

## Глава 2

# ОСНОВЫ АЛГОРИТМИЗАЦИИ

### § 2.1

#### Алгоритмы и исполнители

*Ключевые слова:*

- алгоритм
- свойства алгоритма
  - дискретность
  - понятность
  - определённости
  - результативность
  - массовость
- исполнитель
- характеристики исполнителя
  - круг решаемых задач
  - среда
  - режим работы
  - система команд
- формальное исполнение алгоритма

##### 2.1.1. Понятие алгоритма

Каждый человек в повседневной жизни, в учёбе или на работе решает огромное количество задач самой разной сложности. Сложные задачи требуют длительных размышлений для нахождения решения; простые и привычные задачи человек решает не задумываясь, автоматически. В большинстве случаев решение каждой задачи можно разбить на простые этапы (шаги). Для многих таких задач (установка программного обеспечения, сборка шкафа, создание сайта, эксплуатация технического устройства, покупка авиабилета через Интернет и т. д.) уже разработаны и

предлагаются пошаговые инструкции, при последовательном выполнении которых можно прийти к желаемому результату.

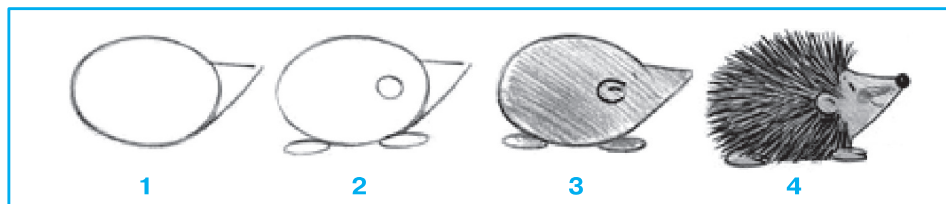
**Пример 1.** Задача «Найти среднее арифметическое двух чисел» решается в три шага:

- 1) задумать два числа;
- 2) сложить два задуманных числа;
- 3) полученную сумму разделить на 2.

**Пример 2.** Задача «Внести деньги на счёт телефона» подразделяется на следующие шаги:

- 1) подойти к терминалу по оплате платежей;
- 2) выбрать оператора связи;
- 3) ввести номер телефона;
- 4) проверить правильность введённого номера;
- 5) вставить денежную купюру в купюроприёмник;
- 6) дождаться сообщения о зачислении денег на счёт;
- 7) получить чек.

**Пример 3.** Этапы решения задачи «Нарисовать весёлого ёжика» представлены графически:



Нахождение среднего арифметического, внесение денег на телефонный счёт и рисование ежа — на первый взгляд совершенно разные процессы. Но у них есть общая черта: каждый из этих процессов описывается последовательностью кратких указаний, точное следование которым позволяет получить требуемый результат. Последовательности указаний, приведённые в примерах 1–3, являются алгоритмами решения соответствующих задач. Исполнитель этих алгоритмов — человек.

Алгоритм может представлять собой описание некоторой последовательности вычислений (пример 1) или шагов нематема-

тического характера (примеры 2–3). Но в любом случае перед его разработкой должны быть чётко определены начальные условия (исходные данные) и то, что предстоит получить (результат). Можно сказать, что **алгоритм** — это описание последовательности шагов в решении задачи, приводящих от исходных данных к требуемому результату.

В общем виде схему работы алгоритма можно представить следующим образом (рис. 2.1).



**Рис. 2.1.** Общая схема работы алгоритма

Алгоритмами являются изучаемые в школе правила сложения, вычитания, умножения и деления чисел, многие грамматические правила, правила геометрических построений и т. д.

Анимации «Работа с алгоритмом» (193576), «Наибольший общий делитель» (170363), «Наименьшее общее кратное» (170390) помогут вам вспомнить некоторые алгоритмы, изученные на уроках русского языка и математики (<http://sc.edu.ru/>).

