА ОТ ОБЖОРСТВА, ИЛИ ДОСРОЧНЫЙ ВЫХОД ИЗ ЦИКЛА

Карлсон, который живет на крыше, может съесть очень много плюшек. Наверное, штук сто. Или даже двести. Но не больше! Иначе он просто лопнет. А теперь предположим, что в кошельке, на его счастье (или беду), оказалось пятьсот монет. Как в этой ситуации будет работать алгоритм на рис. 67?

Папа Циркуль. Бедный Карлсон! Ему не позавидуешь. Алгоритм заставит его съесть пятьсот плюшек. Все до единой! И он наверняка умрет от обжорства.

Мурзик. Почему?

Папа Циркуль. Потому что из цикла на рис. 67 нельзя выйти раньше времени. Если в кошельке пятьсот монет, каждая команда цикла

- Возьми из кошелька денежку
- Купи себе плюшку
- Съешь плюшку

будет исполнена ровно пятьсот раз. И только после этого на вопрос: «В кошельке есть денежки?» — мы получим ответ «нет» и сможем уйти из цикла.

Мурзик. Как же быть?

Папа Циркуль. Чтобы спасти Карлсона, нужно организовать досрочный выход из цикла. Для этого в алгоритм нужно ввести дополнительную команду-вопрос: «Карлсон уже наелся?» (рис. 72)

Мурзик. Что это даст?

Папа Циркуль. Предположим, в кошельке пятьсот монет, а Карлсон может съесть всего двадцать плюшек. После того как цикл повторится двадцать раз, на вопрос: «Карлсон уже наелся?» — будет получен ответ «да». И мы благополучно выйдем из цикла.

Обрати внимание: цикл на рис. 72 имеет не один, а два выхода: *основной* и *досрочный*. Через первый мы выходим, когда в кошельке кончились деньги, через второй — когда Карлсон наелся.

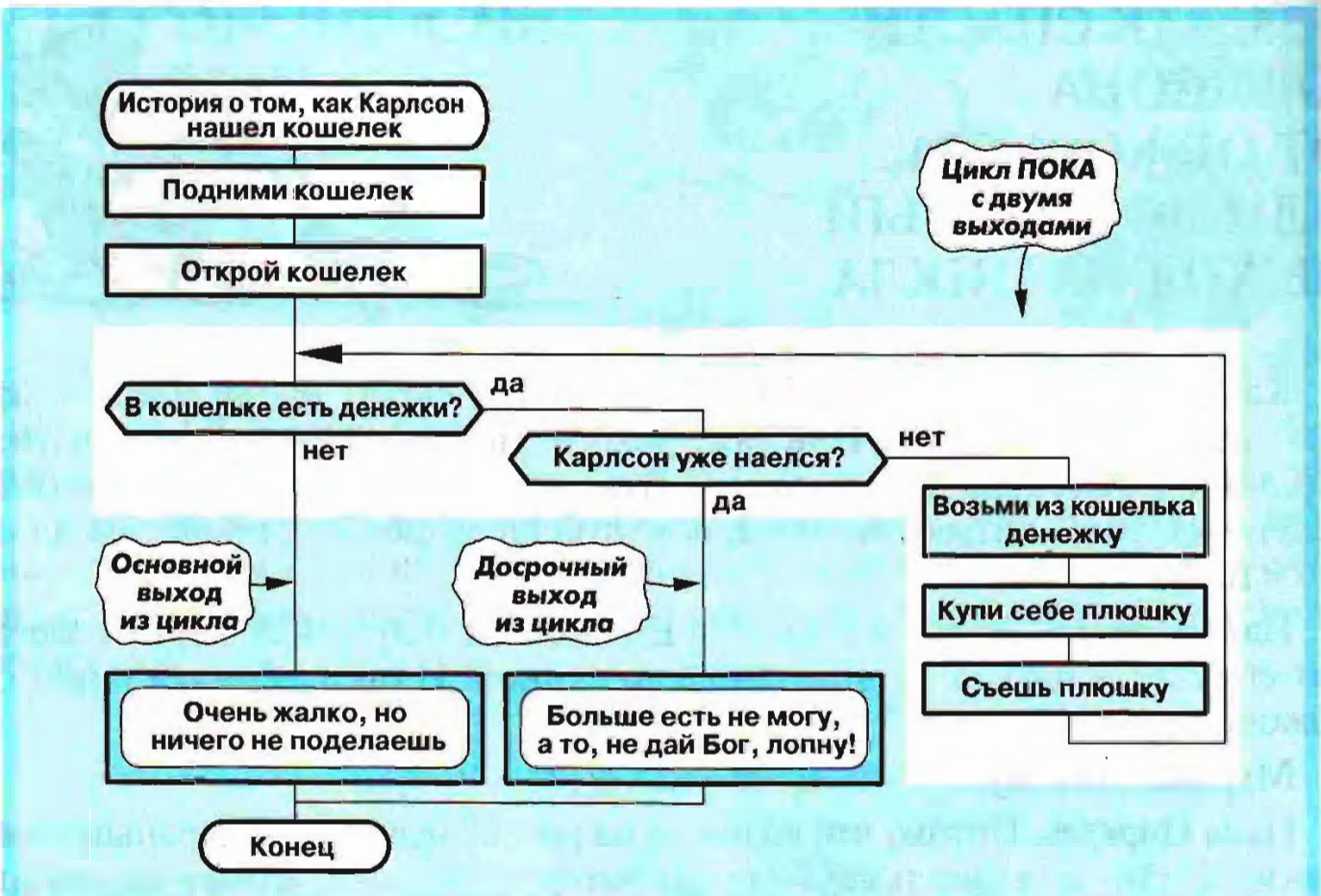


Рис. 72. Досрочный выход из цикла происходит потому, что Карлсон больше не хочет есть. Сравни с рис. 67

Зачем нужен досрочный выход из цикла → **Чтобы закончить цикл независимо от основного условия**

Как организовать досрочный выход из цикла → **Необходимо в цепь обратной связи ввести дополнительную команду-вопрос**

Задача Миши Проверялкина

Известно, что в кошельке a монет, а Карлсон может съесть b плюшек. Сколько раз будет выполнен цикл на рис. 72? Как завершится цикл: через основной выход или досрочный? Ответь на эти вопросы для следующих случаев:

- $a = 0, b = 5$;
- $a = 1, b = 1$;
- $a = 2, b = 1$;
- $a = 2, b = 2$;
- $a = 3, b = 4$.

§ 56. ДОСРОЧНЫЙ ВЫХОД ИЗ ЦИКЛА «ДО»

Папино слово — закон. А папа сказал: сегодня нужно покрасить забор. На рис. 71 показан случай, когда все идет по плану и работа благополучно доводится до конца.

Однако в жизни вечно случается то одно, то другое. Например, ни с того, ни с сего кончилась краска. Этот случай представлен на рис. 73. Алгоритм на рис. 73 имеет два выхода из цикла: основной (забор удалось покрасить) и досрочный (забор остался недокрашенным).

Однако что значит «кончилась краска»? Одно дело, если краски нет в ведре — тогда ее можно взять в сарае. И совсем другое, если в сарае краски тоже нет. Последний случай показан на рис. 74.

Жизнь полна неожиданностей. Кому охота красить дурацкий забор, если все нормальные люди играют в футбол? Этот полезный для футбола и вредный для забора случай отражен на рис. 75. Данный алгоритм интересен тем, что в нем три выхода из цикла: основной и два досрочных. В первом случае забор будет покрашен как надо, во втором дело не ладится из-за нехватки краски, в третьем — из-за любви к футболу.

Запомни

В цикле только один вход. А выходов может быть много. Один выход — основной, остальные — досрочные

Досрочный выход из цикла

Рис. 73. Досрочный выход из цикла, потому что кончилась краска

Рис. 74. Досрочный выход из цикла, потому что краска в ведре кончилась. И в сарае краски тоже нет

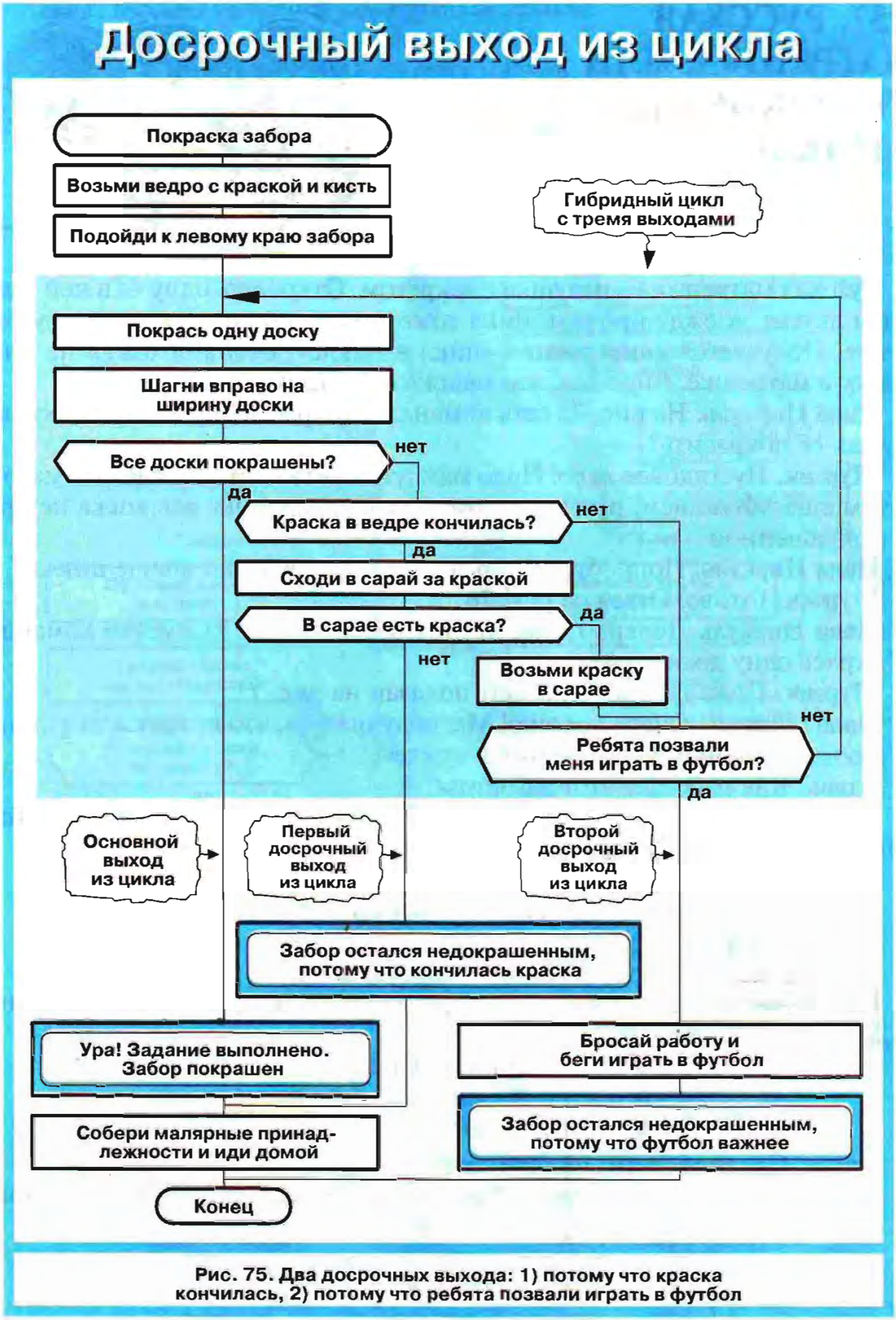


Рис. 75. Два досрочных выхода: 1) потому что краска кончилась, 2) потому что ребята позвали играть в футбол

§ 57. РУССКАЯ МАТРЕШКА, ИЛИ ЧТО ТАКОЕ ЦИКЛ В ЦИКЛЕ

Русская матрешка — игрушка с секретом. Откроешь одну — в ней прячется другая. Между прочим, цикл тоже можно спрятать внутри другого цикла. Получается конструкция «цикл в цикле», очень похожая на матрешку в матрешке. Впрочем, это присказка. Сказка будет впереди.

Папа Циркуль. На рис. 71 есть команда «Покрась одну доску». Сообрази, как её покрасить?

Мурзик. Пустяковое дело! Надо макнуть кисть в краску, сделать мазок, потом еще... В общем, придется помахать кистью, пока вся доска не станет окрашенной.

Папа Циркуль. Попробуй изобразить покраску доски в виде цикла.

Мурзик. Готово! Ответ на рис. 76.

Папа Циркуль. Теперь помести этот цикл на рис. 71 вместо команды «Покрась одну доску».

Мурзик. Пожалуйста. Результат показан на рис. 77.

Папа Циркуль. Очень хорошо! Мы получили то, что нужно: алгоритм, в котором есть конструкция «цикл в цикле».

Алина. Как работает этот алгоритм?

Папа Циркуль. Предположим, забор красит Том Сойер. Сначала Том выполняет две команды (рис. 77):

- Возьми ведро с краской и кисть
- Подойди к левому краю забора

Самое интересное начинается дальше.

Алина. Дальше нужно покрасить первую доску.

Папа Циркуль. Для этого Том исполняет команды, содержащиеся в цикле ДО (2):

- Обмакни кисть в краску
- Сделай мазок кистью по доске
- Покраска доски окончена?

Алина. Предположим, не окончена.

Папа Циркуль. В этом случае Том продолжает красить до тех пор, пока не будет выполнено условие окончания цикла ДО (2):

Покраска доски окончена?	=	да

Алина. Можно привести конкретный пример?

Папа Циркуль. Допустим, чтобы покрасить одну доску, нужно семь раз макнуть кисть в краску и сделать семь мазков. Значит, цикл ДО (2) будет выполнен ровно семь раз.

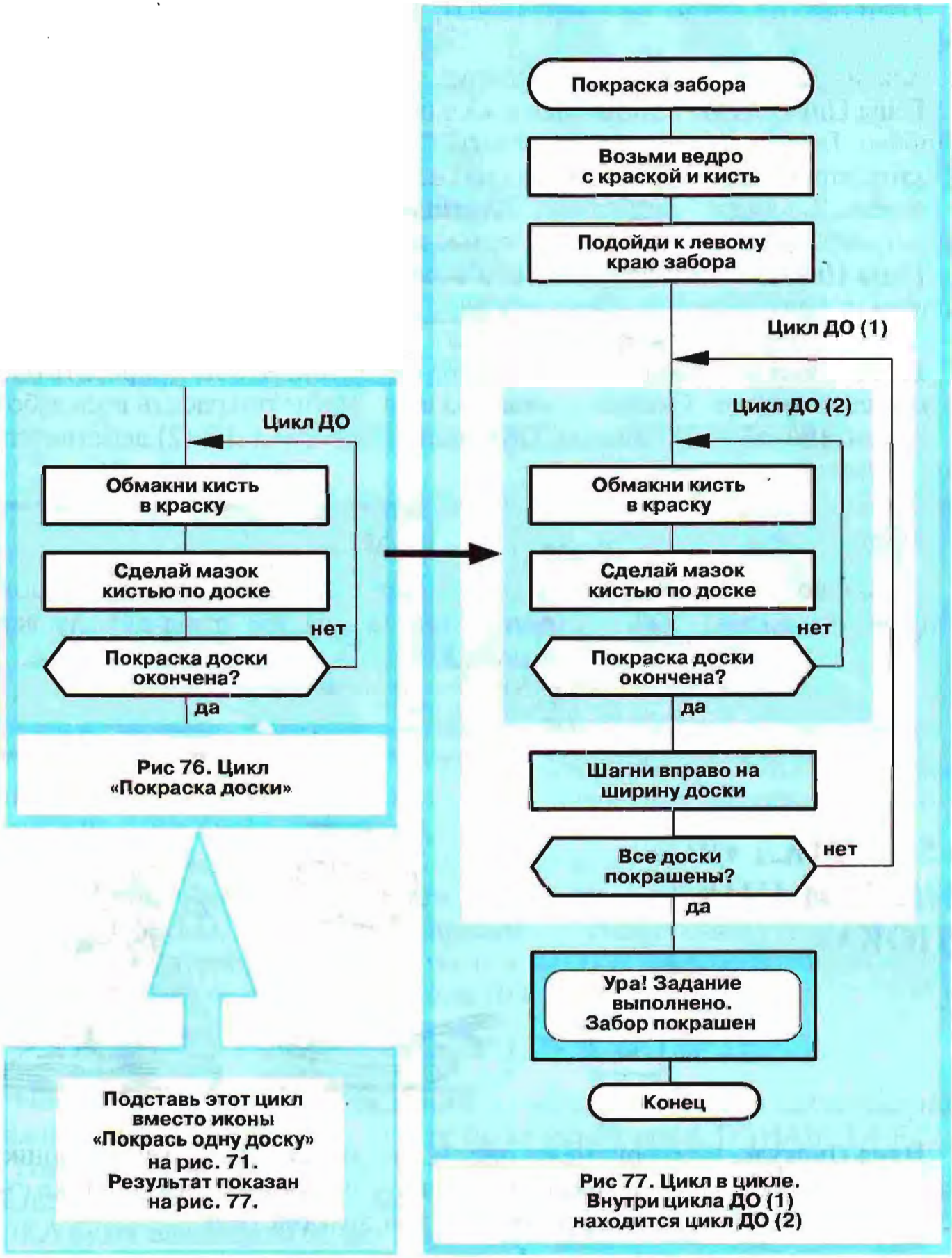


Рис 76. Цикл «Покраска доски»

Рис 77. Цикл в цикле. Внутри цикла ДО (1) находится цикл ДО (2)

Алина. А дальше?

Папа Циркуль. Том выполняет команды цикла ДО (1):

- Шагни вправо на ширину доски
- Все доски покрашены?

Если нет (не все доски покрашены), Том примется за следующую доску.

Алина. Давайте проследим маршрут.

Папа Циркуль. Из иконы «Все доски покрашены?» выходим через «нет» направо. По стрелке попадаем на вход цикла ДО (1). Затем Том начинает красить вторую доску. Он семь раз макнет кисть и сделает семь мазков.

Алина. Я, кажется, поняла. Покрасив вторую доску, Том шагнет вправо на ширину доски и возьмется за третью доску. И так далее.

Папа Циркуль. Вот именно. Если в заборе сто досок, цикл 1 будет выполнен сто раз, а цикл 2 — семьсот раз.

Алина. Почему семьсот?

Папа Циркуль. Считай сама. В заборе сто досок. Чтобы покрасить одну, нужно семь мазков. Сколько нужно мазков, чтобы покрасить весь забор?

Алина. $100 \times 7 = 700$ мазков. Получается, что цикл ДО (2) действительно повторяется семьсот раз.

Вопрос Саши Ехидного

Сколько раз выполняется команда на рис. 77, если в заборе a досок, причем на каждую нужно потратить b мазков. Найди ответ для случаев:

- $a = 1000$, $b = 5$
- $a = 300$, $b = 10$
- $a = 50$, $b = 12$

§ 58. ЦИКЛ «ДО» ВНУТРИ ЦИКЛА «ПОКА»

Папа Циркуль. В алгоритме на рис. 72 заменим икону «Съешь плюшку» на цикл ДО, состоящий из икон «Откуси кусок плюшки» и «Плюшка съедена?» (рис. 78). Посмотри внимательно на этот рисунок.