**Пример 4.** Некоторый алгоритм приводит к тому, что из одной цепочки символов получается новая цепочка следующим образом:

1. Вычисляется длина (в символах) исходной цепочки символов.
2. Если длина исходной цепочки нечётна, то к исходной цепочке справа приписывается цифра 1, иначе цепочка не изменяется.
3. Символы попарно меняются местами (первый — со вторым, третий — с четвёртым, пятый — с шестым и т. д.).
4. Справа к полученной цепочке приписывается цифра 2.

Получившаяся таким образом цепочка является результатом работы алгоритма.

Так, если исходной была цепочка  $A\#B$ , то результатом работы алгоритма будет цепочка  $\#A1B2$ , а если исходной цепочкой была  $AB\@$ , то результатом работы алгоритма будет цепочка  $BA\@B2$ .

www



### 2.1.2. Исполнитель алгоритма

Каждый алгоритм предназначен для определённого исполнителя.

---

**Исполнитель** — это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

---



Различают формальных и неформальных исполнителей. Формальный исполнитель одну и ту же команду всегда выполняет одинаково. Неформальный исполнитель может выполнять команду по-разному.

Рассмотрим более подробно множество формальных исполнителей. Формальные исполнители необычайно разнообразны, но для каждого из них можно указать следующие характеристики: круг решаемых задач (назначение), среду, систему команд и режим работы.

**Круг решаемых задач.** Каждый исполнитель создаётся для решения некоторого круга задач — построения цепочек символов, выполнения вычислений, построения рисунков на плоскости и т. д.

**Среда исполнителя.** Область, обстановку, условия, в которых действует исполнитель, принято называть средой данного исполнителя. Исходные данные и результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен алгоритм. Среду можно рассматривать как полный набор характеристик, описывающих состояние исполнителя.

**Система команд исполнителя.** Различают команды-приказы и команды-запросы. Предписание исполнителю о выполнении отдельного законченного действия отдаётся *командой-приказом*. *Команды-запросы* позволяют узнать текущие характеристики среды исполнителя. Совокупность всех команд, которые могут быть выполнены некоторым исполнителем, образует систему команд данного исполнителя (СКИ). Алгоритм составляется с учётом возможностей конкретного исполнителя, иначе говоря, в системе команд исполнителя, который будет его выполнять.

**Режимы работы исполнителя.** Для большинства исполнителей предусмотрены *режимы непосредственного (ручного) управления и программного управления*. В первом случае исполнитель ожидает команд от человека и каждую поступившую команду немедленно выполняет. Во втором случае исполнителю сначала

задаётся полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме. Ряд исполнителей работает только в одном из названных режимов.

Рассмотрим примеры исполнителей.

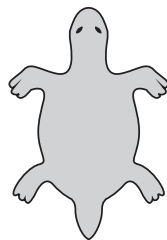


**Пример 5.** Исполнитель **Черепаша** перемещается на экране компьютера, оставляя след в виде линии. Система команд Черепаша состоит из следующих команд:

Вперёд  $n$  (где  $n$  — целое число) — вызывает передвижение Черепаша на  $n$  шагов в направлении движения — в том направлении, куда развёрнуты её голова и корпус;

Направо  $m$  (где  $m$  — целое число) — вызывает изменение направления движения Черепаша на  $m$  градусов по часовой стрелке.

Запись Повтори  $k$  [ $\langle$ Команда1 $\rangle$   $\langle$ Команда2 $\rangle$  ...  $\langle$ Команда $n$  $\rangle$ ] означает, что последовательность команд в скобках повторится  $k$  раз.



Подумайте, какая фигура появится на экране после выполнения Черепашой следующего алгоритма.

Повтори 12 [Направо 45 Вперёд 20 Направо 45]



**Пример 6.** Система команд исполнителя **Вычислитель** состоит из двух команд, которым присвоены номера:

1 — вычти 1

2 — умножь на 3

Первая из них уменьшает число на 1, вторая увеличивает число в 3 раза. При записи алгоритмов для краткости указываются лишь номера команд. Например, алгоритм 21212 означает следующую последовательность команд:

умножь на 3

вычти 1

умножь на 3

вычти 1

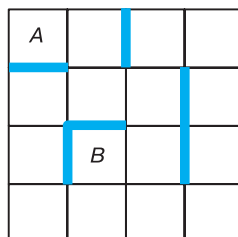
умножь на 3

С помощью этого алгоритма число 1 будет преобразовано в 15:  $((1 \cdot 3 - 1) \cdot 3 - 1) \cdot 3 = 15$ .



**Пример 7.** Исполнитель **Робот** действует на клетчатом поле, между соседними клетками которого могут стоять стены. Робот передвигается по клеткам поля и может выполнять следующие команды, которым присвоены номера:

- 1 — вверх
- 2 — вниз
- 3 — вправо
- 4 — влево



При выполнении каждой такой команды Робот перемещается в соседнюю клетку в указанном направлении. Если же в этом направлении между клетками стоит стена, то Робот разрушается.

Что произойдёт с Роботом, если он выполнит последовательность команд 32323 (здесь цифры обозначают номера команд), начав движение из клетки *A*? Какую последовательность команд следует выполнить Роботу, чтобы переместиться из клетки *A* в клетку *B*, не разрушившись от столкновения со стеной?



**Пример 8.** К пятизначному натуральному числу применяется следующий алгоритм:

- 1) вычислить сумму первых трёх цифр;
- 2) вычислить сумму последних двух цифр;
- 3) записать полученные два числа друг за другом в порядке возрастания (неубывания).

Пример работы алгоритма для числа 56789:

- 1)  $5 + 6 + 7 = 18$ ;
- 2)  $8 + 9 = 17$ ;
- 3) упорядочив, получаем 1718.

Выясним наименьшее и наибольшее пятизначные числа, в результате применения к которым этого алгоритма получается такой же результат.

В старших разрядах наименьшего пятизначного числа должны быть самые маленькие цифры; первая цифра должна быть как можно меньше; сумма первых трёх цифр должна быть меньшей из возможных. В нашем случае это 17:

$$17 = 1 + 7 + 9 \text{ или } 17 = 1 + 8 + 8.$$

Есть единственный вариант, позволяющий получить вторую сумму из двух цифр:  $18 = 9 + 9$ .

Составим наименьшее пятизначное число: 17 999.

В старших разрядах наибольшего пятизначного числа должны быть самые большие цифры; первая цифра должна быть как можно больше; сумма первых трёх цифр должна быть большей из возможных. В нашем случае это 18:  $18 = 9 + 9 + 0$ .

Вторую сумму из двух цифр можно получить только так:  $17 = 9 + 8$ .

Составим наибольшее пятизначное число: 99098.

При разработке алгоритма:

- 1) выделяются фигурирующие в задаче объекты, устанавливаются свойства объектов, отношения между объектами и возможные действия с объектами;
- 2) определяются исходные данные и требуемый результат;
- 3) определяется последовательность действий исполнителя, обеспечивающая переход от исходных данных к результату;
- 4) последовательность действий записывается с помощью команд, входящих в систему команд исполнителя.

Можно сказать, что алгоритм — план управления исполнителем.

По ссылке <http://www.niisi.ru/kumir/> вы можете скачать систему КуМир (Комплект учебных Миров) со встроенными исполнителями Робот, Чертёжник, Водолей и др. Кумир работает в операционных системах Windows и Linux.

### 2.1.3. Свойства алгоритма

Не любая инструкция, последовательность предписаний или план действий может считаться алгоритмом. Каждый алгоритм обязательно обладает следующими свойствами: дискретностью, понятностью, определённой, результативностью и массовостью.

**Свойство дискретности** означает, что путь решения задачи разделён на отдельные шаги (действия). Каждому действию соответствует предписание (команда). Только выполнив одну команду, исполнитель может приступить к выполнению следующей команды.