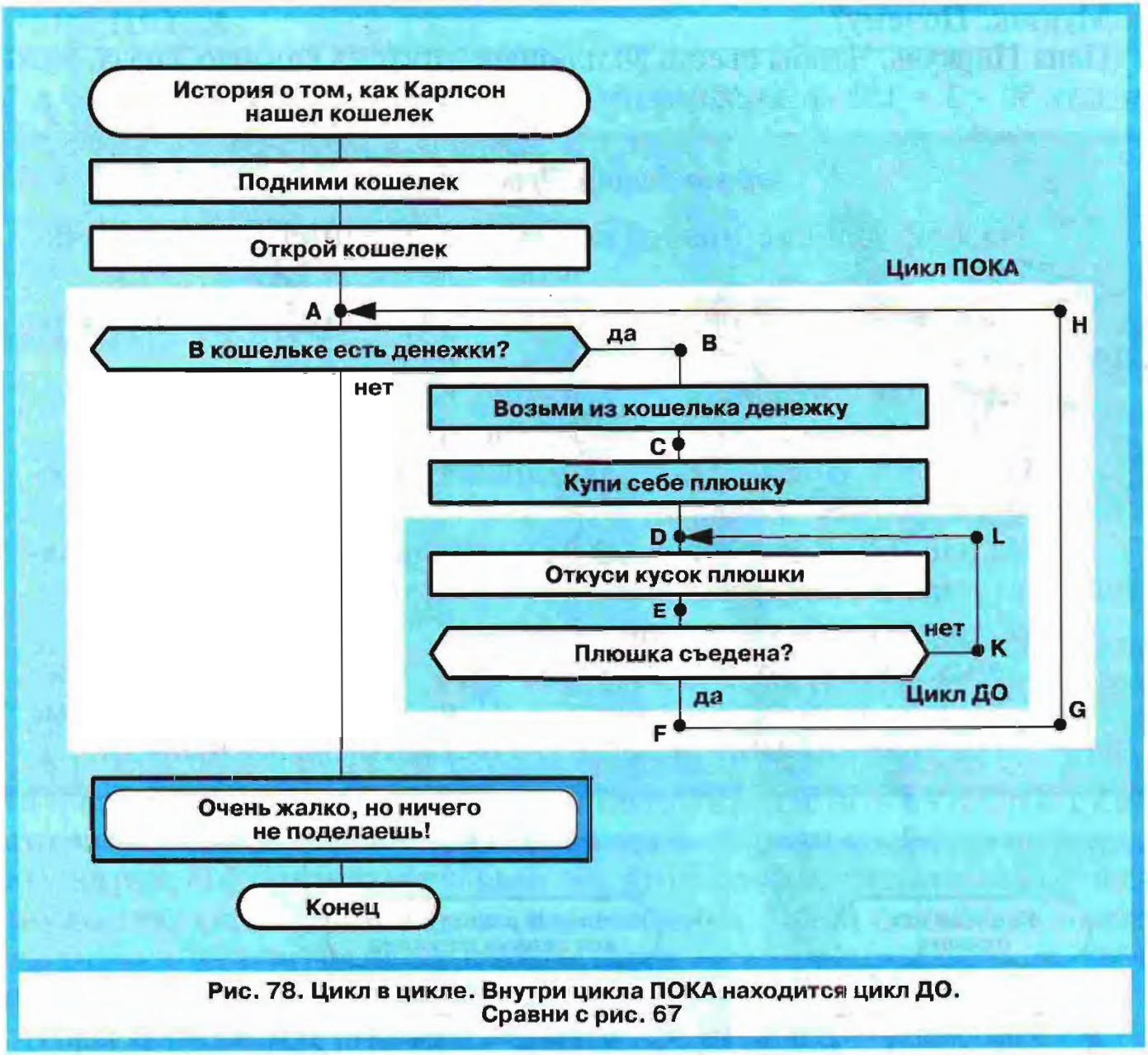


Рис. 78. Цикл в цикле. Внутри цикла ПОКА находится цикл ДО. Сравни с рис. 67

Мурзик. Ба! Перед нами снова цикл в цикле. Цикл ПОКА, построенный с помощью иконы «В кошельке есть денежки?», является внешним. В нем «прячется» внутренний цикл ДО.

Папа Циркуль. Что ты можешь о нем сказать?

Мурзик. При выполнении цикла ДО дракон-поезд кружит по внутренней петле DEKLD. За это время Карлсон откусывает один кусок плюшки, потом другой и так далее. Когда он покончит с первой плюшкой, будет выполнено условие окончания цикла ДО.

Плюшка съедена?	=	да

После этого маршрут дракон-поезда меняется. Если в кошельке по-прежнему есть деньги, поезд поедет по внешней петле FGHABCDEF.

Папа Циркуль. Предположим, в кошельке пятьдесят монет, а Карлсон съедает плюшку за три приема. Это значит, что каждая команда цикла ПОКА будет выполнена 50 раз, а каждая команда цикла ДО — 150 раз.

Мурзик. Почему?

Папа Циркуль. Чтобы съесть 50 плюшек, откусив каждую 3 раза, нужно сделать $50 \times 3 = 150$ «откусываний».

Задачи Миши Проверялкина

1. Назови условие продолжения и окончания цикла на рис. 71—75.
2. Укажи команду «Вопрос», тело цикла и цепь обратной связи на рис. 71—75.
3. Чем отличаются алгоритмы на рис. 67 и 72?
4. Чем отличаются алгоритмы на рис. 67 и 78?
5. На рис. 77 замени цикл ДО (2) на икону-вставку. Нарисуй основной алгоритм и процедуру.
6. На рис. 78 замени цикл ДО на икону-вставку. Нарисуй основной алгоритм и процедуру.
7. Исправь ошибку в алгоритме на рис. 79.
8. Придя домой, школьник решил сначала сделать домашнее задание, а потом идти гулять до ужина. Исправь ошибку в алгоритме на рис. 80, чтобы желание школьника оказалось выполненным.

Рис. 79. Исправь ошибку

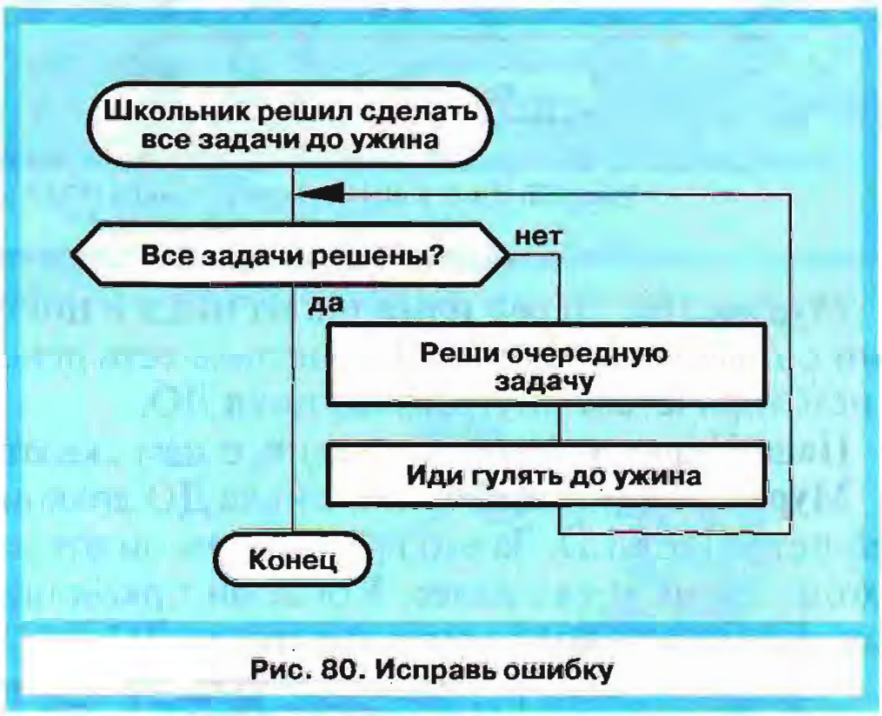


Рис. 80. Исправь ошибку

Глава пятая

КАК РАЗДЕЛИТЬ АЛГОРИТМ НА СМЫСЛОВЫЕ ЧАСТИ?

§ 59. ЧТО ТАКОЕ ВЕТКА?

Когда принцесса Анна развелась со своим мужем маркизом Ришелье, возник спор о разделе имущества. Судья потребовал указать: какие покупки принцесса сделала до, а какие после замужества.

А теперь забудем об этой семейной драме и сравним между собой рис. 81*а* и *б*. Легко видеть, что рис. 81*а* не позволит ответить на вопрос судьи. Зато рис. 81*б*, наоборот, содержит нужную информацию. Более того, алгоритм на рис. 81*б* нарочно нарисован так, что покупки, сделанные до и после замужества, четко разделены на два столбика. Такой прием называется разделением алгоритма на смысловые части. А сами части называются *ветки*.

Папа Циркуль. На рис. 45 представлена уже знакомая нам схема «Сборы на рыбалку». Попробуй разбить ее на смысловые части.

Мурзик. Это очень легко. «Сборы на рыбалку» — это алгоритмический рассказ, в котором можно выделить три крупных куска, три самостоятельные темы:

- Подъем и завтрак
- Укладка вещей
- Поездка

Каждую тему можно нарисовать в виде ветки. Результат я изобразил на рис. 82.

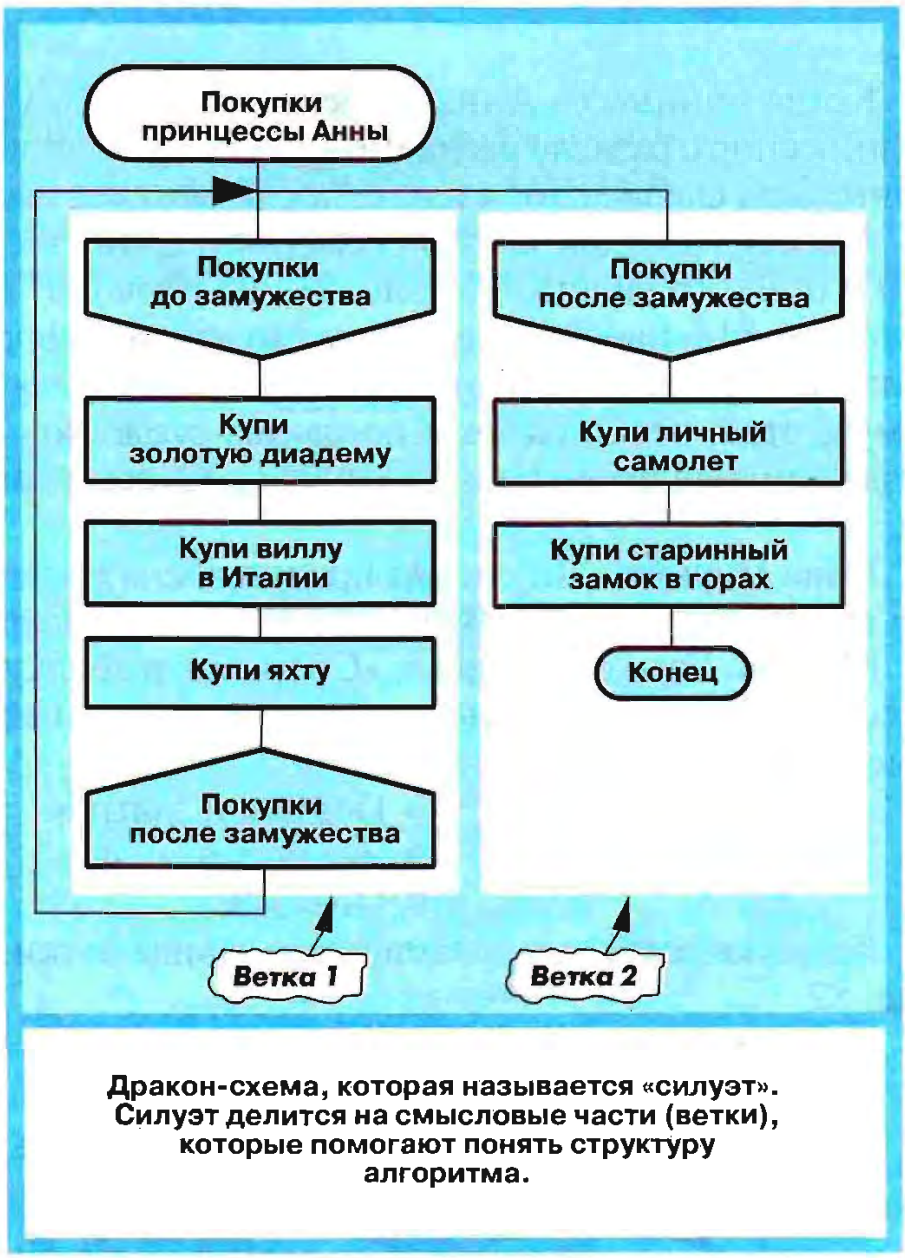
Папа Циркуль. Молодец!

Алина. Как называются новые иконы?
Папа Циркуль. Ответ ясен из таблицы.

Икона	Название иконы	Пояснение
	Имя ветки	Обозначает начало ветки
	Адрес	Обозначает конец любой ветки, кроме последней

Дракон-схема, которая называется «примитив». В примитиве ветки отсутствуют.

а



Дракон-схема, которая называется «силуэт». Силуэт делится на смысловые части (ветки), которые помогают понять структуру алгоритма.

б

Рис. 81. Чем различаются примитив и силуэт?

Алина. Где начало и конец у первой ветки на рис. 82?

Папа Циркуль. Начало — икона «Подъем и завтрак». Конец — икона «Укладка вещей». Между началом и концом размещается тело ветки. Оно содержит иконы «Встань пораньше» и «Позавтракай».

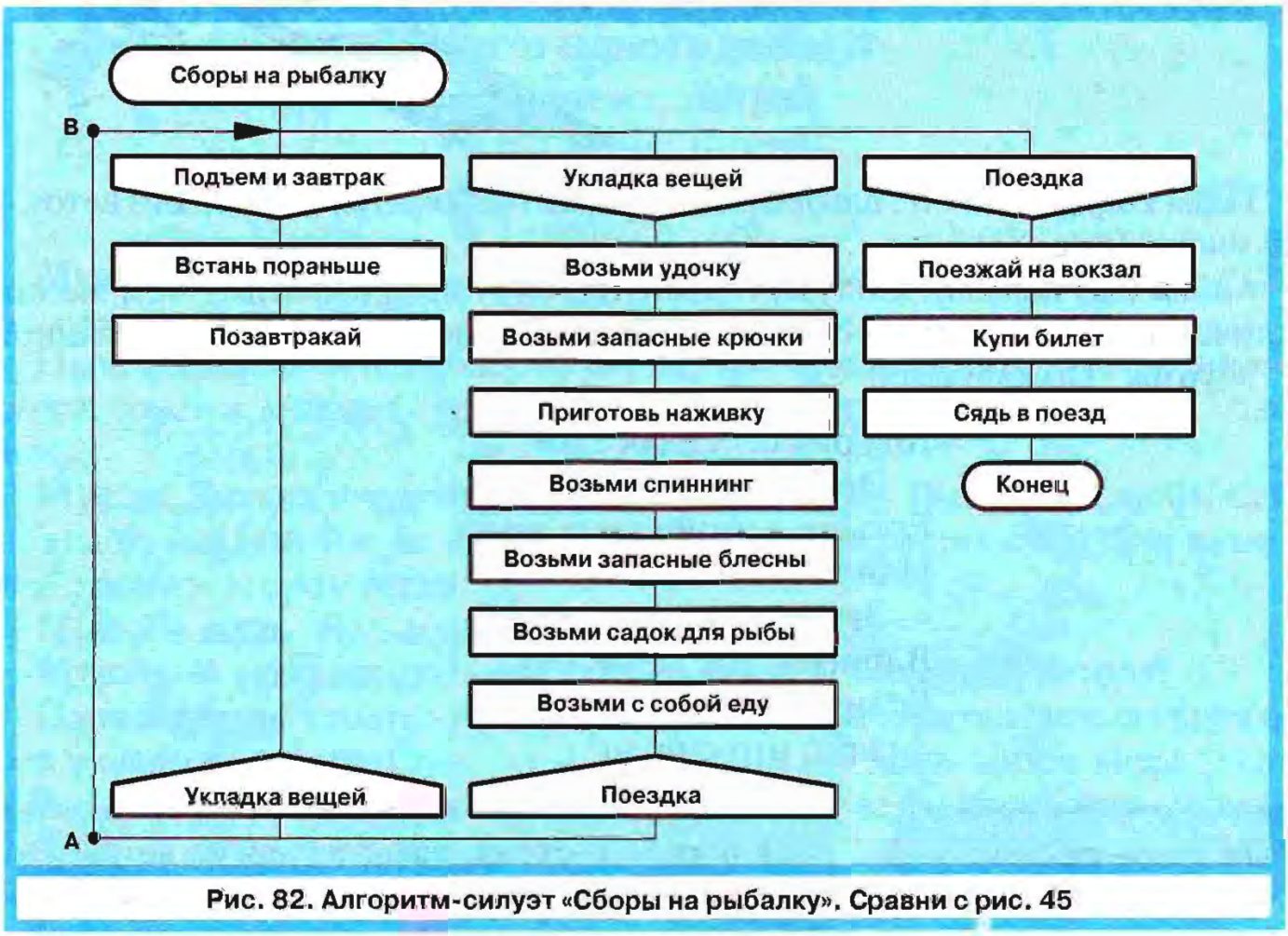
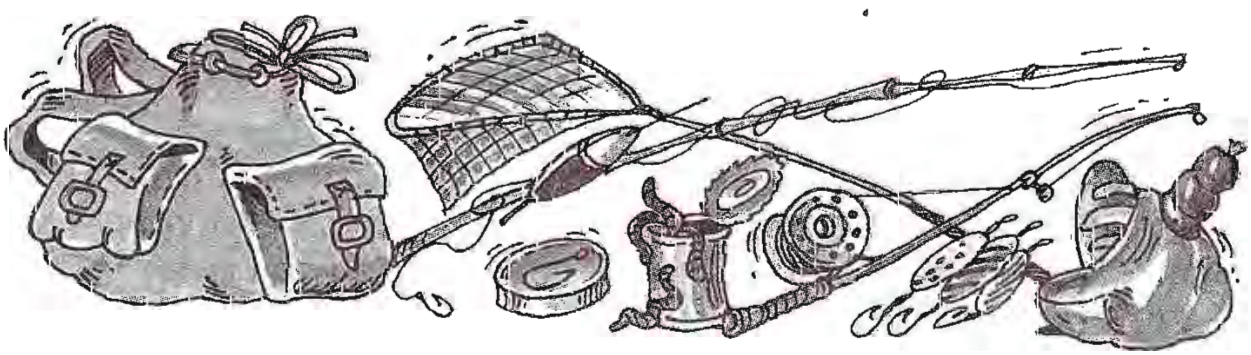


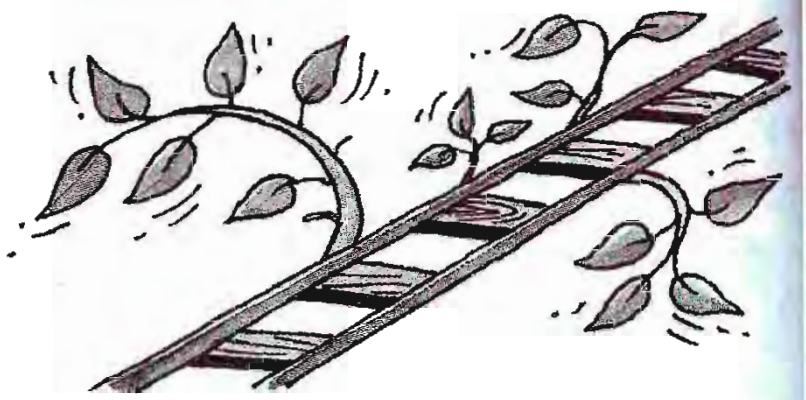
Рис. 82. Алгоритм-силуэт «Сборы на рыбалку». Сравни с рис. 45

Что такое ветка

Это смысловая часть алгоритма, которая содержит:
- **икону «Имя ветки» (в ней пишут название смысловой части)**
- **тело ветки**
- **одну или несколько икон «Адрес» (в любой ветке, кроме последней)**
- **икону «Конец» (в последней ветке)**



§ 60. СИЛУЭТ И ПРИМИТИВ



Папа Циркуль. Итак, алгоритм с ветками называется *силуэт*, без веток — *примитив* (рис. 81).

Алина (*заучивает*). Силуэт, примитив, силуэт, примитив... Чем же они отличаются? Непонятно! Мурзик, придумай мне понимательный стишок.

Мурзик. Пожалуйста!

Говорит сосед соседке:
— Эй, соседка, дай ответ —
Есть ли в этой схеме ветки
Или веток в схеме нет?
— Знают даже малолетки:
В примитиве веток нет.
Если ты увидишь ветки,
Значит, это силуэт!

Что такое силуэт → **Это дракон-схема, разделенная на ветки**

Что такое примитив → **Это дракон-схема, не имеющая веток**

§ 61. КАКИМ ОБРАЗОМ ДРАКОН-ПОЕЗД ЕДЕТ ИЗ НАЧАЛА В КОНЕЦ СИЛУЭТА?

Мурзик. Как работает алгоритм на рис. 82?

Папа Циркуль. Работа любого алгоритма, будь то примитив или силуэт, состоит в том, что дракон-поезд должен пройти путь от иконы «Заголовок» до иконы «Конец». Значит, любой маршрут силуэта ведет из начала в конец. Это общее правило.

Для примитива и силуэта
Действует всюду правило это:
«Маршруты идут из начала в конец».
Тот, кто запомнил, тот — молодец!

Выехав из иконы «Заголовок», дракон-поезд мчится вниз по крайней левой ветке. Он движется через станции (рис. 82):

- Подъем и завтрак
- Встань пораньше
- Позавтракай
- Укладка вещей

Мурзик. Икона «Адрес» — последняя станция первой ветки. Куда ехать дальше?

Папа Циркуль. Ответ содержится внутри самой иконы. Эта икона потому и зовется «Адрес», что сообщает адрес (название) следующей станции.

Мурзик. Значит, следующая станция называется «Укладка вещей». Сейчас мы ее найдем. Где же она? (*Водит пальцем по рисунку 82.*) Ага, нашел! Она в начале второй ветки.

Папа Циркуль. Правильно.

Мурзик. А как дракон-поезд туда доберется? По какой линии?

Папа Циркуль. Смотри сюда. Выехав из иконы «Адрес», поезд сворачивает налево и попадает в точку А (рис. 82). Потом движется вверх к точке В. Затем едет направо по стрелке и въезжает в верхнюю икону «Укладка вещей».

Мурзик. Дальше все понятно. Поезд скользит вниз по второй ветке. Добравшись до иконы «Адрес», узнает адрес следующей станции («Поездка»). Затем огибает схему по линии АВ, попадает в начало третьей ветки и спускается по ней до конца. Все, приехали! Маршрут окончен.

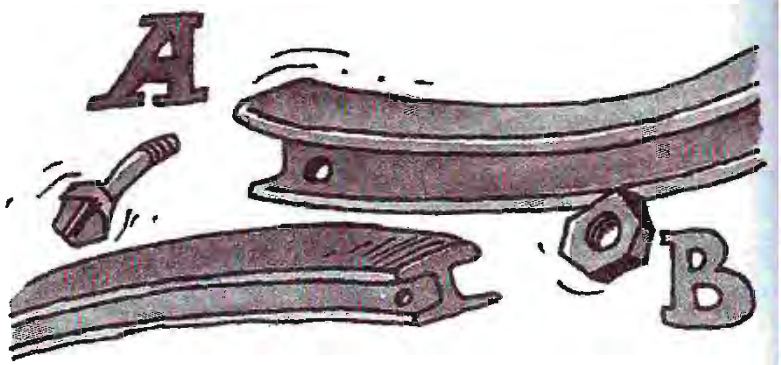
Правило

Из иконы «Заголовок» дракон-поезд едет вниз по крайней левой ветке

Правило

Из иконы «Адрес»  **дракон-поезд едет в икону «Имя ветки»** 

§ 62. В ЧЕМ СЕКРЕТ ИКОНЫ «АДРЕС»?



Папа Циркуль. Сейчас мы поступим, как чеховский злоумышленник — будем разбирать рельсы. Но сначала ответь, Мурзик, сколько линий проходит через точки A и B на рис. 82?

Мурзик. Три: одна вертикальная (AB) и две горизонтальных.

Папа Циркуль. Сотри их.

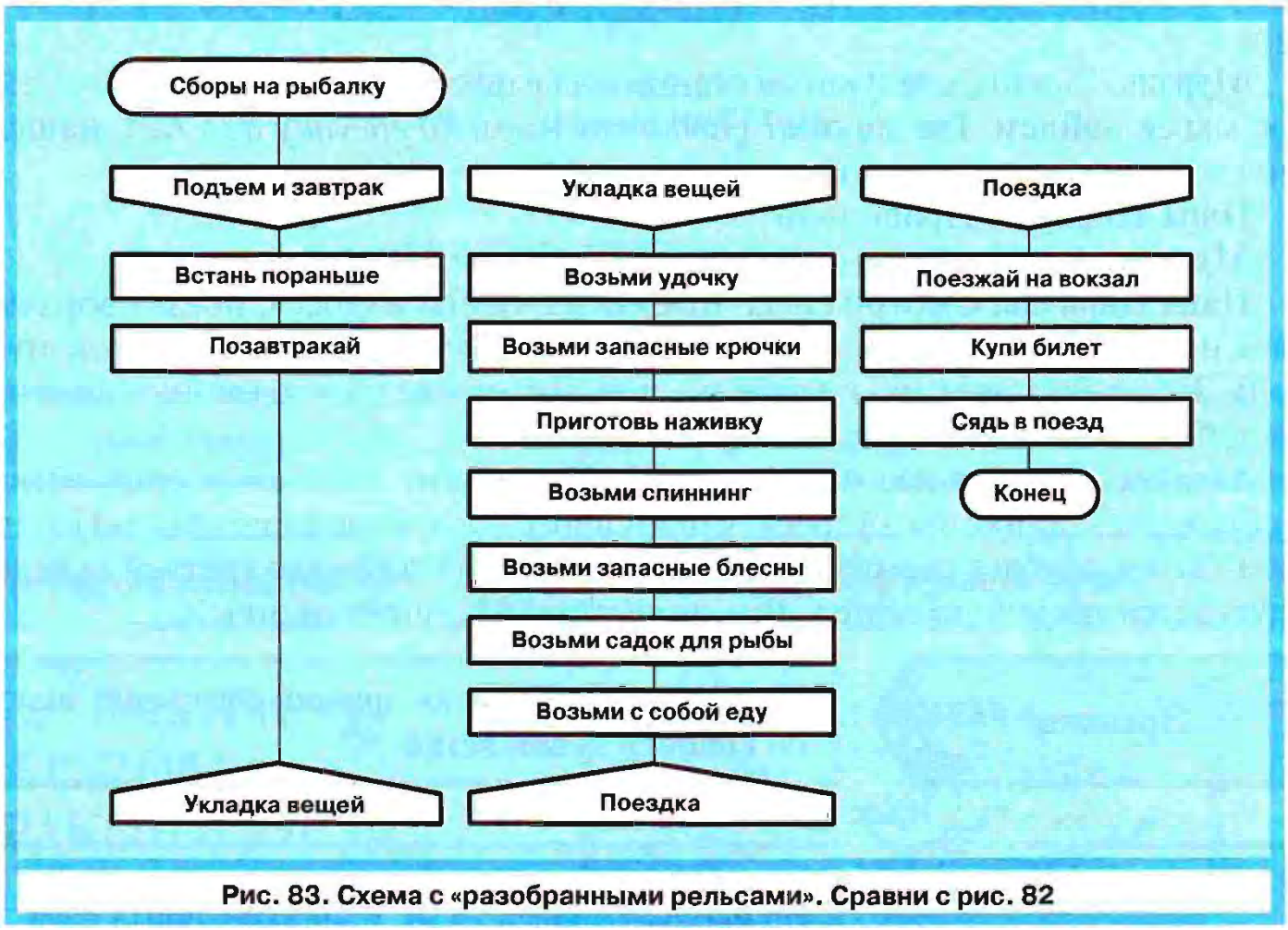


Рис. 83. Схема с «разобранными рельсами». Сравни с рис. 82

Мурзик. Готово! (*Результат показан на рис. 83.*) Рельсы разобраны, и дракон-дорога рассыпалась на три ветки.

Папа Циркуль. Тебе нравится алгоритм на рис. 83?

Мурзик. Нет, не нравится. На дороге сплошные тупики и обрывы. Проехав по первой ветке, дракон-поезд попадет в тупик и остановится у иконы «Адрес». Потому что неизвестно, куда ехать дальше.

Папа Циркуль. Неизвестно? Ты уверен?

Мурзик. Ой, я ошибся! Я забыл, что в иконе «Адрес» написан адрес следующей станции.

Папа Циркуль. Конечно! Адрес известен, и дракон-поезд может его прочитать. Но как же он доберется до цели — ведь рельсы-то разобраны!

Мурзик. Изучая § 44, мы узнали, что дракон-поезд умеет делать акробатические прыжки. Именно так он и поступит. Зная адрес, он прыгнет туда, куда нужно, — в начало второй ветки.

Папа Циркуль. А дальше?

Мурзик. Доехав до конца второй ветки, он совершит второй прыжок и попадет в третью ветку.

Папа Циркуль. Что отсюда следует?

Мурзик. Получается, что рельсы, которые мы убрали, нарисованы просто для красоты. На самом деле они не нужны. Потому что маршрут дракон-поезда определяют не они, а указания, написанные в иконе «Адрес».

Папа Циркуль. Молодец! Ты попал в точку! Чтобы лучше все это понять, давай вспомним: что означает икона «Адрес»? Зачем она нужна?

Мурзик. Она означает конец ветки.

Папа Циркуль. Верно, но главный секрет не в этом. Икона «Адрес А» — это команда «Прыгни в начало ветки А». Или, говоря по-научному: «Передай управление в ветку А».

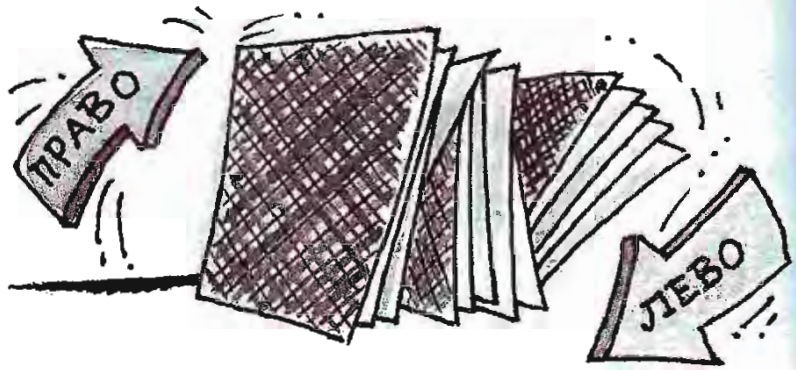
Алина. Кто выполняет эту команду?

Папа Циркуль. Как всегда, робот-исполнитель. Команда говорит роботу: «Брось данную ветку и начни выполнять ветку А». (Выполнять ветку – значит исполнять записанные в ней команды).

Алина. Нельзя ли привести пример?

Папа Циркуль. Посмотри на рис. 83. Предположим, робот прочитал команду «Поездка», написанную в иконе «Адрес». Подчиняясь приказу, он покидает прежнюю ветку, находит ветку с названием «Поездка» и начинает ее выполнять.

§ 63. ВЕТКИ СЛЕДУЕТ РИСОВАТЬ НЕ КАК ПОПАЛО, А УПОРЯДОЧЕННО



Давайте мысленно перетасуем ветки на рис. 83, как колоду карт, и расположим их в произвольном порядке. Легко сообразить, что подобная операция не влияет на работу алгоритма. Потому что очередность работы веток зависит от команд «Адрес» и не зависит от расположения веток на бумаге.

Мурзик. Словом, сколько ветки ни тасуй, получим тот же самый алгоритм.

Папа Циркуль. Здесь есть одна тонкость. Перестановка веток не влияет на правильность алгоритма. Однако алгоритм должен быть не только правильным, но и понятным, эргономичным. Хаотичное расположение веток затрудняет понимание, что недопустимо.

Поэтому нужно обязательно упорядочить ветки. Для этого служит правило: «Чем правее, тем позже». Оно означает: чем правее находится ветка, тем позже она работает. Это правило мы будем соблюдать (рис. 82).

Запомни

Ветки должны быть упорядочены слева направо согласно правилу: «Более правая ветка работает позже, чем любая ветка, расположенная левее нее»

§ 64. ЧТО ТАКОЕ ВЕТОЧНЫЙ ЦИКЛ?

Мурзик. На рис. 71 изображен циклический алгоритм «Покраска забора». Можно ли нарисовать его в виде силуэта?

Папа Циркуль. Нет ничего проще. Крибле-крабле-бумс! — и желаемый результат уже появился на рис. 84. Кстати, на этом рисунке есть одна заковыка. Попробуй, Мурзик, ее найти.

Мурзик. Где же она? Ага, нашел! Во второй ветке!

Папа Циркуль. Верно. Выкладывай, что ты раскопал.

Мурзик. В этой ветке слово «Покраска» встречается дважды: вверху и внизу.

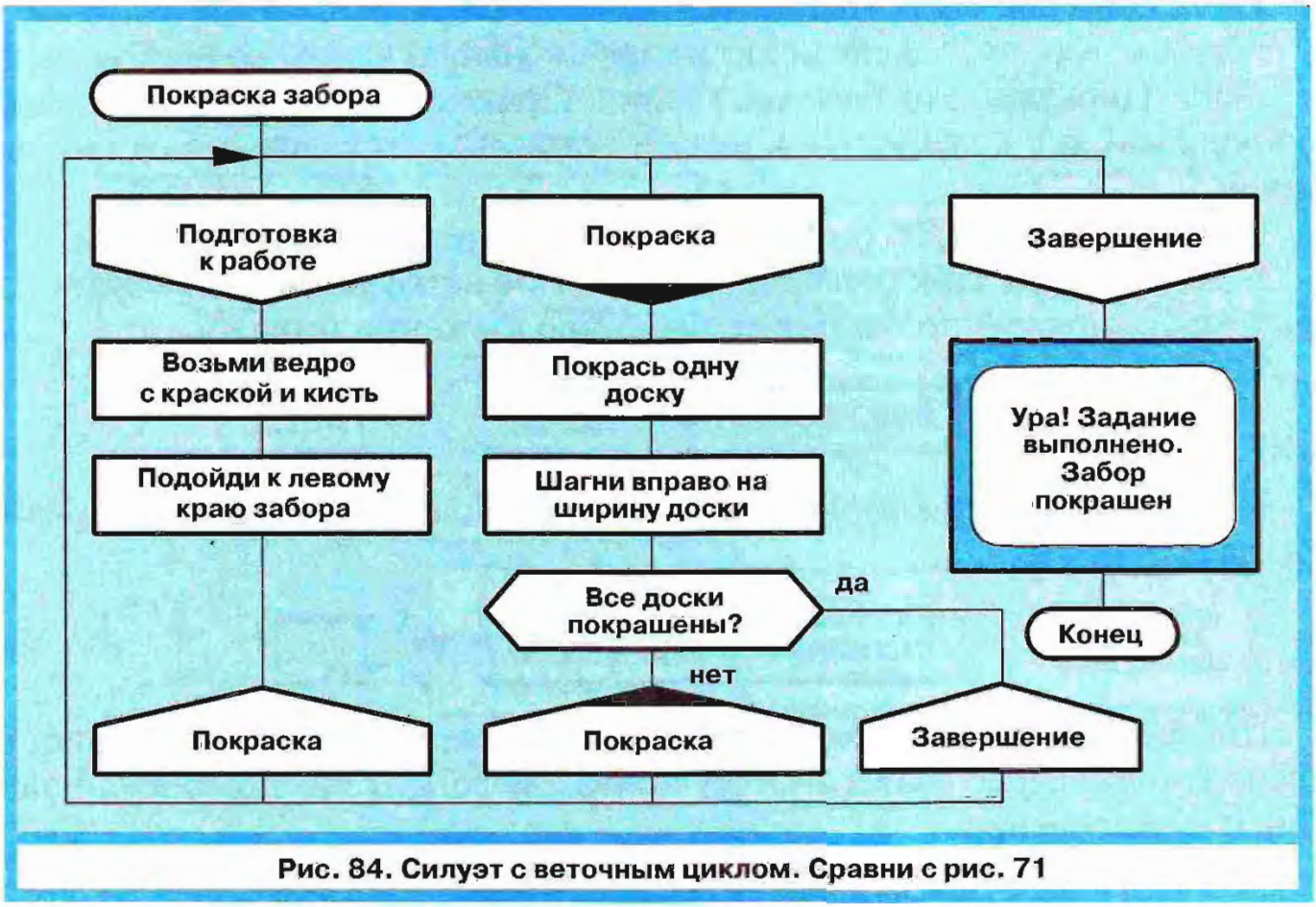


Рис. 84. Силуэт с веточным циклом. Сравни с рис. 71

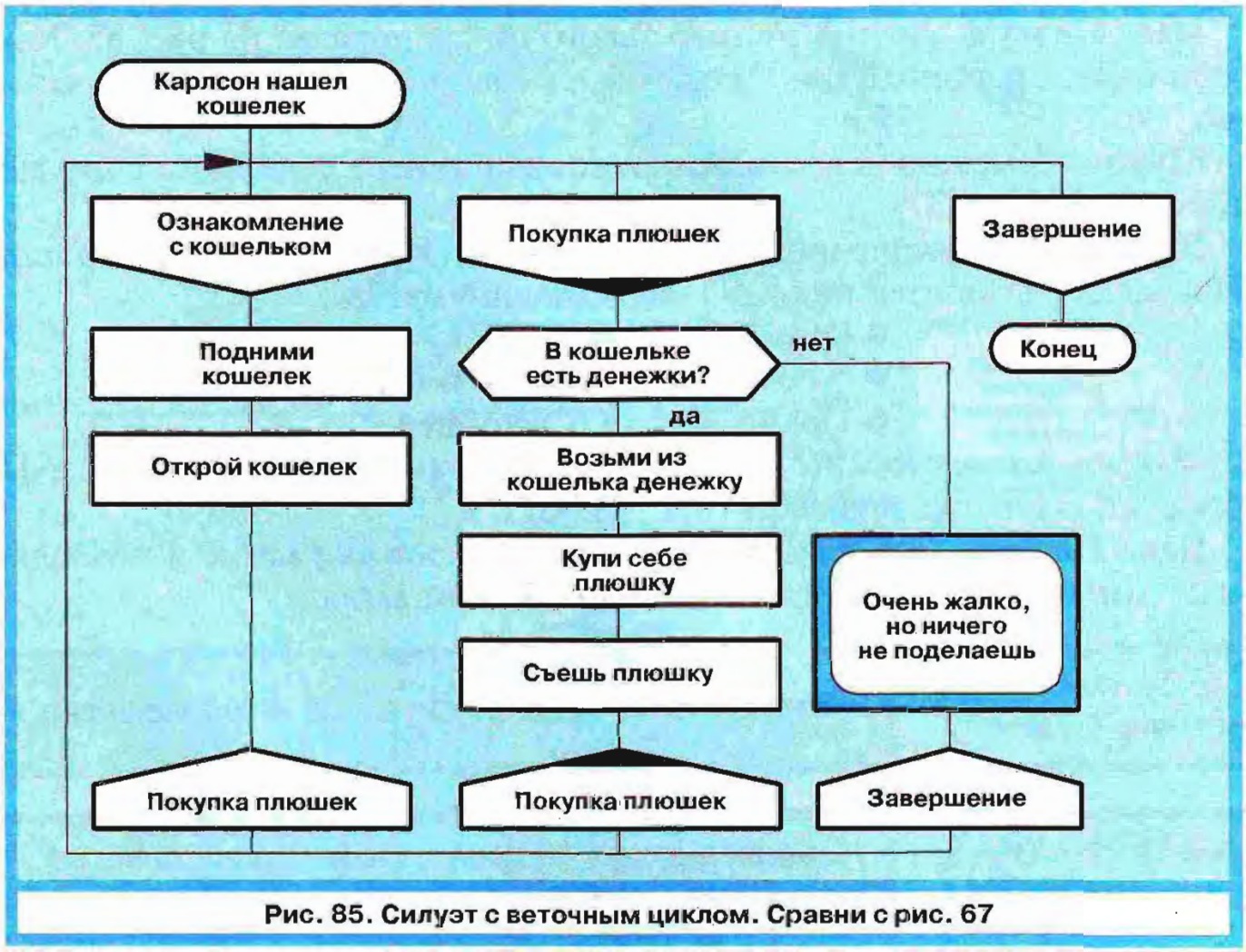


Рис. 85. Силуэт с веточным циклом. Сравни с рис. 67

Папа Циркуль. Ну и что?

Мурзик. Как что? Да ведь это же цикл! Только какой-то необычный.

Папа Циркуль. Это *веточный* цикл. Дракон-поезд, доехав до адреса «Покраска», тут же вернется к началу ветки и будет ездить по ней вновь и вновь.

Мурзик. И как долго он будет утюжить одну и ту же ветку?

Папа Циркуль. Циклическое движение по ветке «Покраска» будет продолжаться, пока выполняется условие продолжения цикла:

Все доски покрашены?	=	нет

Когда Том Сойер кончит красить забор, появится условие окончания цикла:

Все доски покрашены?	=	да

После этого дракон-поезд, проходя через икону «Вопрос», повернет направо и через адрес «Завершение» попадет в ветку «Завершение». А там и конец — делу венец.

Алина. Мне очень понравился веточный цикл. Он красивый и элегантный.

Папа Циркуль. Другой пример такого цикла показан на рис. 85. Как и любой цикл, веточный цикл может иметь основной и досрочный выходы (рис. 86).

Мурзик. А можно использовать веточный цикл в сочетании с циклами ПОКА и ДО?

Папа Циркуль. Конечно. Например, на рис. 87 внутри веточного цикла «Покраска» находится цикл ДО, содержащий иконы:

- Обмакни кисть в краску
- Сделай мазок кистью по доске
- Покраска доски закончена?

Мурзик. А зачем нужны маленькие черные треугольники в иконах «Покраска» и «Покупка плюшек» (рис. 84—87)?

Папа Циркуль. Это флажки. Они привлекают внимание и позволяют легко заметить веточный цикл даже при беглом взгляде.