**Свойство понятности** означает, что алгоритм состоит только из команд, входящих в систему команд исполнителя, т. е. из таких команд, которые исполнитель может воспринять и по которым может выполнить требуемые действия.



**Свойство определённости** означает, что в алгоритме нет команд, смысл которых может быть истолкован исполнителем неоднозначно; недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какую команду выполнять следующей. Благодаря этому результат алгоритма однозначно определяется набором исходных данных: если алгоритм несколько раз применяется к одному и тому же набору исходных данных, то на выходе всегда получается один и тот же результат.

**Свойство результативности** означает, что алгоритм должен обеспечивать получение результата после конечного, возможно, очень большого, числа шагов. При этом результатом считается не только обусловленный постановкой задачи ответ, но и вывод о невозможности продолжения по какой-либо причине решения данной задачи.

**Свойство массовости** означает, что алгоритм должен обеспечивать возможность его применения для решения любой задачи из некоторого класса задач. Например, алгоритм нахождения корней квадратного уравнения должен быть применим к любому квадратному уравнению, алгоритм перехода улицы должен быть применим в любом месте улицы, алгоритм приготовления лекарства должен быть применим для приготовления любого его количества и т. д.

**Пример 9.** Рассмотрим один из методов нахождения всех простых чисел, не превышающих некоторое натуральное число  $n$ . Этот метод называется «решето Эратосфена» по имени предложившего его древнегреческого учёного Эратосфена (III в. до н. э.).

Для нахождения всех простых чисел, не больших заданного числа  $n$ , следуя методу Эратосфена, нужно выполнить следующие шаги:

- 1) выписать подряд все натуральные числа от 2 до  $n$  (2, 3, 4, ...,  $n$ );
- 2) заключить в рамку 2 — первое простое число;
- 3) вычеркнуть из списка все числа, делящиеся на последнее найденное простое число;
- 4) найти первое неотмеченное число (отмеченные числа — зачёркнутые числа или числа, заключённые в рамку) и заключить его в рамку — это будет очередное простое число;
- 5) повторять шаги 3 и 4 до тех пор, пока не останется неотмеченных чисел.





Более наглядное представление о методе нахождения простых чисел вы сможете получить с помощью размещённой в Единой коллекции цифровых образовательных ресурсов анимации «Решето Эратосфена» (180279).

Рассмотренная последовательность действий является алгоритмом, так как она удовлетворяет свойствам:

- дискретности — процесс нахождения простых чисел разбит на шаги;
- понятности — каждая команда понятна ученику 8 класса, выполняющему этот алгоритм;
- определённости — каждая команда трактуется и выполняется исполнителем однозначно; имеются указания об очередности выполнения команд;
- результативности — через некоторое число шагов достигается результат;
- массовости — последовательность действий применима для любого натурального  $n$ .

Рассмотренные свойства алгоритма позволяют дать более точное определение алгоритма.



---

**Алгоритм** — это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами дискретности, понятности, определённости, результативности и массовости.

---

### 2.1.4. Возможность автоматизации деятельности человека

Разработка алгоритма — как правило, трудоёмкая задача, требующая от человека глубоких знаний, изобретательности и больших временных затрат.

Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям.



**Пример 10.** Из кучи, содержащей любое, большее 3, количество каких-либо предметов, двое играющих по очереди берут по одному или по два предмета. Выигрывает тот, кто своим очередным ходом сможет забрать все оставшиеся предметы.



Рассмотрим алгоритм, следуя которому первый игрок наверняка обеспечит себе выигрыш.

1. Если число предметов в куче кратно 3, то уступить ход противнику, иначе начать игру, взяв 1 или 2 предмета так, чтобы осталось количество предметов, кратное 3.
2. Своим очередным ходом каждый раз дополнять число предметов, взятых соперником, до 3 (число оставшихся предметов должно быть кратно 3).