Что такое веточный цикл	**Это повторное исполнение одной и той же ветки**
Как построить веточный цикл	**Напиши в иконе «Адрес» название данной ветки (или более левой ветки)**

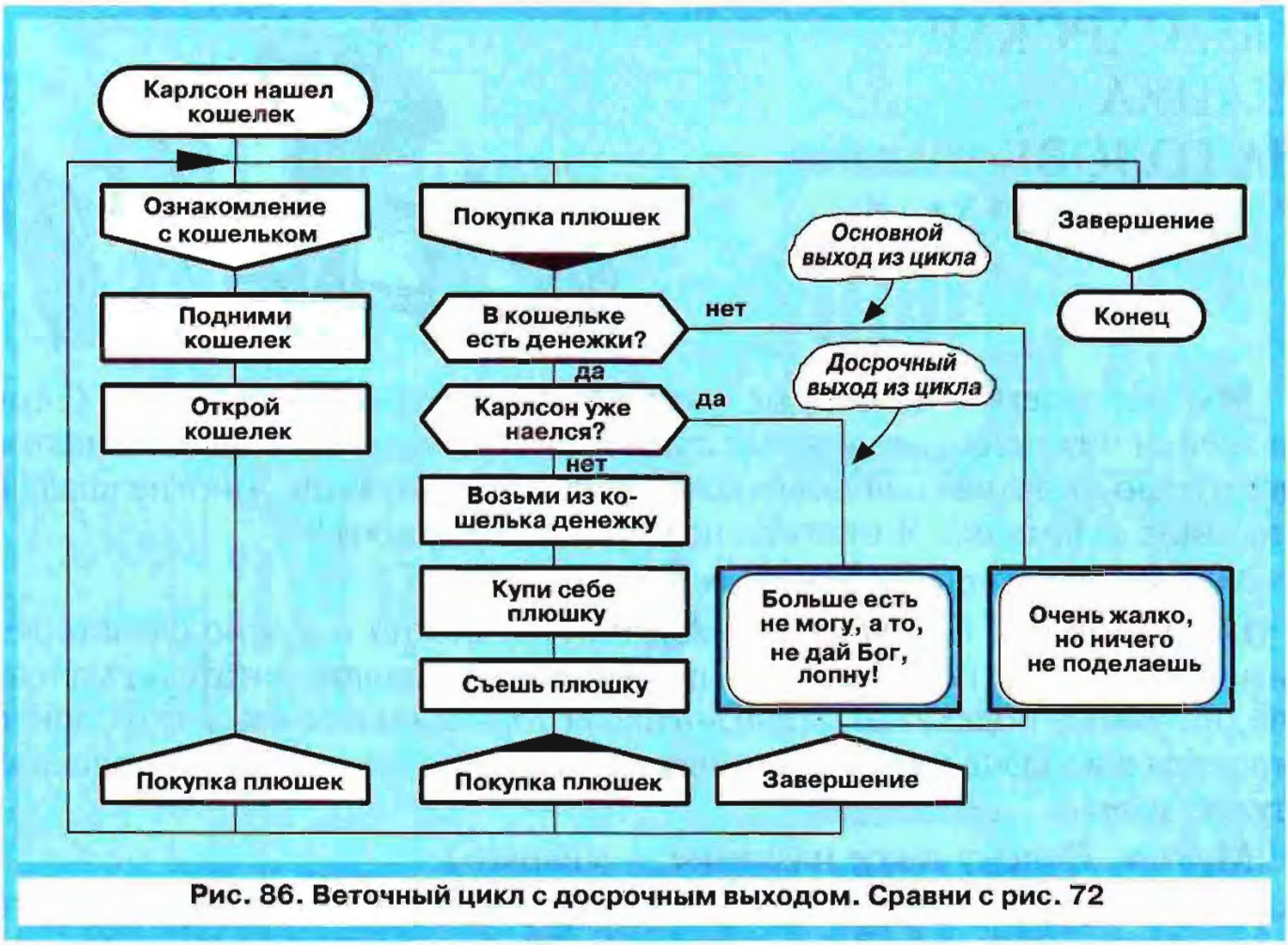


Рис. 86. Веточный цикл с досрочным выходом. Сравни с рис. 72

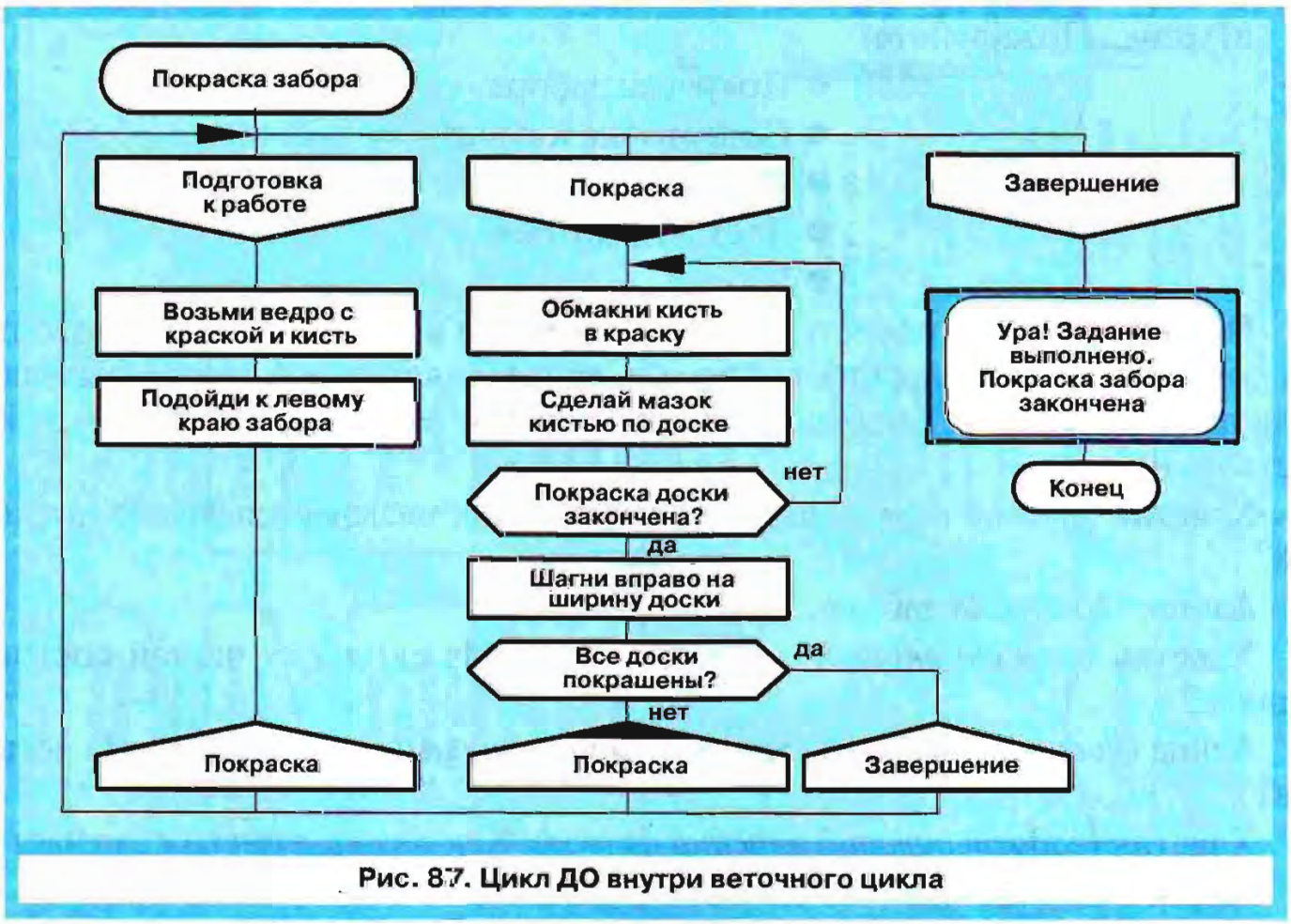


Рис. 87. Цикл ДО внутри веточного цикла

§ 65. ЦАРСКАЯ ШАПКА НА ГОЛОВЕ У АЛГОРИТМА

Мы уже знаем — некоторые алгоритмы трудны для понимания. С точки зрения читателя, запутанный алгоритм, да еще незнакомый — настоящее пугало, бедствие для новичков. Увидев такое чудище, многие впадают в панику: «Мамочка! Я никогда не смогу этого понять!»

Мурзик. Неужели все так плохо?

Папа Циркуль. Конечно, нет! Алгоритмы можно и нужно сделать легкими для восприятия. Для этого нужно вовремя давать читателю маленькие, но умные подсказки, «проглотив» которые, он мог бы легко сориентироваться в задаче и быстрее понять материал. Одной из таких подсказок служит *шапка*.

Мурзик. Почему такое название — «шапка»?

Папа Циркуль. Потому что она находится вверху, «на голове» у алгоритма. Найди верхние пять икон на рис. 88.

Мурзик. Пожалуйста:

- Покраска забора
- Подготовка к работе
- Покраска
- Доставка краски
- Уборка

Папа Циркуль. Это и есть шапка. Изюминка в том, что шапка позволяет ответить на три царских вопроса. Сейчас Хлястик и Алина разыграют для нас сценку. Хлястик будет задавать царские вопросы, а Алина — отвечать на них.

Хлястик (*задает первый царский вопрос*). Как называется задача на рис. 88?

Алина. Покраска забора.

Хлястик (*задает второй царский вопрос*). Из скольких частей состоит задача?

Алина (*водит пальцем по схеме и считает иконы «Имя ветки»*). Из четырех.

Хлястик (*задает третий царский вопрос*). Как называется каждая часть?

Алина (*читает названия веток*).

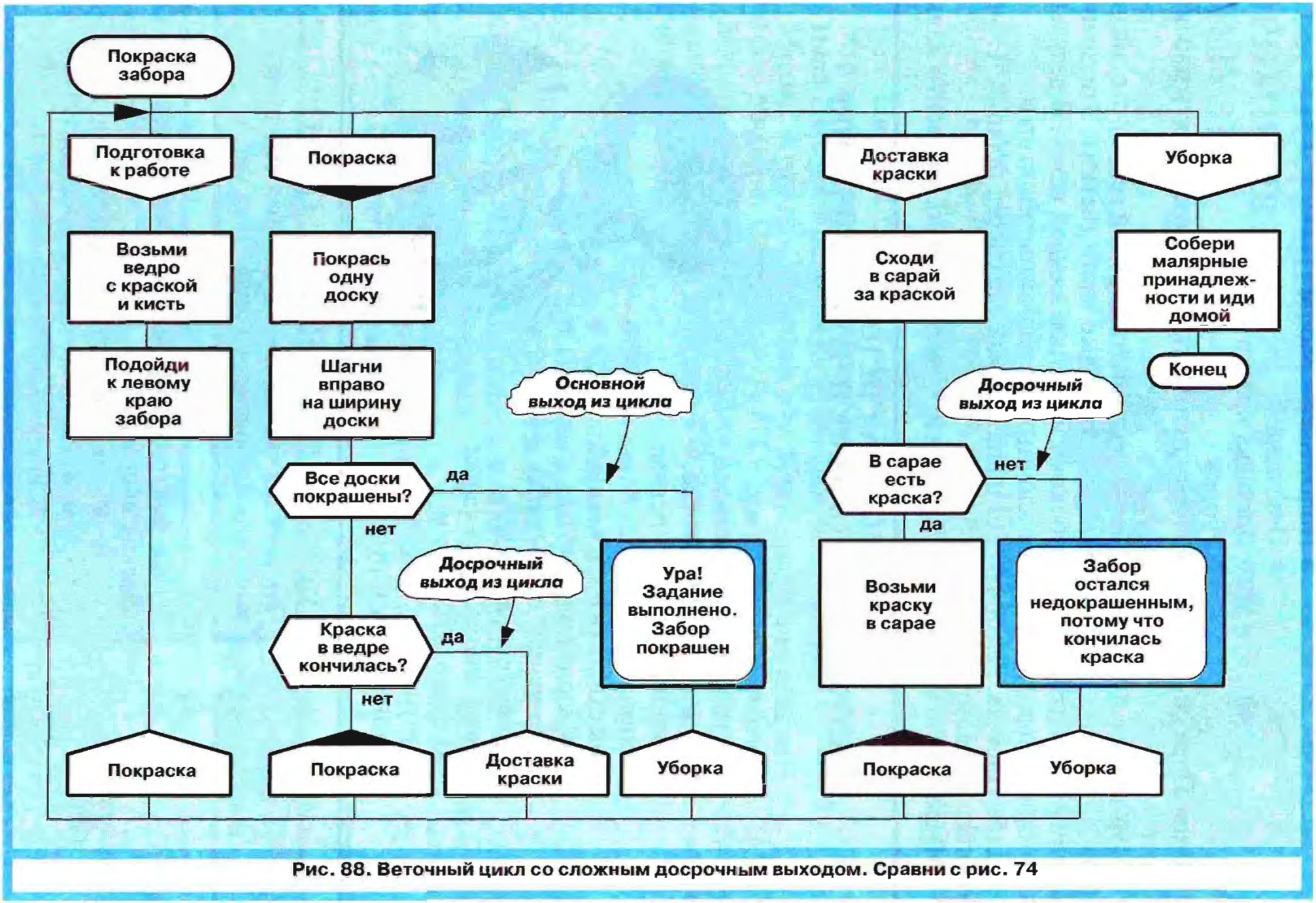


Рис. 88. Веточный цикл со сложным досрочным выходом. Сравни с рис. 74

- Подготовка к работе
- Покраска
- Доставка краски
- Уборка

Папа Циркуль. Раздобыть царские ответы — значит понять царский секрет.

Мурзик. Все ясно! Шапка позволяет узнать царский секрет, т. е. назначение алгоритма и его деление на смысловые части (ветки). Эти знания помогают перейти к более глубокому изучению каждой ветки. В результате алгоритм перестает быть загадочным и становится понятным.

Папа Циркуль. Алгоритм без шапки — как гвоздь без шляпки. С виду хорошо, а пользоваться неудобно.

Мурзик. Или как чемодан без ручки — нести тяжело, а бросить жалко.

НЕВЕРОЯТНАЯ ИСТОРИЯ О БЕЗУМНОМ АЛГОРИТМЕ И ЧУДЕСНОЙ ШАПКЕ

Что за шум и кутерьма?
Алгоритм сошел с ума!
Он то дрался-бесновался,
То мычал в глухой тоске,
То кричал на тарабарском,
Непонятном языке.
Как схватить его в охапку?
Как злодея одолеть?
Надо шапку, чудо-шапку
На него скорей надеть!
В этой шапке, чудо-шапке
Он становится иным:
И учтивым, и приятным,
И покорным, и ручным.

Что такое шапка

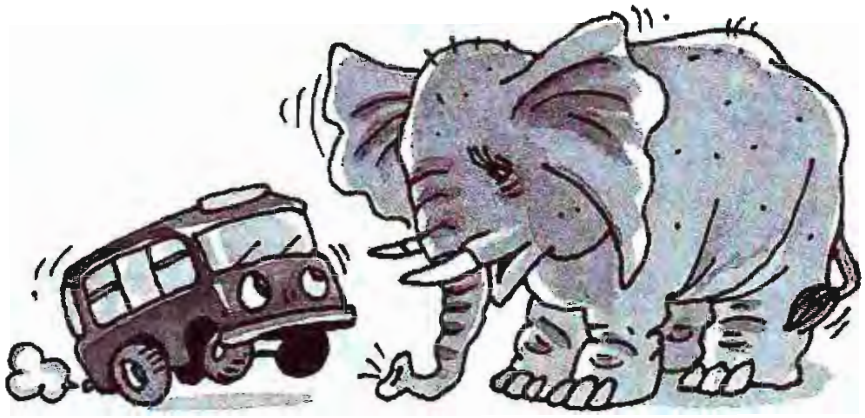
Это верхняя часть силуэта, которая содержит заголовок алгоритма и иконы «Имя ветки»

Зачем нужна шапка

Чтобы быстро понять алгоритм и ответить на три царских вопроса:

- **как называется алгоритм?**
- **из скольких частей он состоит?**
- **как называется каждая часть?**

§ 66. ЧТО ЛУЧШЕ: ПРИМИТИВ ИЛИ СИЛУЭТ?



Мурзик. Хочу нарисовать алгоритм «Поездка на автобусе».

Папа Циркуль. Рисуй — кто тебе мешает?

Мурзик. Как лучше изобразить задачу: в виде примитива или в виде силуэта?

Папа Циркуль. Попробуй оба способа и сравни.

Мурзик (*рисует два алгоритма на рис. 89 а и б*). Готово!

Папа Циркуль. Какой из них тебе больше нравится?

Мурзик. Конечно, силуэт на рис. 89*а*.

Папа Циркуль. Почему?

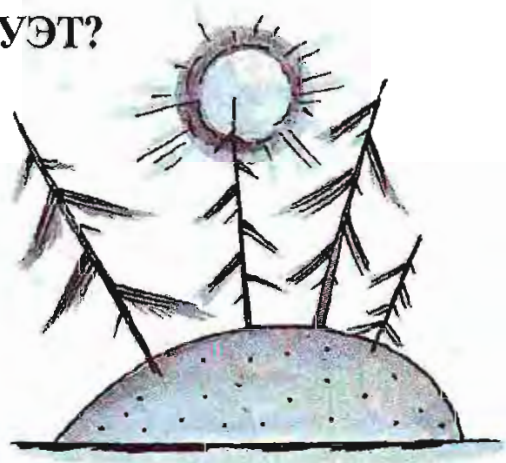
Мурзик. Примитив на рис. 89*б* не позволяет увидеть деление задачи на смысловые части. Иное дело рис. 89*а*. Тут сразу ясно — алгоритм состоит из четырех частей:

- Поиск автобуса
- Ожидание посадки
- Посадка в автобус
- Поездка

Папа Циркуль. Пожалуй, ты прав. В силуэте смысловые части алгоритма четко выделены. Они зрительно и пространственно разнесены в поле чертежа, делая его более понятным. А в примитиве смысловые части не выделены и перемешаны («все в одной куче»), что затрудняет чтение и анализ сложных алгоритмов.

ЧТО ВЫБРАТЬ: ПРИМИТИВ ИЛИ СИЛУЭТ?

Елки-палки, лес густой!
Если алгоритм простой,
Под веселенький мотив
Мы начертим примитив.
Если же у нас с тобой
Алгоритм, как слон, большой,
Здесь, друзья, сомнений нет:
Нужно выбрать силуэт!



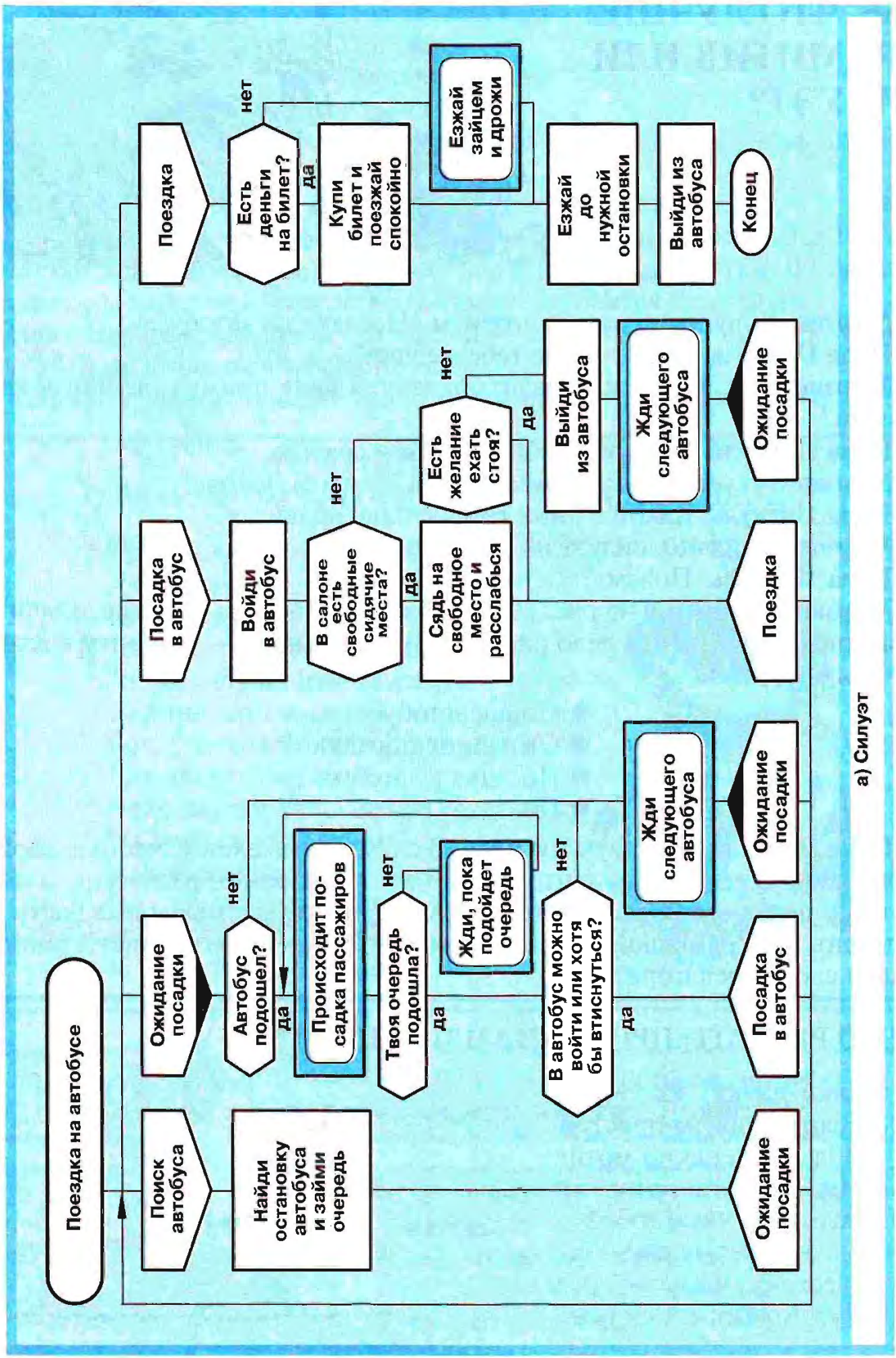
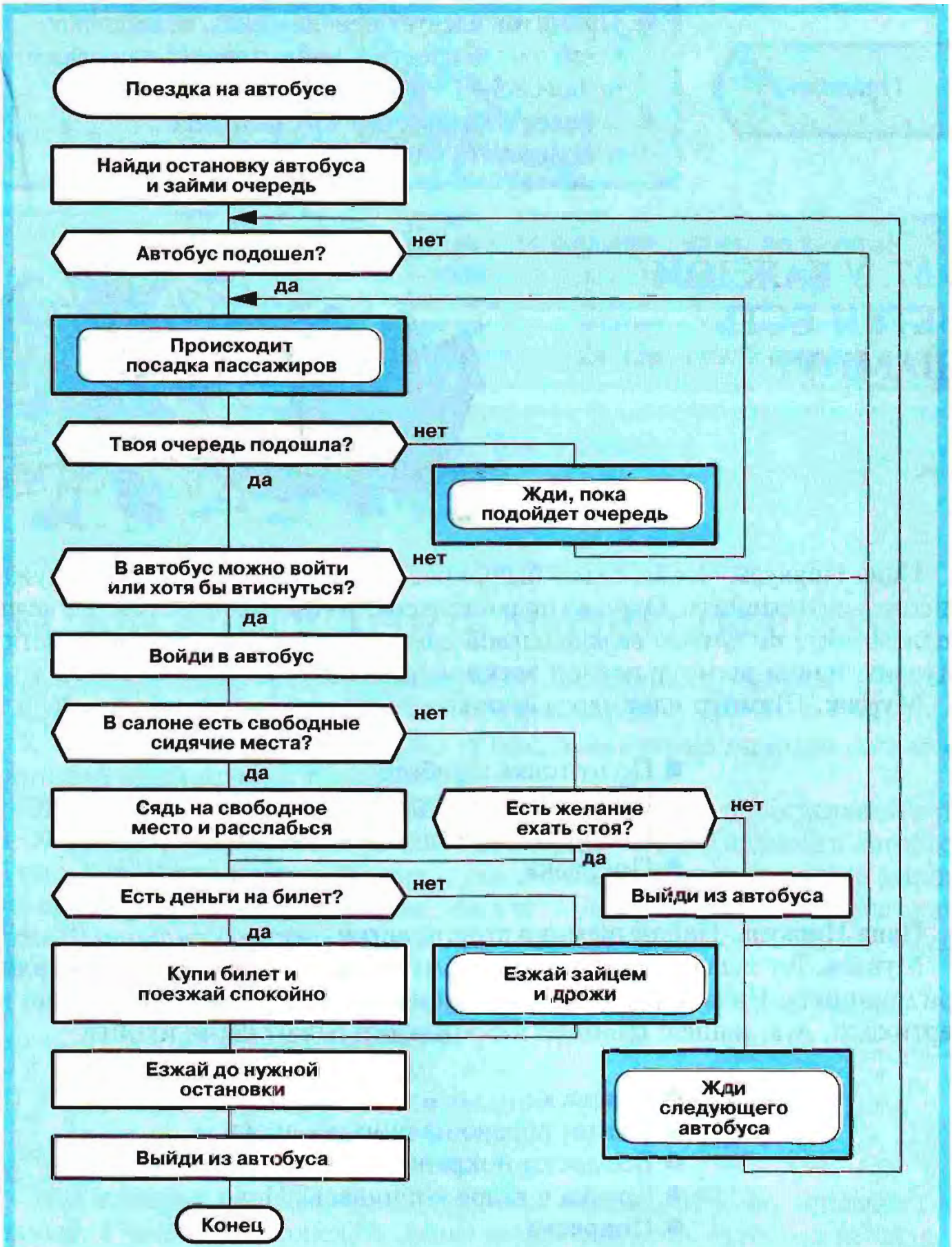


Рис. 89. Алгоритм «Поездка на автобусе»



б) Примитив

Рис. 89. Продолжение

Правило

- **Примитив следует использовать, если дракон-схема очень простая (примитивная) и содержит не более 5—15 икон**
- **В более сложных случаях выгоднее использовать силуэт**

§ 67. У КАЖДОЙ ВЕТКИ ЕСТЬ ШАМПУР

Папа Циркуль. Чтобы ветка была красивой (эргономичной), ее нужно рисовать по правилам. Одно из правил гласит: *левый маршрут каждой ветки должен идти по прямой вертикальной линии.* Эта линия — шампур ветки. Мурзик, найди шампур первой ветки на рис. 88.

Мурзик. Шампур идет через иконы:

- Подготовка к работе
- Возьми ведро с краской и кисть
- Подойди к левому краю забора
- Покраска

Папа Циркуль. Найди шампур второй ветки.

Мурзик. Тут задачка потруднее, потому что ветка разветвленная — в ней три маршрута. Надо найти самый левый из них и убедиться, что он идет по вертикали. Ага, нашел! Шампур второй ветки бежит через иконы:

- Покраска
- Шагни вправо на ширину доски
- Все доски покрашены? Нет
- Краска в ведре кончилась? Нет
- Покраска

Алина (*жалобно и печально*). Я, к сожалению, ничего не поняла. Мурзик, голубчик, придумай мне понимательный стишок.

Мурзик. Просьба дамы для меня закон.

Знают мамы, знают детки:
Левый путь у каждой ветки,
Вертикальный и прямой,
Сверху вниз летит стрелой!
Левый путь, прямой, как шнур,
Называется шампур.

Что такое шампур ветки → **Это вертикальная линия, по которой проходит левый маршрут ветки**

Запомни → **Каждая ветка имеет свой шампур**

Задачи Миши Проверялкина

1. Найди шампуры всех веток на рис. 86—89.
2. Найди шапки на рис. 85—89.

§ 68. КАК УПОРЯДОЧИТЬ МАРШРУТЫ ВЕТКИ

В добрые старые времена мы выучили два хороших правила:

1. Главный маршрут должен идти по шампуру.

2. Побочные маршруты нужно упорядочить слева направо согласно принципу «Чем правее, тем хуже».

Эти правила годятся не только для примитива, но и для отдельной ветки. Хочешь пример? Пожалуйста. Рассмотрим ветку «Посадка в автобус» на рис. 89*а*. Ветка имеет три маршрута. Левая вертикаль (главный маршрут) описывает наибольший успех, так как ты будешь ехать в автобусе сидя. Правый маршрут означает наименьший успех, поскольку ты вышел из автобуса, так никуда и не уехав.

Наиболее интересен средний маршрут, проходящий через иконы:

- Посадка в автобус
- Войди в автобус
- В салоне есть свободные сидячие места? Нет
- Есть желание ехать стоя? Да
- Поездка

Этот маршрут занимает промежуточное положение — он описывает не полный, а частичный успех. В самом деле, ты будешь ехать, но не сидя, а стоя, т. е. поездка состоится (этот хорошо), но без комфорта (это плохо).

Таким образом, все три маршрута данной ветки упорядочены слева направо по принципу «Чем правее, тем хуже». Проверь, что данное правило соблюдается и в других ветках (кроме веток с циклом).

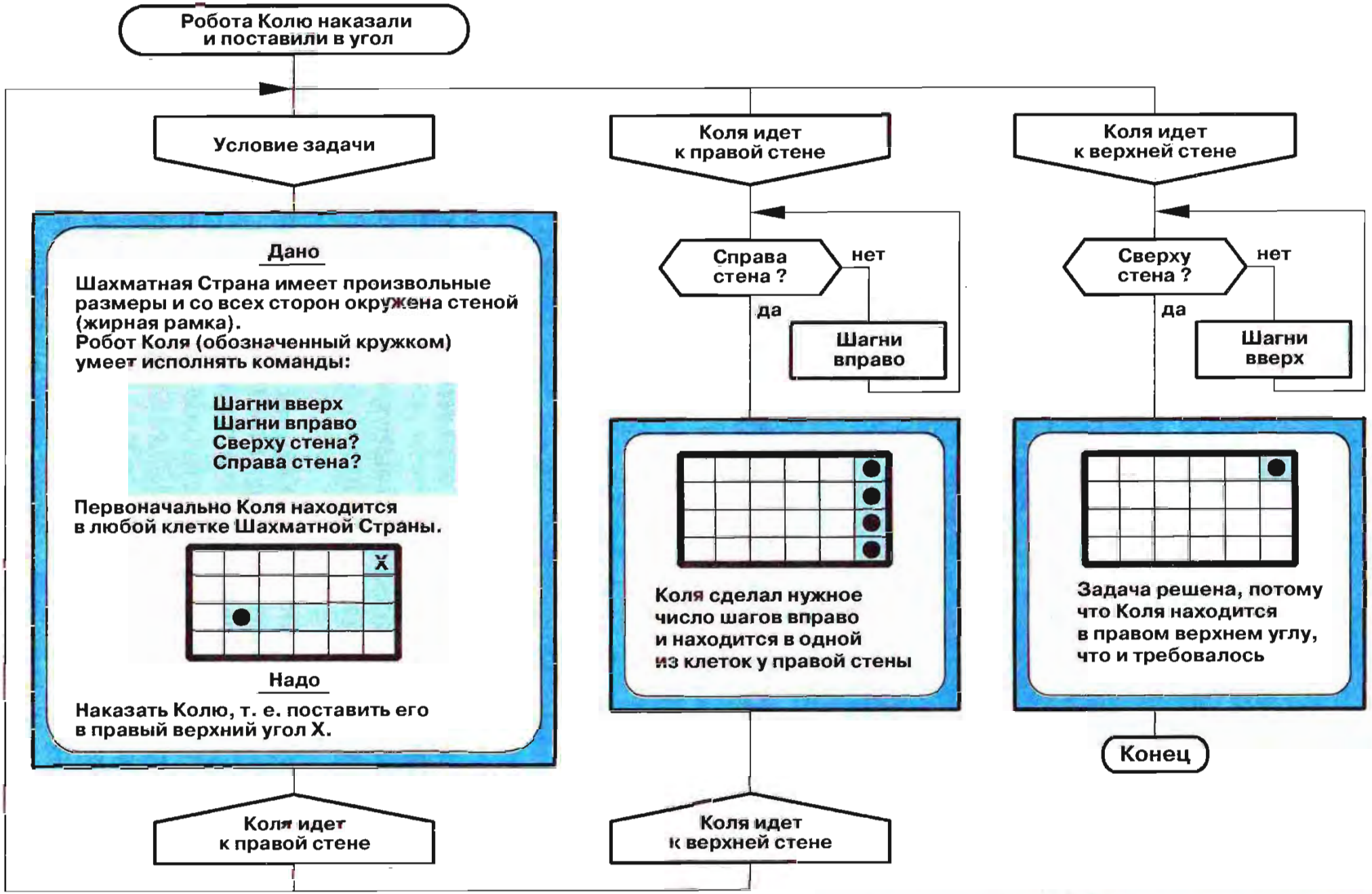


Рис. 90. Алгоритм, приказывающий роботу Коле встать в угол

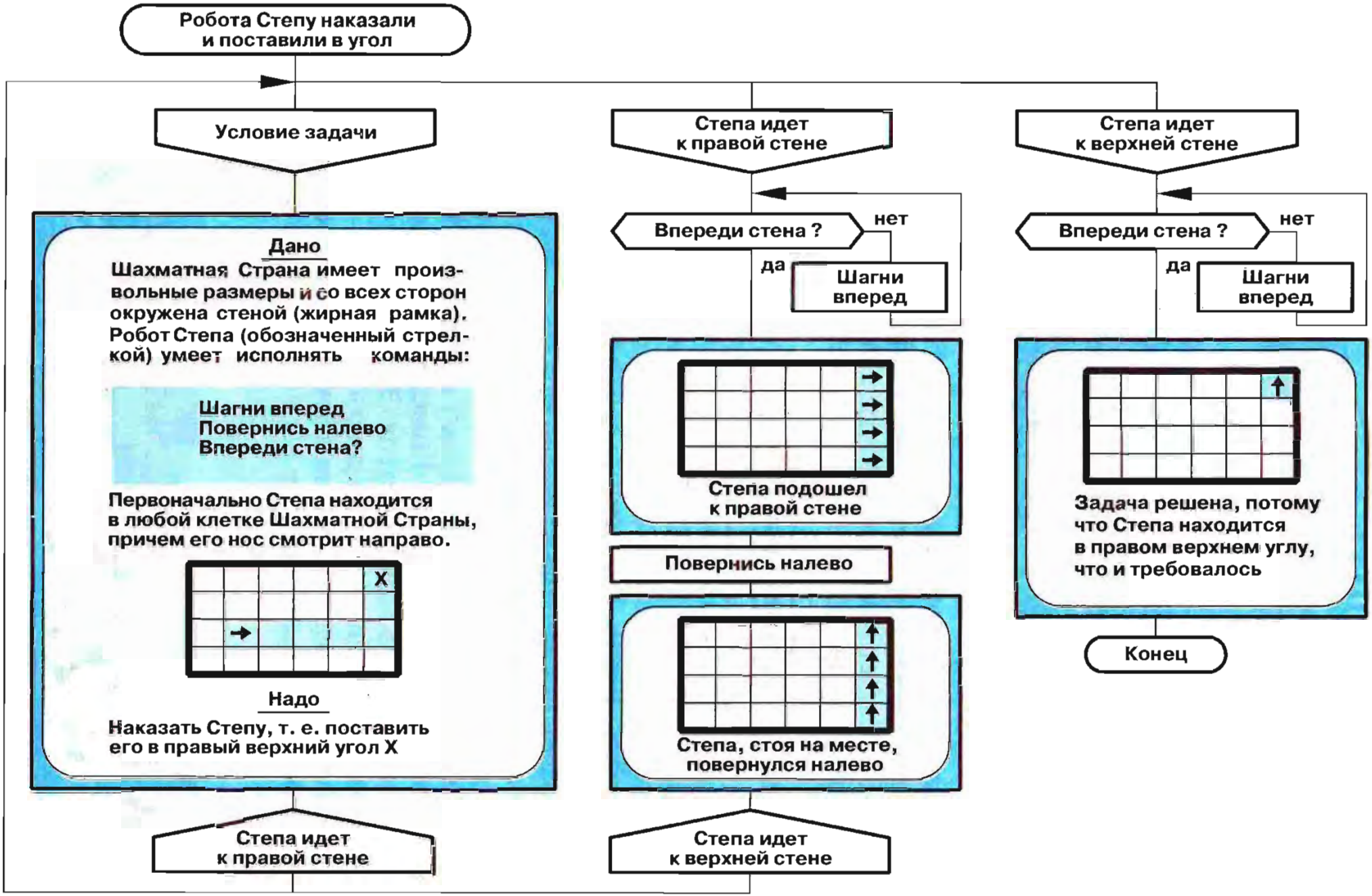


Рис. 91. Алгоритм, приказывающий роботу Степе встать в угол

§ 69. КАК ПОСТРОИТЬ ЦИКЛИЧЕСКИЙ АЛГОРИТМ ДЛЯ КОЛИ И СТЕПЫ?

Папа Циркуль. Робот Коля провинился — его нужно поставить в угол.

Мурзик. В какой угол?

Папа Циркуль. В правый верхний угол Шахматной Страны (рис. 90).

Мурзик. Но как это сделать? Ведь размеры страны неизвестны, а место, где прячется Коля, — тем более.

Папа Циркуль. А нам и не нужно их знать. Ты лучше скажи: чем отличается правый верхний угол от всех прочих клеток (рис. 90)?

Мурзик. Существует только одна клетка, для которой одновременно выполняются два условия:

Справа стена?	=	да
Сверху стена?	=	да

Это и есть правый верхний угол.

Папа Циркуль. Верно. Записанные тобой условия являются подсказкой к решению задачи. А само решение показано на рис. 90.

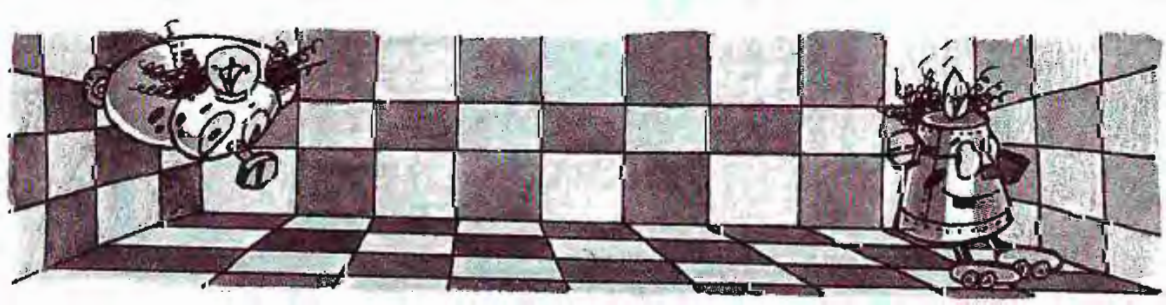
Мурзик. Зачем нужен цикл во второй ветке?

Папа Циркуль. Выполняя цикл, Коля подходит вплотную к правой стене.

Мурзик. А зачем нужен цикл в третьей ветке?

Папа Циркуль. Выполняя цикл, Коля двигается вдоль правой стены вверх, пока не упрется в верхнюю стену. Это значит, что он пришел куда нужно и оказался в правом верхнем углу.

Кстати, робот Степа тоже провинился и тоже был поставлен в угол — смотри алгоритм на рис. 91.



§ 70. КОЛЯ СТОРОЖИТ СКЛАД С КОНФЕТАМИ

Робот Коля получил задание — охранять склад, где хранятся изумительные шоколадные конфеты, пирожные, мороженое и прочая потрясающая вкуснятина.

— Я тоже хочу охранять такой склад! — завопил Мурзик, и у него потекли слюнки.

— Между прочим, роботы не едят конфет, — заметил Папа Циркуль. — Имено поэтому сторожем поставили Колю, а не тебя.

Мурзик покраснел и сказал: А что должен делать сторож?

Папа Циркуль. Он должен непрерывно ходить вокруг склада в течение рабочей смены — восемь часов. Алгоритм, решающий задачу, имеет пять циклов (рис. 92).

Первый цикл содержит иконы «Шагни вправо» и «Сверху стена?». Выполняя эти команды, Коля перемещается от исходной позиции до правого конца склада.

Второй цикл включает иконы «Шагни вверх» и «Слева стена?». Этот цикл заставляет Колю двигаться вверх вдоль правой стены склада.

Следующие два цикла находятся в третьей ветке. Они позволяют Коле завершить путешествие вокруг склада и вернуться в исходную точку.

Пятый цикл — веточный. Он содержит условие: «Прошло 8 часов?» Если не прошло, Коля будет снова и снова ходить вокруг склада. Однако через восемь часов его смена завершится и алгоритм кончит работу.