Исполнитель может не вникать в смысл того, что он делает, и не рассуждать, почему он поступает так, а не иначе, т. е. он может действовать формально. Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека. Для этого:

- 1) процесс решения задачи представляется в виде последовательности простейших операций;
- 2) создаётся машина (автоматическое устройство), способная выполнять эти операции в последовательности, заданной в алгоритме;
- 3) человек освобождается от рутинной деятельности, выполнение алгоритма поручается автоматическому устройству.

## САМОЕ ГЛАВНОЕ

Исполнитель — некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Формальный исполнитель одну и ту же команду всегда выполняет одинаково. Для каждого формального исполнителя можно указать: круг решаемых задач, среду, систему команд и режим работы.

Алгоритм — предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами дискретности, понятности, определённости, результативности и массовости.

Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека.



### Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Дополняет ли презентация информацию, содержащуюся в тексте параграфа? Какими слайдами вы могли бы дополнить презентацию?
2. Что называют алгоритмом?
3. Подберите синонимы к слову «предписание».
4. Приведите примеры алгоритмов, изучаемых вами в школе.
5. Кто может быть исполнителем алгоритма?
6. Приведите пример формального исполнителя. Приведите пример, когда человек выступает в роли формального исполнителя.
7. От чего зависит круг решаемых задач исполнителя «компьютер»?
8. Рассмотрите в качестве исполнителя текстовый процессор, имеющийся на вашем компьютере. Охарактеризуйте круг решаемых этим исполнителем задач и его среду.
9. Что такое команда, система команд исполнителя? Какие команды должны быть у робота, выполняющего функции: а) кассира в магазине; б) дворника; в) охранника? Обсудите эти вопросы в группе.
10. Исследуйте один из исполнителей системы КуМир. Охарактеризуйте его назначение, среду, СКИ, возможности ручного и программного управления. Используйте встроенную в систему справочную информацию.
11. Перечислите основные свойства алгоритма.
12. К чему может привести отсутствие какого-либо свойства у алгоритма? Приведите примеры.
13. В чём важность возможности формального исполнения алгоритма?
14. Последовательность чисел строится по следующему алгоритму: первые два числа последовательности принимаются равными 1; каждое следующее число последовательности принимается равным сумме двух предыдущих чисел. Запишите 10 первых членов этой последовательности. Выясните, как называется эта последовательность.
15. Некоторый алгоритм получает из одной цепочки символов новую цепочку следующим образом. Сначала записывается исходная цепочка символов, после неё записывается исход-



ная цепочка символов в обратном порядке, затем записывается буква, следующая в русском алфавите за той буквой, которая в исходной цепочке стояла на последнем месте. Если в исходной цепочке на последнем месте стоит буква «Я», то в качестве следующей буквы записывается буква «А». Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была «ДОМ», то результатом работы алгоритма будет цепочка «ДОММОДН». Дана цепочка символов «КОМ». Сколько букв «О» будет в цепочке символов, которая получится, если применить алгоритм к данной цепочке, а затем ещё раз применить алгоритм к результату его работы?

16. Найдите в сети Интернет анимацию шагов алгоритма Эратосфена. С помощью алгоритма Эратосфена найдите все простые числа, не превышающие 50.

[www](#)

17. Что будет результатом исполнения Черепахой (см. пример 5) алгоритма?



Повтори 8 [Направо 45 Вперёд 45]

18. Запишите алгоритм для исполнителя Вычислитель (см. пример 6), содержащий не более 5 команд:



- а) получения из числа 3 числа 16;
- б) получения из числа 1 числа 25.

19. Система команд исполнителя Конструктор состоит из двух команд, которым присвоены номера:



- 1 — приписать 2
- 2 — разделить на 2

По первой из них к числу приписывается справа 2, по второй число делится на 2. Как будет преобразовано число 8, если исполнитель выполнит алгоритм 22212? Составьте алгоритм в системе команд этого исполнителя, по которому число 1 будет преобразовано в число 16 (в алгоритме должно быть не более 5 команд).