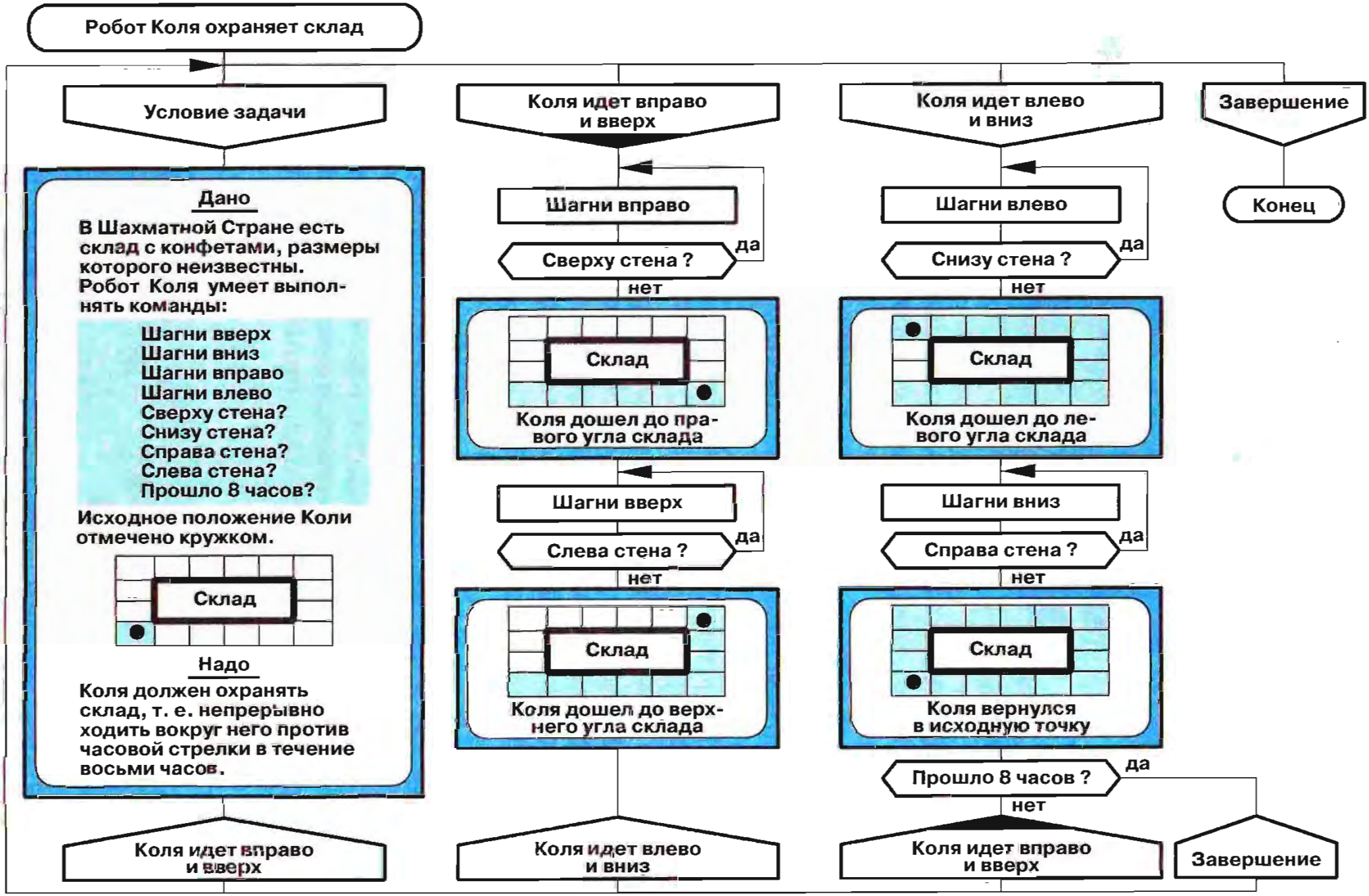


Рис. 92. Алгоритм, приказывающий роботу Коле охранять склад с конфетами

Вопросы Саши Ехидного

1. Сколько циклов на рис. 90? Как они называются?
2. Сколько условий на рис. 90? Назови их.
3. Какие значения принимают условия в начале и конце алгоритма (рис. 90), если Шахматная Страна имеет размеры 10 ×10 клеток, а робот Коля первоначально находится:

- в правом нижнем углу,
- в правом верхнем углу,
- в левом верхнем углу.

Задачи Миши Проверялкина

1. Шахматная Страна имеет 7 клеток по горизонтали и 5 по вертикали. Первоначально Коля находится в левом нижнем углу. Сколько раз будет выполнена каждая команда на рис. 90?
2. Реши задачу на рис. 90 со следующим дополнением. После того как Коля окажется в правом верхнем углу, его нужно отвести сначала в левый верхний угол, затем в левый нижний угол.
3. Шахматная Страна имеет 10 клеток по горизонтали. Первоначально Степа находится в левом верхнем углу. Сколько раз будет выполнена каждая команда на рис. 91? Какое значение имеют условия в начале и конце работы алгоритма?
4. Какие неприятности произойдут, если на рис. 91 убрать команду «Повернись налево»? Как будет работать алгоритм в этом случае? Будет ли выполняться нижний цикл?
5. Реши задачу на рис. 91 при условии, что первоначально Степин нос направлен влево.
6. В какой клетке будет находиться Коля после выполнения первых трех команд на рис. 92?
7. Сколько раз выполняется каждая команда на рис. 92, если размер склада равен 20 × 20 клеток, а Коля обходит склад за 30 минут?
8. Исправь алгоритм на рис. 92. Пусть Коля обходит склад по часовой стрелке. Исходное положение робота то же самое.
9. Исправь алгоритм на рис. 92, считая, что его выполняет робот Степа. В исходном положении Степин нос смотрит вправо.

§ 71. ОШИБКА, КОТОРАЯ НАЗЫВАЕТСЯ «СИАМСКИЕ БЛИЗНЕЦЫ»

Папа Циркуль. Сколько входов у ветки?

Мурзик. У ветки один-единственный вход — икона «Имя ветки». Именно через эту икону дракон-поезд въезжает в ветку.

Папа Циркуль. Верно. Однако некоторые рассеянные с улицы Бассейной забывают это правило и делают грубую ошибку, которая называется «сиамские близнецы».

Мурзик. Что это за ошибка?

Папа Циркуль. Ветку можно сравнить с человеческим телом. Поэтому ветки, как и тела, не должны «срастаться» друг с другом. На рис. 93*б* показано, что две ветки «срослись» с помощью линии XY. Эта линия является ошибочной, вредной.

Мурзик. Почему вредной?

Папа Циркуль. У правой ветки на рис. 93*б* два входа — а должен быть только один!

Мурзик. Откуда же взялся второй вход?

Папа Циркуль. Паразитная линия XY как раз и образует второй вход. По этой незаконной линии дракон-поезд может проехать из левой ветки в правую, что недопустимо.

Мурзик. Как же быть?

Папа Циркуль. Чтобы избавиться от «сиамских близнецов», необходимо разделить «сросшиеся» ветки. Короче говоря, надо изъять из силуэта запрещенную линию XY. Однако делать это нужно аккуратно, чтобы смысл алгоритма не изменился.

Мурзик. Понятно.

Папа Циркуль. Рассмотрим пример. На рис. 94*а* представлен алгоритм-силуэт. Он тебе нравится?

Мурзик. Нет, не нравится. Потому что его испортили «сиамские близнецы».

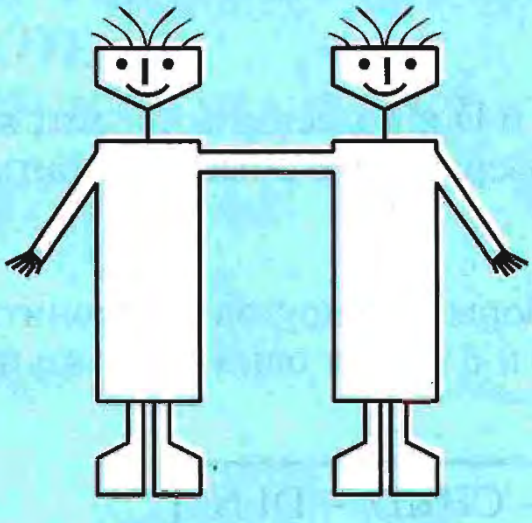
Папа Циркуль. В чем это проявляется?

Мурзик. Между первой и второй ветками нарисована паразитная линия XY. От нее нужно избавиться.

Папа Циркуль. А как это сделать?

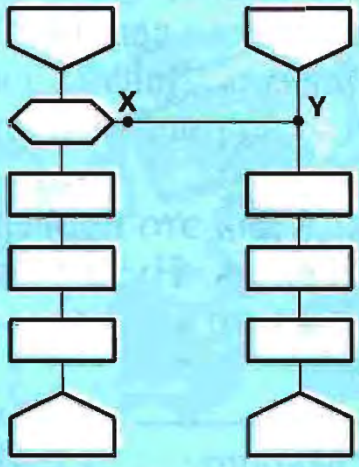
СИАМСКИЕ БЛИЗНЕЦЫ,

у которых срослись плечи



ОШИБКА «СИАМСКИЕ БЛИЗНЕЦЫ»:

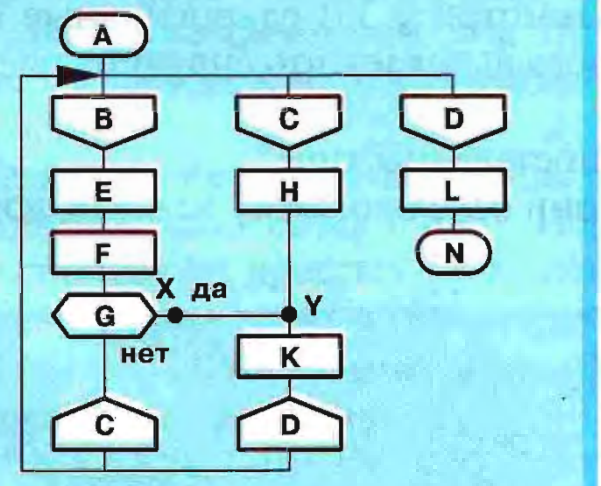
у силуэта «срослись» ветки



Линия XY является запрещенной

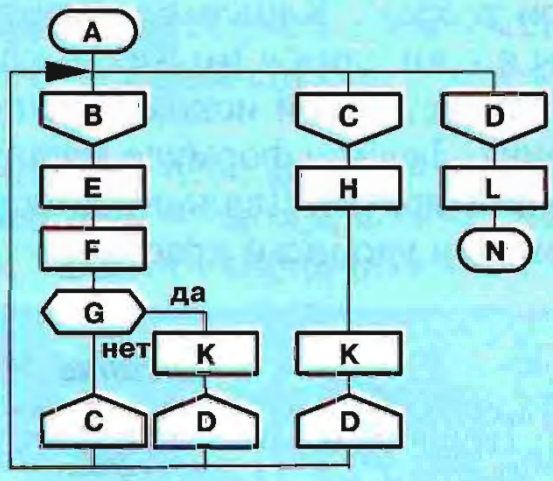
а *б*

Рис. 93. Ошибка «сиамские близнецы» — это запрещенная связь между ветками



Маршрут 1. ABEFG нет C — CHKD — DLN
Маршрут 2. ABEFG да KD — DLN

**Плохая дракон-схема.
Имеется ошибка
«сиамские близнецы».
Линия XY является запрещенной.**



Маршрут 1. ABEFG нет C — CHKD — DLN
Маршрут 2. ABEFG да KD — DLN

**Хорошая дракон-схема.
Ошибка исправлена.
Чтобы убрать запрещенную линию
XY, пришлось удвоить иконы K и D.**

а *б*

Рис. 94. Как исправить ошибку «сиамские близнецы»?

Мурзик. Очень просто. Стереть ее — и дело с концом!

Папа Циркуль. Ай-я-яй! Какой позор! Ты опять наступил на те же грабли! Потому что вместо «сиамских близнецов» у тебя появилась другая ошибка — «висячий хвост».

Мурзик. Какой хвост? Откуда?

Папа Циркуль. Правый выход иконы G, идущий через точку X — это и есть «висячий хвост».

Мурзик. Что же делать?

Папа Циркуль. Давай удвоим иконы K и D и повесим их на этот хвост, как показано на рис. 94*б*. Теперь надо проверить, что смысл алгоритма не изменился.

Мурзик. А как это проверить?

Папа Циркуль. Нужно нарисовать наборы маршрутов и сравнить их. Легко видеть, что алгоритмы на рис. 94 *а* и *б* имеют один и тот же набор маршрутов

Маршрут 1	ABEFG нет C — CHKD — DLN
Маршрут 2	ABEFG да KD — DLN

Следовательно, алгоритм на рис. 94*б* равносилен исходному алгоритму на рис. 94*а* и в то же время не содержит запрещенной линии XY. Согласно первой теореме Кащея Бессмертного (смотри § 33) равносильные алгоритмы имеют один и тот же смысл. Отсюда вытекает, что силуэт на рис. 94 *б* представляет собой искомое решение.

Алина. Зачем в формуле маршрута поставлены тире?

Папа Циркуль. Для наглядности — они отделяют одну ветку от другой. Получается удобно и красиво.

Задание Миши Проверялкина

1. Преобразуй в силуэт алгоритмы на рис. 1—4, 6, 7, 13, 14, 19, 24, 28, 40*б*, 42*б*, 43*в*, 67, 71-75, 77, 78.

АЛГОРИТМИЧЕСКАЯ ОКРОШКА

§ 72. ЧТО ТАКОЕ ПЕРЕКЛЮЧАТЕЛЬ?

Папа Циркуль. Скажи-ка, Мурзик, о чем говорит светофор водителю автомобиля?

Мурзик. Зеленый сигнал говорит: «Жми на газ!» Желтый: «Притормози!» Красный: «Стой! (рис. 95*а*).

Папа Циркуль. Схема на рис. 95*а* позволяет сделать в алгоритме развилку на три направления. Для этого используются две иконы «Вопрос». Однако задачу можно решить и по-другому — с помощью переключателя (рис. 95*б*).

Переключатель — разветвление алгоритма на несколько тропинок, которые потом сливаются в одну. Переключатель — сложная конструкция. Она строится из простых «кирпичиков». Такими «кирпичиками» служат иконы «Выбор» и «Вариант».

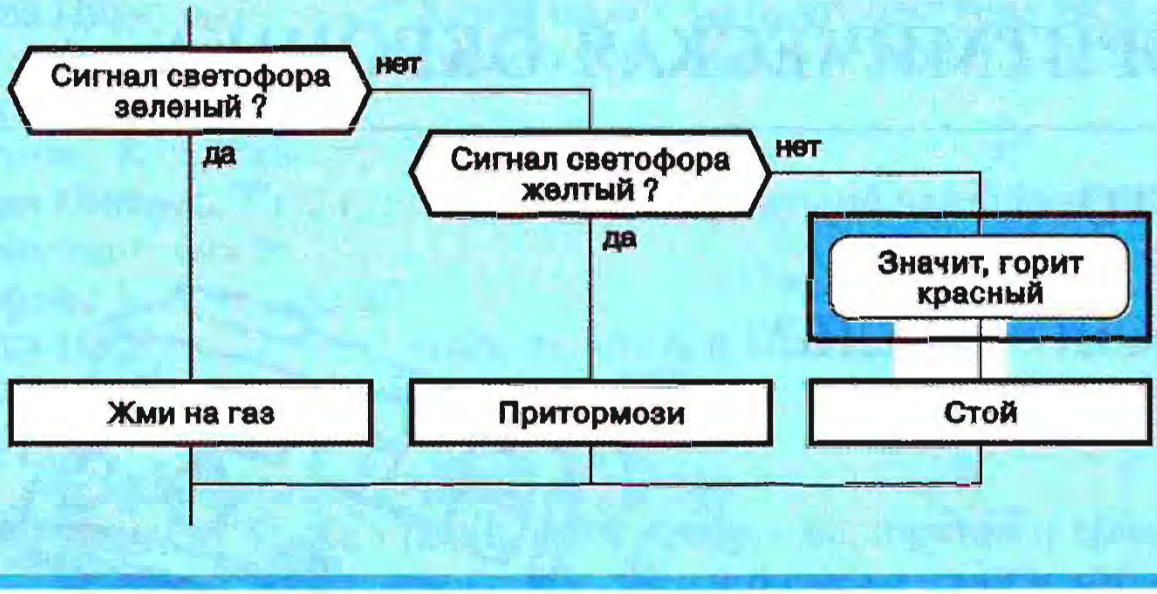
Мурзик. Зачем они нужны?

Папа Циркуль. Икона «Выбор» содержит вопрос, имеющий несколько ответов. Каждый ответ пишут в отдельной рамочке — иконе «Вариант».

Мурзик. Не понимаю. На рис. 95*б* в иконе «Выбор» нет никаких вопросов — там написано слово «Светофор».

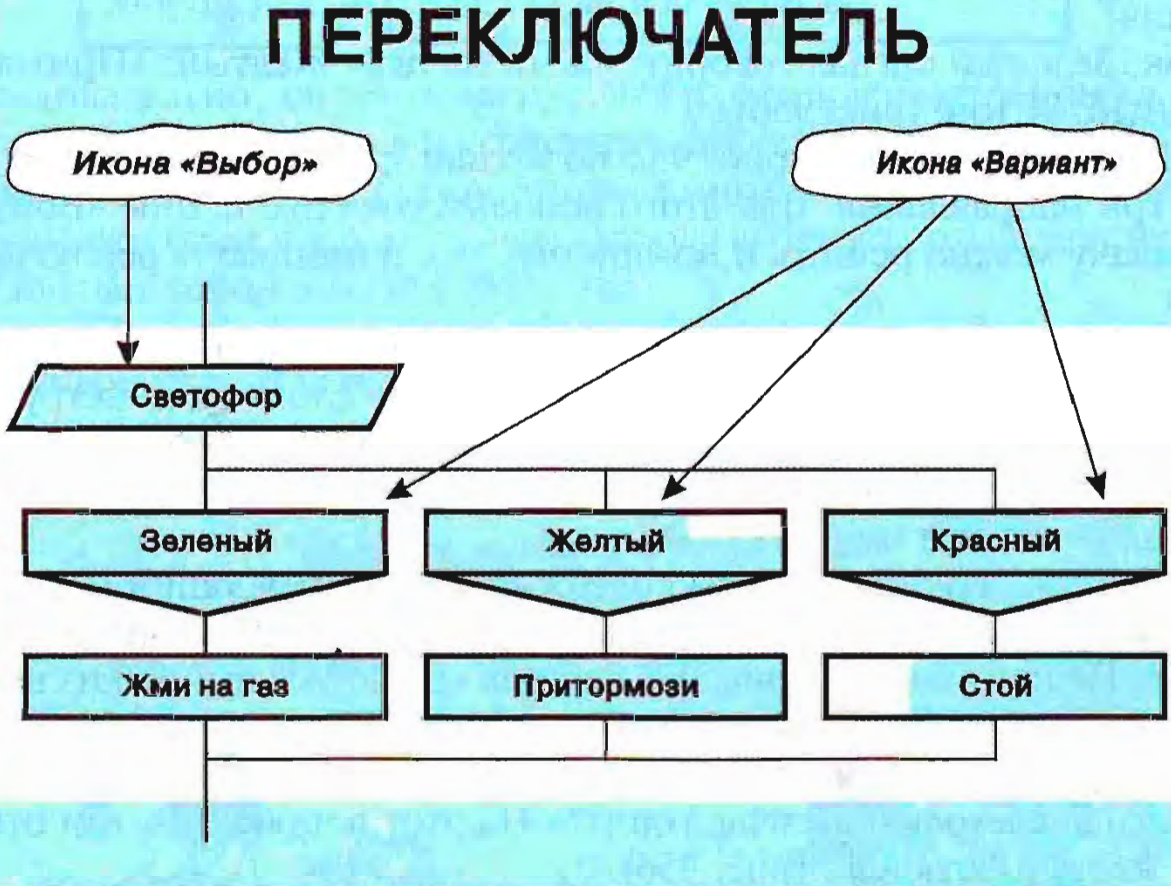
Папа Циркуль. А ты смекни! Слово «светофор» следует понимать так: «Какой сигнал светофора сейчас горит?» На этот вопрос есть три ответа: зеленый, желтый, красный (рис. 95*б*).

Мурзик. Ура, я понял! Например, на рис. 96 фразу «Фамилия ученика» можно прочесть как вопрос: «Какая у ученика фамилия?» — и дать четыре ответа: Иванов, Петров, Сидоров, Калугин. То же самое на рис. 97. Вместо «День недели» читаем: «Какой сегодня день недели?» А в иконах «Вариант» перечисляем ответы: понедельник, вторник и т. д.



Развилка на три направления, построенная с помощью иконы «Вопрос»

а



Развилка на три направления, построенная с помощью переключателя

б

Рис.95. Что лучше: икона «Вопрос» или переключатель ?