20. В какой клетке должен находиться исполнитель Робот (пример 7), чтобы после выполнения алгоритма 3241 в неё же и вернуться?

21. К пятизначному натуральному числу применяется следующий алгоритм:



- 1) вычислить сумму первых двух цифр;
- 2) вычислить сумму последних трёх цифр;

3) записать полученные два числа друг за другом в порядке возрастания (неубывания).

Выясните наименьшее и наибольшее пятизначные числа, в результате применения к которым этого алгоритма получится число 1215.

22. К четырёхзначному натуральному числу применяется следующий алгоритм:

- 1) вычислить сумму первых двух цифр;
- 2) вычислить сумму последних двух цифр;
- 3) записать полученные два числа друг за другом в порядке возрастания (неубывания).

Выясните, какие из приведённых ниже чисел могут получиться в результате работы этого алгоритма: 2118, 1818, 1718, 1214, 123.

23. Все алгоритмы, которые мы рассматривали до этого, можно считать алгоритмами последовательными. Подумайте сами почему. Вместе с тем, в реальной жизни очень много принципиально иных алгоритмов.

Параллельный алгоритм — алгоритм, который может быть реализован по частям на множестве различных исполнителей с последующим объединением полученных результатов и получением корректного результата.

Приведите 2–3 примера параллельных алгоритмов из окружающего нас мира.

24. Три актёра готовятся к спектаклю. С ними работают два опытных гримера. Каждый актёр должен быть накрашен и причесан. Макияж у каждого актёра продолжается полчаса, а причёсывание — только 10 минут. Спланируйте работу гримеров так, чтобы актёры как можно быстрее подготовились к выходу на сцену. Сколько для этого потребуется времени?

25. Группа из четырёх туристов должна пройти по мосту в темноте. Идти по мосту одновременно могут не более двух туристов. При этом они могут пользоваться только одним фонарём. Перебросить фонарь с одного берега на другой нельзя, поэтому кто-то из них должен вернуться с фонарём.

Аня проходит через мост за 1 минуту, Борис — за 2 минуты, Вася — за 5 минут и Даша — за 10 минут.

Какое наименьшее время требуется туристам, чтобы все они перешли по мосту на другой берег?

## § 2.2

### Способы записи алгоритмов

#### *Ключевые слова:*

- словесное описание
- построчная запись
- блок-схема
- школьный алгоритмический язык
- псевдокод

Существуют различные способы записи алгоритмов. Основными среди них являются:

- словесные (на естественных языках);
- графические;
- на языках программирования.



Теоретические исследования нашего соотечественника Андрея Андреевича Маркова (младшего) (1903–1979), выполненные в середине прошлого века, показали, что в общем случае алгоритмы должны содержать предписания двух видов:

1) предписания, направленные на непосредственное преобразование информации (функциональные операторы);

2) предписания, определяющие дальнейшее направление действий (логические операторы).

Именно эти операторы положены в основу большинства способов записи алгоритмов.

### 2.2.1. Словесные способы записи алгоритма

**Словесное описание.** Самой простой является запись алгоритма в виде набора высказываний на обычном разговорном языке. Словесное описание имеет минимум ограничений и является наименее формализованным. Однако все разговорные языки обладают неоднозначностью. Чтобы избежать двусмысленности, тексты алгоритма приходится делать очень подробными. Алгоритм в словесной форме может оказаться очень объёмным и трудным для восприятия.

**Пример 1.** Словесное описание алгоритма нахождения наибольшего общего делителя (НОД) пары натуральных чисел (алгоритм Евклида).

Чтобы найти НОД двух чисел, составьте таблицу из двух столбцов и назовите столбцы  $X$  и  $Y$ . Запишите первое из заданных чисел в столбец  $X$ , а второе — в столбец  $Y$ . Если данные числа не равны, замените большее из них на результат вычитания из большего числа меньшего. Повторяйте такие замены до тех пор, пока числа не окажутся равными, после чего число из столбца  $X$  считайте искомым результатом.