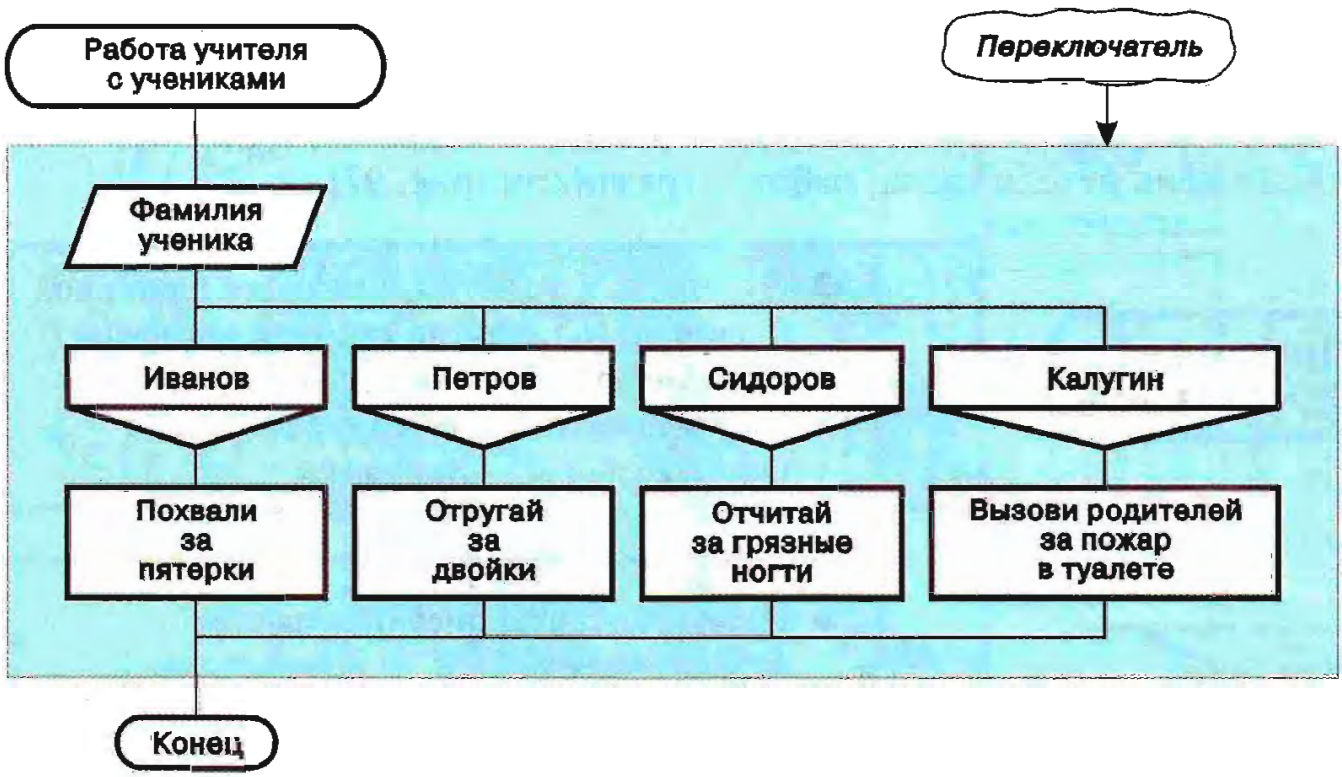


Рис.96. Алгоритм, в котором используется переключатель.

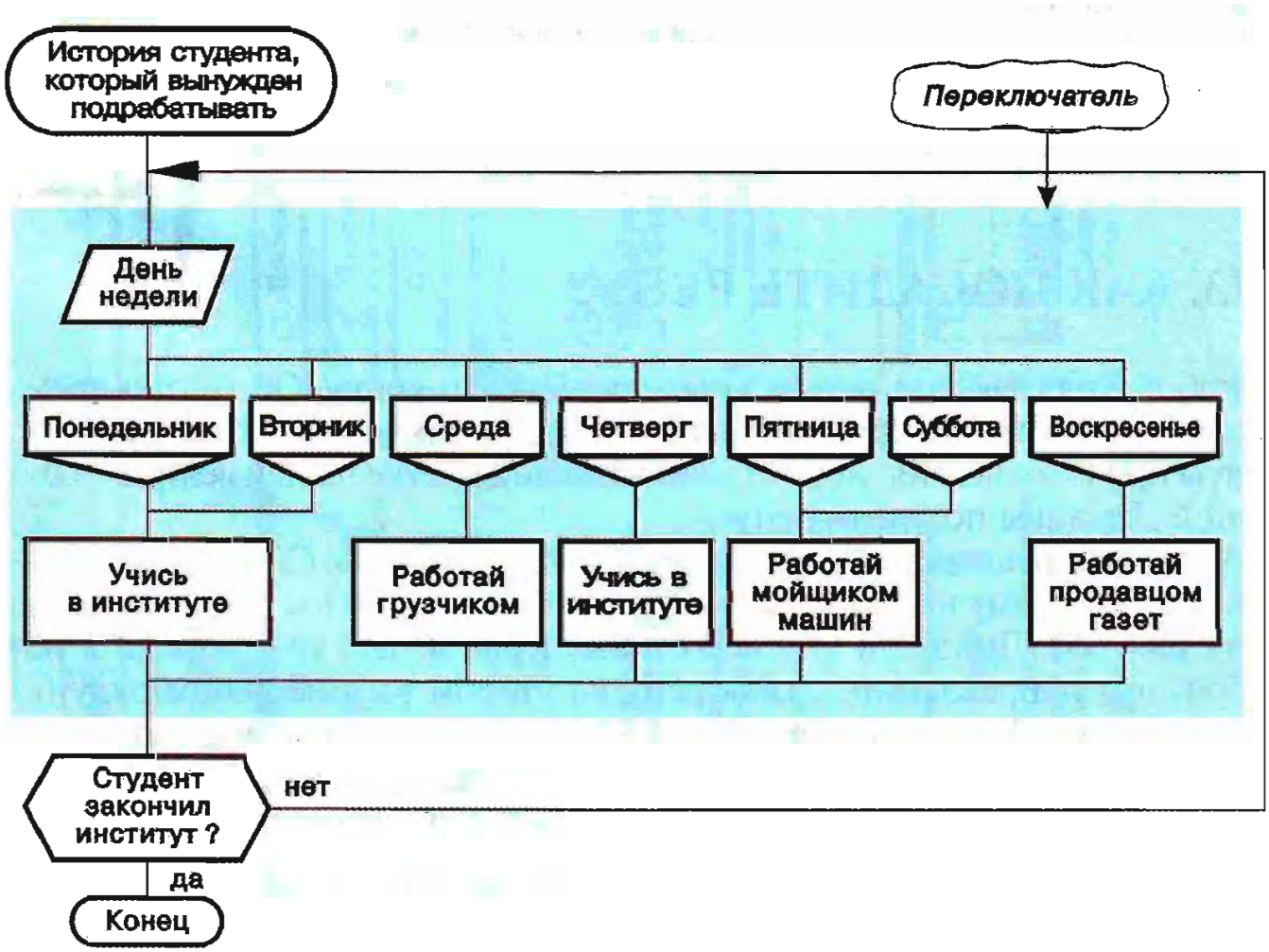


Рис.97. Цикл ДО, в котором есть переключатель.

Папа Циркуль. Опиши, как работает переключатель.
Мурзик. Описание всегда начинается со слова «если».
Если светофор зеленый — жми на газ (рис. 95*б*).
Если фамилия ученика Петров, отругай его за двойки (рис. 96).
Если день недели среда, работай грузчиком (рис. 97).

Что такое переключатель

- **Это часть алгоритма, имеющая один вход и один выход, внутри которой алгоритм разветвляется на несколько дорожек**
- **Переключатель строится с помощью икон «Выбор» и «Вариант»**

Что такое «Выбор»

- **Икона, которую рисуют в начале переключателя**
- **В ней пишут вопрос, имеющий два и более ответов**

Что такое «Вариант»

Икона переключателя, в которой записывают ответ на вопрос

§ 73. КАК ПОСАДИТЬ РЕПУ?

Робот Коля выучил новую команду «Посади репу». Он мирно дремал в верхнем левом углу (рис. 98), когда пришел приказ: «Чтобы увеличить запасы продовольствия, необходимо в каждой клетке нашей великой Шахматной Державы посадить репу».

Алгоритм, решающий задачу, представлен на рис. 98. Сначала Коля идет вниз и сажает репу на вертикальной грядке у левой стены (см. вторую ветку на рис. 98). Дойдя до нижней стены, Коля делает шаг вправо и идет в обратном направлении, сажая репу во втором вертикальном ряду (см. третью ветку). Дойдя до верха, Коля делает еще один шаг вправо и вновь направляется вниз. При этом дракон-поезд попадает в икону-адрес «Коля идет вниз» и возвращается в начало второй ветки. Веточный цикл работает до тех пор, пока Коля не упрется в правую стену. На этом работа заканчивается, потому что все поле засажено репой.

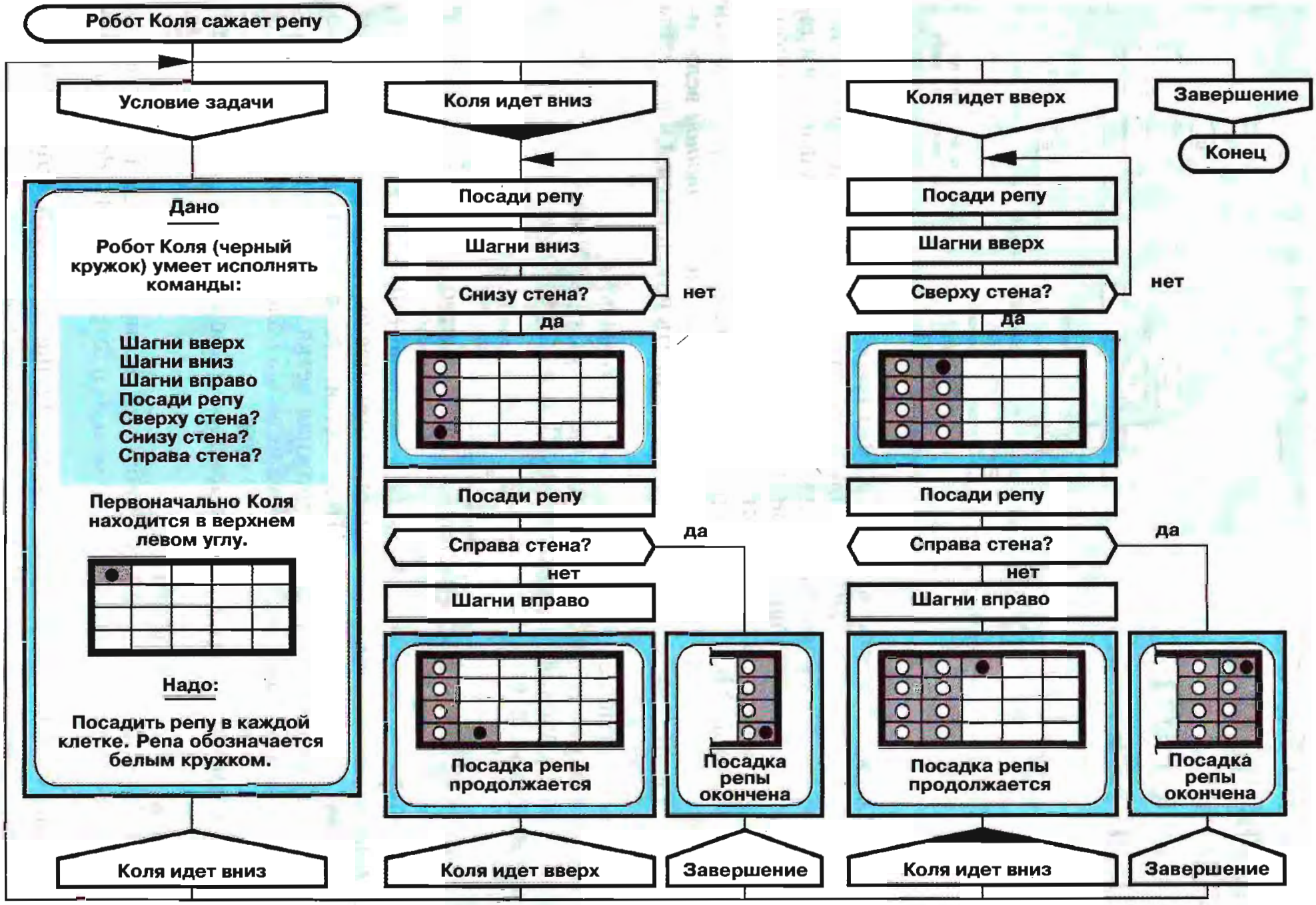


Рис. 98. Алгоритм, приказывающий роботу Коле посадить репу.

§ 74. ЧТО ТАКОЕ ЗАСТЫВШЕЕ УСЛОВИЕ И КАК С НИМ БОРОТЬСЯ?

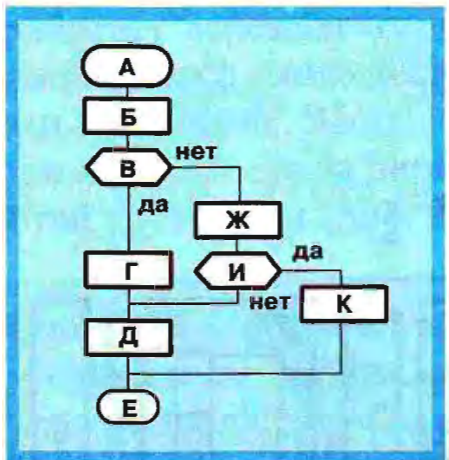


Рис. 99. Алгоритм, содержащий застывшее условие И ≡ да

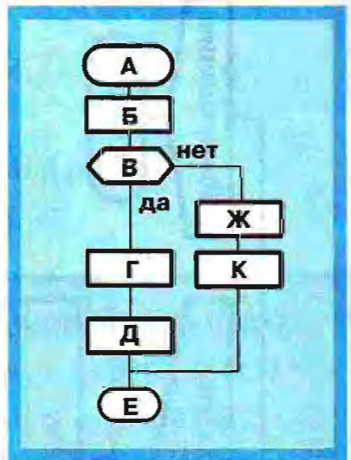


Рис. 100. Тот же алгоритм после удаления застывшего условия

Папа Циркуль. Вспомни, Мурзик, что такое условие?

Мурзик. Это переменная величина, которая принимает два значения: «да» или «нет». Она управляет движением дракон-поезда. Давайте взглянем на рис. 99. Если условие И принимает значение «да» (И = да), дракон-поезд поедет вправо. Если И = нет, он направится вниз.

Папа Циркуль. Молодец, все правильно. А теперь рассмотрим особый случай, когда условие И является не переменной, а *постоянной* величиной. Это значит, что условие теряет способность изменяться. Оно как бы каменеет, становится *застывшим*.

Мурзик. Застывшее условие? Как это понимать?

Папа Циркуль. Представь, что на рис. 99 условие И всегда принимает только одно значение, например «да». В этом случае говорят, что условие И тождественно равно «да».

$$И \equiv да$$

Символ ≡ обозначает тождественное равенство.

Мурзик. Ну и что?

Папа Циркуль. А вот что. Если И ≡ да, значит, на рис. 99 дракон-поезд будет двигаться только вправо и никогда, понимаешь, никогда не поедет вниз.

Мурзик. Ой, как интересно! Получается, что нижний выход развилки И — это заброшенная железнодорожная ветка, где насыпь обвалилась, шпалы давно сгнили, а рельсы провалились в болото. Такая дорога никому не нужна — ведь по ней никто не ездит.

Папа Циркуль. Ты прав. Чтобы эта дорога нас не путала, ее нужно удалить из схемы.

Мурзик. А как это сделать?

Папа Циркуль. Очень просто. Выбросим икону И и заменим ее прямой линией, которая соединяет выход иконы Ж и вход иконы К. Результат показан на рис. 100.

Мурзик. Отлично! Мы убрали из схемы лишние детали, которые никому не нужны. В итоге схема стала проще и понятнее.

§ 75. ОТКУДА ВЗЯЛИСЬ САБЛЕЗУБЫЕ ТИГРЫ?

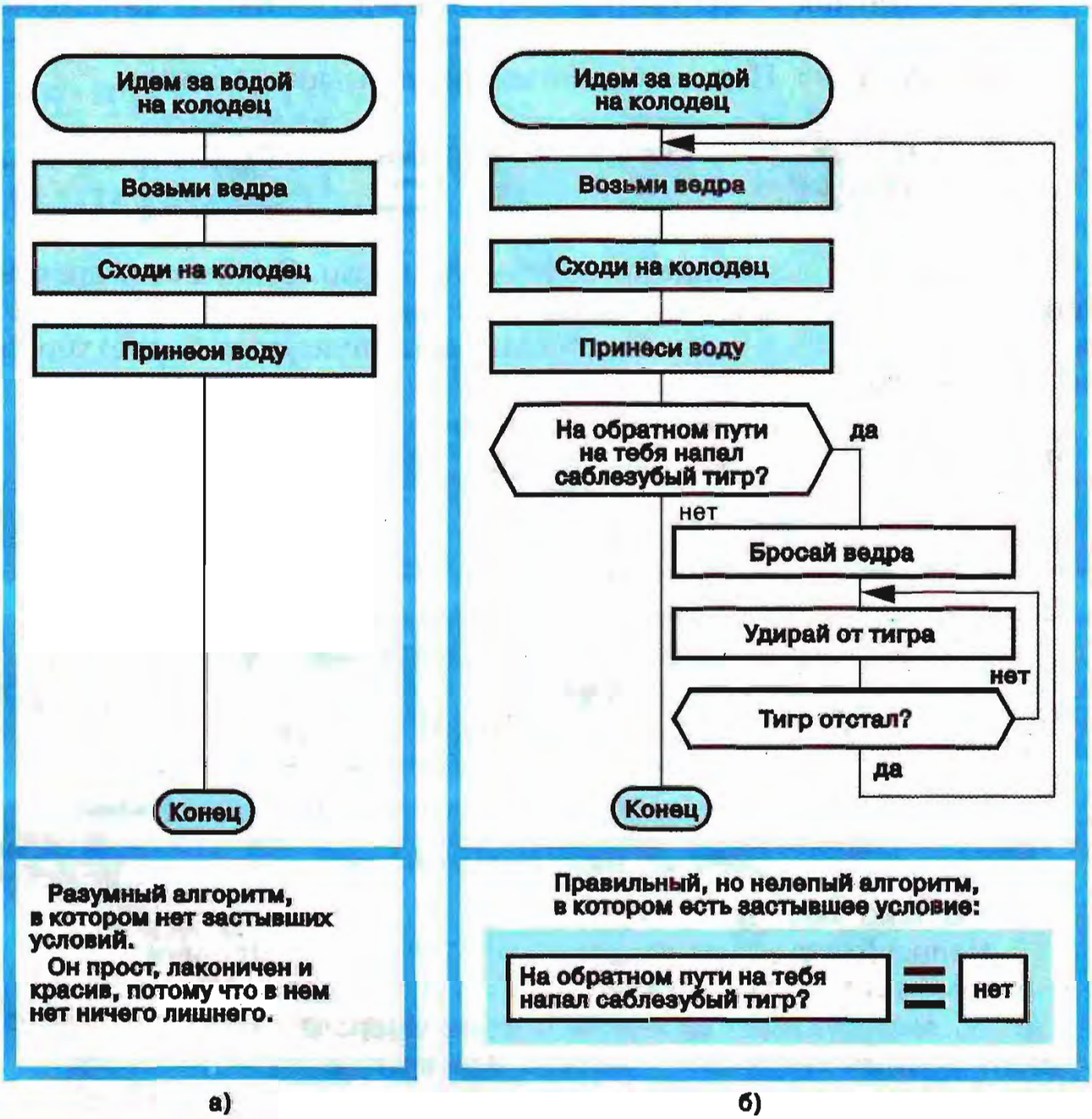


Рис. 142. Застывшие условия приносят вред: они делают алгоритм громоздким, путаным и трудным для понимания.

Папа Циркуль. Чтобы запастись водой, надо взять ведра и сходить на колодец (рис. 101). Однако некий горе-ученый, стремясь «улучшить» алгоритм, добавил в него вопрос насчет саблезубых тигров. Зачем, спрашивается? В нашей деревне тигров отродясь не видали. А саблезубые — те и вовсе вымерли вместе с динозаврами.

Мурзик. Очень интересно! Получается, что никаких тигров здесь нет?

Папа Циркуль. Конечно, нет. Откуда им взяться?

Мурзик. Это значит...

Папа Циркуль. Это значит, что в развилке, где говорится о саблезубых чудищах, дракон-поезд всегда будет двигаться вниз и никогда не повернет направо.

Мурзик. Все ясно. Перед нами типичное застывшее условие:

Папа Циркуль. Застывшие условия — это плохо. Они делают дракон-схему громоздкой, путаной и отвлекают внимание от главного. Чтобы алгоритм стал наглядным и эргономичным, застывшие условия надо убрать, как показано на рис. 101*а*.

Что такое застывшее условие

Это условие, которое никогда не меняет свое значение

Запомни

- **Застывшие условия усложняют схему и приносят вред**
- **Их следует удалить из алгоритма**

Задания Миши Проверялкина

1. Сколько маршрутов на рис. 96?
2. Нарисуй буквенную дракон-схему для рис. 97. Напиши для нее формулы всех маршрутов.
3. Нарисуй буквенную схему переключателя с двумя иконами «Вариант» и двумя иконами «Действие». Преврати ее в смысловую дракон-схему.
4. Придумай смысловую дракон-схему, содержащую переключатель с тремя иконами «Вариант».
5. Шахматная Страна на рис. 98 имеет один столбец. Сколько раз дракон-поезд пройдет по первой ветке и сколько по второй?
6. Ответь на вопрос 5 для случаев, когда Шахматная Страна имеет 2, 3, 4, 5, 6 и 7 столбцов.
7. Преобразуй схему на рис. 99 для следующих случаев:
   - В ≡ да
   - В ≡ нет
   - И ≡ нет
   - В ≡ да, И ≡ нет
   - В ≡ нет, И ≡ да
   - В ≡ нет, И ≡ нет

КАК СТАТЬ УМНЕЕ?

§ 76. ЧТО ТАКОЕ АЛГОРИТМИЧЕСКОЕ МЫШЛЕНИЕ?