**Построчная запись.** Это запись на естественном языке, но с соблюдением некоторых дополнительных правил:

- каждое предписание записывается с новой строки;
- предписания (шаги) алгоритма нумеруются;
- исполнение алгоритма происходит в порядке возрастания номеров шагов, начиная с первого (если не встречается никаких специальных указаний).

Кроме слов естественного языка предписания могут содержать математические выражения и формулы.

**Пример 2.** Построчная запись алгоритма Евклида.

1. Обозначить первое из заданных чисел  $X$ , второе обозначить  $Y$ .
2. Если  $X = Y$ , то перейти к п. 8.
3. Если  $X > Y$ , то перейти к п. 4, иначе перейти к п. 6.
4. Заменить  $X$  на  $X - Y$ .
5. Перейти к п. 2.
6. Заменить  $Y$  на  $Y - X$ .

7. Перейти к п. 2.
8. Считать  $X$  искомым результатом.

Построчная запись алгоритма позволяет избежать ряда неопределённостей; её восприятие не требует дополнительных знаний. Вместе с тем использование построчной записи требует от человека большого внимания.

### 2.2.2. Блок-схемы

Наилучшей наглядностью обладают графические способы записи алгоритмов; самый распространённый среди них — блок-схема.

**Блок-схема** представляет собой графическое изображение, дающее представление о порядке работы алгоритма. Здесь предписания изображаются с помощью различных геометрических фигур, а последовательность выполнения шагов указывается с помощью линий, соединяющих эти фигуры. Линии связи *справа налево* и *снизу вверх* изображаются *со стрелками*. Направления линий связи *слева направо* и *сверху вниз* считаются стандартными, эти линии можно изображать *без стрелок*.

Рассмотрим некоторые условные обозначения, применяемые в блок-схемах.

Выполнение алгоритма всегда начинается с блока начала и оканчивается при переходе на блок конца (рис. 2.2, *а*). Из начального блока выходит одна линия связи; в конечный блок входит одна линия связи.

Внутри блока данных (рис. 2.2, *б*) перечисляются величины, значения которых должны быть введены (исходные данные) или выведены (результаты) в данном месте алгоритма. В блок данных входит одна линия связи, и из блока выходит одна линия связи.

В блоке обработки данных (рис. 2.2, *в*) содержится описание тех действий, которые должны быть выполнены при переходе на этот блок (выполнение определённой операции или группы операций, приводящее к изменению значения, формы или размещения информации). В блок обработки данных входит одна линия связи, и из блока выходит одна линия связи.

Проверка условия изображается с помощью блока принятия решения, внутри которого записывается это условие (рис. 2.2, *г*). В блок принятия решения входит одна линия, а выходят две





линии, около которых записываются результаты проверки условия.

Комментарии (рис. 2.2, д) используются для добавления пояснительных записей, делающих блок-схему более понятной.