Пароход уперся в берег.
Капитан кричит: вперед!
Как такому ротозею
Доверяют пароход?

Папа Циркуль. В чем ошибка капитана?

Мурзик. Он действует глупо, бестолково, без всякого плана. По принципу «тяп-ляп — и готово»!

Папа Циркуль. Верно. Умный человек знает: чтобы не попасть впросак и добиться желаемой цели, нужно заранее продумывать и планировать свои действия.

Алина. Легко сказать. А как это сделать?

Папа Циркуль. Если работа сложная, нужно составить алгоритм ее выполнения. Потому что алгоритм — это заранее составленный детальный план действий. Такой план очень полезен. Он позволяет лучше понять предстоящую работу и выполнить ее более качественно.

Мурзик. Я хочу стать умнее и добиться успеха в жизни. Как это сделать? Можно ли, например, заснуть глупым, а проснуться умным?

Папа Циркуль. Чепуха! Так не бывает. Так умным не станешь. Кто дураком заснул, тот дураком и проснется.

Алина. Как же быть?

Папа Циркуль. Чтобы добиться успеха, нужно упорно работать, развивать свой интеллект, учиться мыслить ясно и отчетливо. Очень важно научиться мыслить *алгоритмически*. Алгоритмическое мышление — замечательная вещь.

Человек, обладающий таким мышлением, составляет алгоритмы легко и быстро. Он щелкает их как орехи! Он сразу схватывает суть сложной задачи и, если нужно, может нарисовать алгоритм ее решения.

Алгоритмическое мышление помогает отчетливо увидеть шаги, ведущие к цели, заметить все препятствия и умело их обойти. Способность к алгоритмическому мышлению — важная черта умного человека.

Что такое алгоритмическое мышление

Это искусство размышлять, умение планировать свои действия, способность предусматривать различные обстоятельства и поступать соответственно с ними

§ 77. СЕКРЕТЫ ПАПЫ ЦИРКУЛЯ

Папа Циркуль. А теперь слушай меня внимательно.

Ты ходишь в школу, чтобы научиться решать учебные и жизненные задачи. Задачи решаются не наобум, а по правилам. Значит, чтобы справиться с задачами, нужно знать правила.

Задачи бывают простые — с ними все ясно. А бывают и сложные. Сложные задачи трудно понять, в них легко запутаться. В жизни много сложной работы и трудных задач. Попадая в сложную ситуацию, ты, как и многие другие, будешь делать досадные ошибки, горевать и тратить уйму времени на их исправление. Поэтому хочу дать тебе несколько советов.

- Чтобы не запутаться в сложной задаче, ее надо представить в ясной, наглядной и доходчивой форме.
- Ты сумеешь избежать многих ошибок в жизни, если научишься описывать сложную работу в виде алгоритма.
- Чтобы развить алгоритмическое мышление, нужно тренироваться. Научись писать алгоритмические сочинения. Постарайся стать чемпионом по сочинению алгоритмов.

§ 78. АЛГОРИТМИЧЕСКОЕ СОЧИНЕНИЕ

Итак, мы добрались почти до конца. Впереди — самое интересное. В этой книжке мы узнали, что такое алгоритмы и с чем их едят. Пришла пора проверить полученные знания. Напиши несколько алгоритмических сочинений, выбрав из списка понравившиеся тебе темы.

Указание

На рис. 102 и 103 представлены примеры алгоритмических сочинений на тему «Поездка на метро» и «Рыбная ловля». При написании собственных сочинений используй эти примеры в качестве образца или подсказки.

ТЕМЫ АЛГОРИТМИЧЕСКИХ СОЧИНЕНИЙ

1. Как построить дом?
2. Как посадить вишневый сад?
3. Как сдать экзамен?
4. Как провести выходной день?
5. Как сходить в пеший туристический поход?
6. Как сходить в поход на байдарках?
7. Как собирать ягоды?
8. Как приготовить борщ?
9. Как сделать табуретку?
10. Как погладить брюки?
11. Как вымыть голову?
12. Как повесить картину?
13. Как вымыть пол?
14. Как спилить дерево?
15. Как сшить рубашку?
16. Как постирать белье?
17. Как привести комнату в порядок?
18. Как собрать и вынести мусор?
19. Как погасить пожар?
20. Как развести костер?

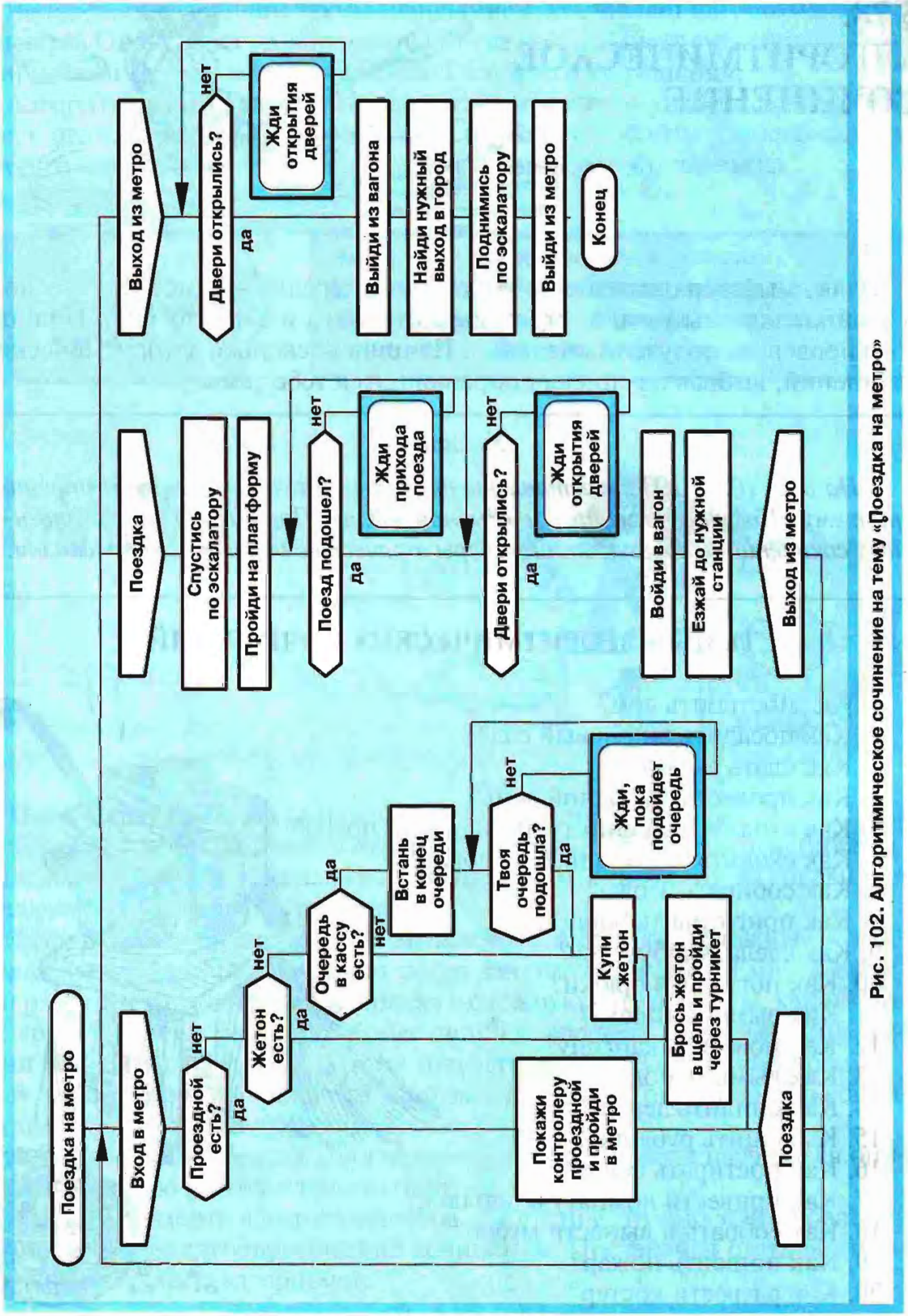


Рис. 102. Алгоритмическое сочинение на тему «Поездка на метро»

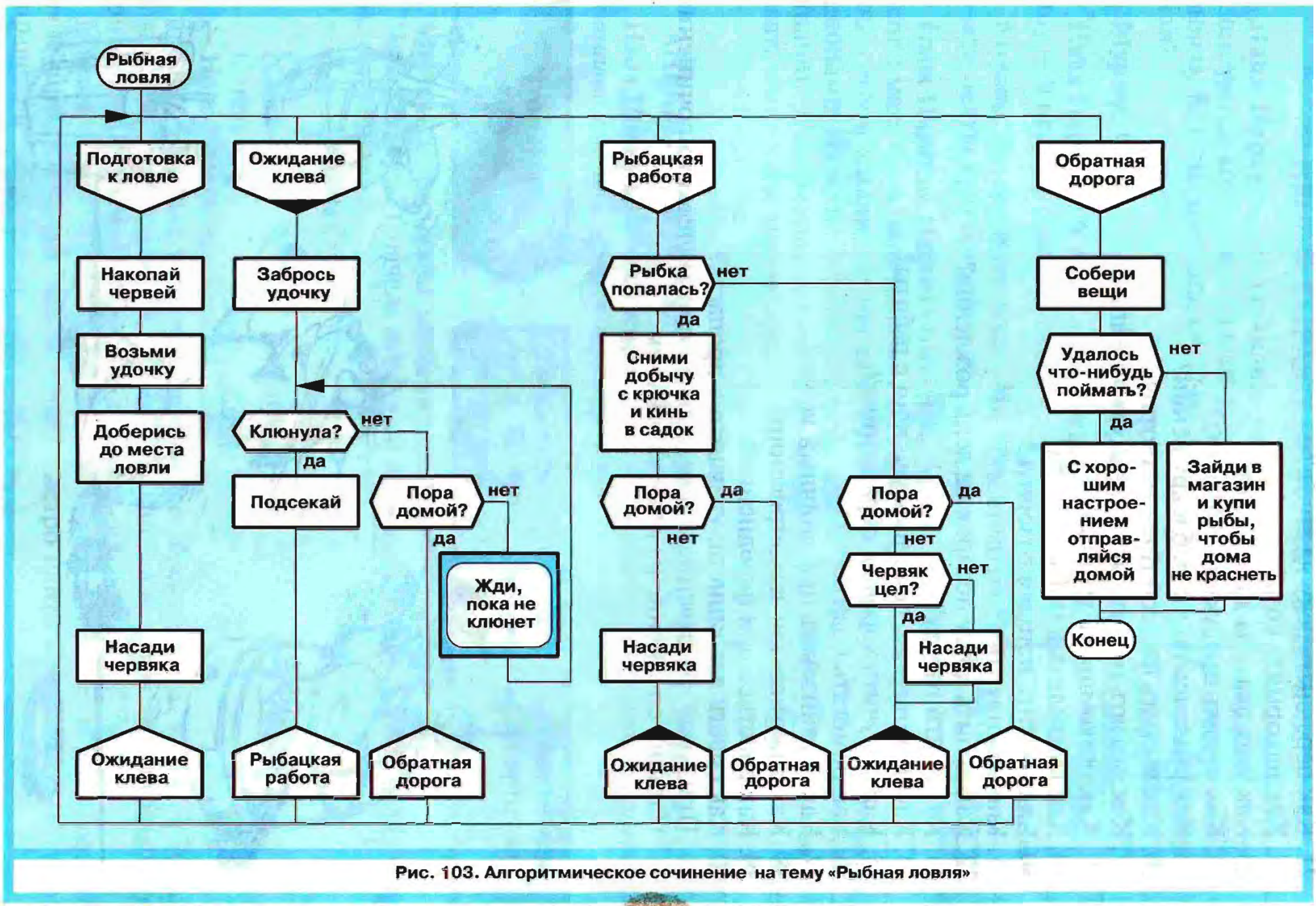


Рис. 103. Алгоритмическое сочинение на тему «Рыбная ловля»

21. Как переплыть реку на лодке?
22. Как накормить кота?
23. Как ухаживать за животными?
24. Как купить продукты в магазине?
25. Как спастись при кораблекрушении?
26. Как выжить на необитаемом острове?
27. Как оказать первую медицинскую помощь?
28. Как утихомирить хулигана?
29. Как выследить и поймать вора?
30. Как научить попугая говорить?
31. Как работать продавцом газет?
32. Как купить другу подарок на день рождения?
33. Как найти работу?
34. Как действовать, если тебя уволили с работы?
35. Как получить хорошее образование?
36. Как помогать родителям?
37. Как организовать праздничный вечер?
38. Как спастись при землетрясении?
39. Как спастись при наводнении?
40. Как спасти альпиниста, упавшего в пропасть?
41. Придумай собственные темы для алгоритмического сочинения. Напиши сочинение на одну из этих тем.

ИКОНЫ БЫВАЮТ НЕ ТОЛЬКО В ЦЕРКВИ

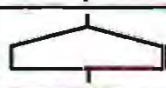
Папа Циркуль. Поздравляю тебя, Мурзик! Наши уроки кончились. Давай займемся повторением. Как ты думаешь, сколько икон мы выучили?

Мурзик. Очень много. Наверное, штук сто!

Папа Циркуль. А вот и нет! Посмотри на таблицу — в ней всего девять икон.

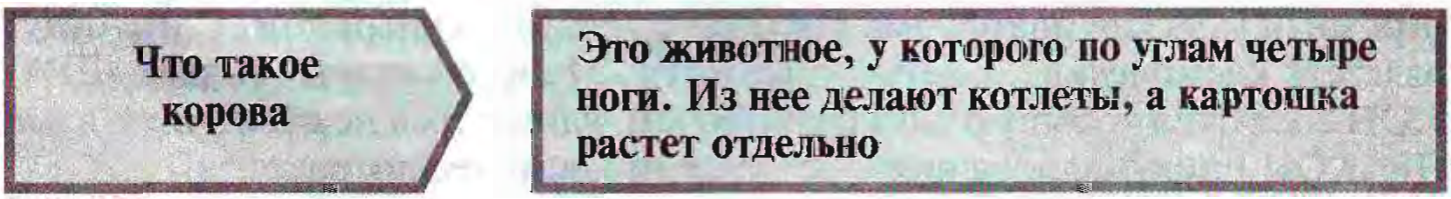
Мурзик. Неужели так мало? Но ведь мы нарисовали целую кучу алгоритмов!

Папа Циркуль. Иконы похожи на буквы. В алфавите мало букв, но из них можно составить тысячи слов и миллионы предложений. Так же и с иконами. Их мало, но из них можно делать комбинации. С помощью девяти икон можно нарисовать миллионы алгоритмов.

Икона	Название иконы
	Заголовок
	Конец
	Действие
	Вопрос
	Комментарий
	Вставка
	Имя ветки
	Адрес
	Часы

АНЕКДОТ С ИЗЮМОМ, ИЛИ ЧТО ТАКОЕ ЗАПОМИНАТЕЛЬ?

Папа Циркуль. В этой книжке в конце многих параграфов нарисован *запоминатель*. Вот пример запоминателя.



Мурзик. Зачем нужен запоминатель?

Папа Циркуль. Сначала послушай анекдот. Приходит один чудак в булочную и говорит:

— У вас батоны есть?

— Есть.

— С изюмом?

— С изюмом.

— Наковыряйте полкило!

Так вот, в книжке тоже есть «изюм», т. е. важные места, которые обязательно нужно запомнить. К сожалению, иногда их очень трудно заметить. Приходится «выковыривать» их из текста, как изюм из булки.

Мурзик. Я понял. Запоминатель позволяет сразу заметить все «изюминки» и «проглотить» их. Он помогает быстро найти самые важные мысли и правила и облегчает их запоминание.

МОРСКОЙ СТАРТ: ЧУДО В ТРОПИКАХ

Хлястик. Я слышал, что ДРАКОН попал на Землю из космоса. Это правда?

Папа Циркуль. На твой вопрос ответит автор нашей книжки. Он ракетчик, ему и карты в руки.

Автор. Весь мир знает о легендарном космическом корабле «Буран». Я тоже участвовал в его создании. Так вот, некоторые алгоритмы «Бурана» написаны на языке ДРАКОН.

Мурзик. Значит, в космосе тоже нужны алгоритмы?

Автор. Еще как!

Ракета — она как слепая коза!
Чтоб был в ней и смысл, и лоск,
Ракете нужно иметь глаза
И самое главное — мозг!

Глаза ракеты — зоркие приборы, позволяющие «подглядеть», куда она летит, и узнать, что творится вокруг. Мозг ракеты — бортовой компьютер. Он может сам, без участия человека, «руководить» космической экспедицией. В электронной памяти компьютера «спрятаны» алгоритмы, управляющие полетом ракеты.

Алина. Как интересно!

Автор. А теперь перенесемся в Тихий океан, где когда-то бороздил морские просторы отчаянный Магеллан. Но сейчас его подвиги затмило новое чудо. Прямо на экваторе, у загадочных островов Кирибати, плавает огромная морская платформа высотой с десятиэтажный дом. С нее будут взлетать космические ракеты по проекту «Морской старт». И здесь ДРАКОН сослужил хорошую службу. Впрочем, удивляться не стоит. Ведь язык ДРАКОН придумали ученые — создатели космических ракет.

Мурзик. Значит, ДРАКОН прилетел к нам из космоса?

Автор. Выходит, так.

Мурзик (*декламирует*).

Дракон-дракон-дракошечка!
Влети ко мне в окошечко!

Алина. Кстати, я давно хотела спросить: у нашей книжки будет продолжение?

Автор (*растерянно*). Продолжение? Нет, не будет.

Алина. Почему?

Автор (*смущенно*). Надоело заниматься писаниной. В жизни столько интересного — глаза разбегаются. Хочется все успеть — и в футбол поиграть, и на каток сходить, и на ракете слетать в соседнюю галактику. Времени нет писать книжки.

Все (*хором*). Ну пожалуйста!

Хлястик. Продолжение обязательно нужно написать. Потому что Информатика — удивительно интересная наука. Я раньше думал, что алгоритмы — это скучно. А тут, оказывается, сплошные приключения!

Алина. Если у нашей книжки не будет продолжения, значит, мы расстаемся навсегда. А это очень грустно. Ведь мы так весело играли и привязались друг к другу. Я так люблю Мурзика и Папу Циркуля, и даже Кащея с его лягушками. Я буду скучать по ним (смахивает слезинку).

Автор (*неожиданно соглашается*). Ладно, так и быть, напишу. Как говорит Мурзик, по просьбе дамы. Если, конечно, ракета на обратном пути не сломается. В общем, через год-другой вас ждет сюрприз.

Мурзик. Ура! Мы снова будем вместе!

(Продолжение следует)

Дорогие родители и учителя!

Понравилась ли вам эта книга? Какие места оказались удачными, а какие — трудными для детей? Справились ли они с задачами?

Я буду весьма признателен, если вы сумеете выкроить время и сообщить мне ваши отзывы, пожелания и критические замечания. Все отклики будут с благодарностью приняты и использованы при переработке книги для следующего издания. Не забудьте указать возраст вашего ребенка.

Письма направляйте по адресу:

117342, Москва, ул Введенского, д. 1, НПО автоматики и приборостроения, отделение 03, Паронджанову Владимиру Даниеловичу

Если вас заинтересовала тема, более подробные сведения можно найти в моей книге:

Паронджанов В. Д. Как улучшить работу ума. (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания). — М.: Радио и связь, 1998. — 352 с. Илл.: 154.

СОДЕРЖАНИЕ

ДЛЯ МЛАДШЕГО И СРЕДНЕГО ШКОЛЬНОГО ВОЗРАСТА
«ШКОЛЬНИКУ ДЛЯ РАЗВИТИЯ ИНТЕЛЛЕКТА»

Паронджанов Владимир Даниелович

ЗАНИМАТЕЛЬНАЯ ИНФОРМАТИКА

Ответственный за выпуск А. Ю. БИРЮКОВА
Технический редактор М. В. ГАГАРИНА
Корректор О. И. ИВАНОВА

Лиц. изд. № 071328 от 09.08.96.

Налоговая льгота — общероссийский классификатор продукции ОК-005-93, том 2; 953000 — книги, брошюры.

Подписано к печати с готовых диапозитивов 18.01.2000.
Формат 70x100 1/16. Печ. л. 9,5. Тираж 12 000 экз.
Заказ № 3547. С – 898.

Издательский дом «Росмэн».
125124, Москва, а/я 62. 1-я ул. Ямского поля, 28.

МЕЛКООПТОВЫЙ СКЛАД:
Москва, 1-я ул. Ямского поля, 28 (левое крыло).
Тел.: (095) 257-34-75.

ОТДЕЛ ОПТОВЫХ ПРОДАЖ:
все города России, СНГ: (095) 257-46-61;
Москва и Московская область: (095) 257-41-32.

Отпечатано на Тверском ордена Трудового Красного Знамени полиграфкомбинате детской литературы им. 50-летия СССР Министерства Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.
170040, Тверь, пр. 50-летия Октября, 46.