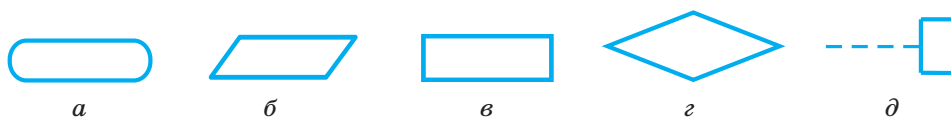


Рис. 2.2. Обозначения на блок-схемах

**Пример 3.** Запись алгоритма Евклида с помощью блок-схемы (рис. 2.3).

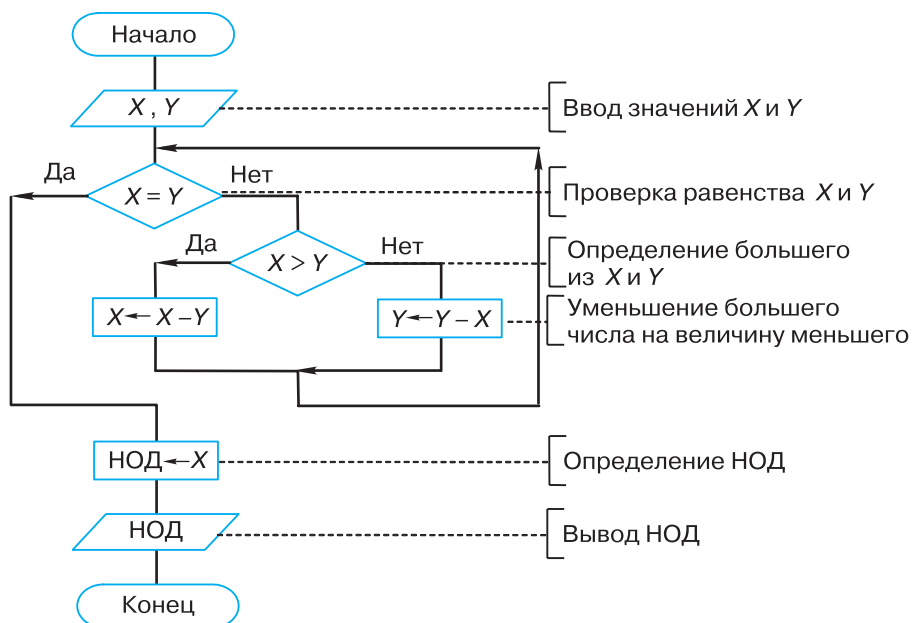


Рис. 2.3. Запись алгоритма Евклида с помощью блок-схемы

Создание детальной блок-схемы сложного алгоритма — трудоёмкая задача. Кроме того, блок-схема, не уместяющаяся на одном стандартном листе, теряет своё основное преимущество — наглядность. При разработке сложных алгоритмов блок-схемы удобно использовать в качестве средства для наглядного представления решения задачи в общем виде.

### 2.2.3. Языки программирования

Языки программирования — формальные языки, предназначенные для записи компьютерных программ, т. е. алгоритмов, исполнителем которых является компьютер. Каждый из них характеризуется:

- **алфавитом** — набором используемых символов;
- **синтаксисом** — системой правил, по которым из символов алфавита образуются правильные конструкции языка;
- **семантикой** — системой правил, строго определяющей смысл и способ употребления конструкций языка.

Языков программирования очень много. На уроках информатики в 8–9 классах мы будем изучать язык программирования **Паскаль (Pascal)**; его основы достаточно подробно изложены в следующей главе учебника. Кроме того, желающие смогут попробовать свои силы в программировании на школьном (учебном) алгоритмическом языке, который также называют русским алгоритмическим языком или алгоритмическим языком КуМир.



**Школьный алгоритмический язык** был введён в употребление академиком А. П. Ершовым в 1985 году.

Андрей Петрович Ершов (1931–1988) — выдающийся советский учёный, инициатор введения курса информатики в школы нашей страны. Его работы оказали огромное влияние на формирование и развитие вычислительной техники во всём мире.

Для записи алгоритмов на школьном алгоритмическом языке используется некоторое ограниченное множество слов, смысл и способ употребления которых заданы раз и навсегда. Это так называемые служебные слова: **алг** (алгоритм), **нач** (начало), **кон** (конец) и др. При записи алгоритмов в книгах служебные слова выделяются жирным шрифтом, в тетради и на доске — подчёркиванием.

В общем виде программу на школьном алгоритмическом языке можно представить так:

```
алг <название алгоритма>  
нач  
  <последовательность команд>  
кон
```

С основными конструкциями школьного алгоритмического языка вы познакомитесь, работая с Роботом, Черепахой, Чертёжником и другими исполнителями, встроенными в систему программирования КуМир.

Эти же конструкции мы будем использовать при записи алгоритмов на псевдокоде — смеси русского и школьного алгоритмического языка.



**Пример 4.** Алгоритм, позволяющий из полного сосуда ёмкостью 12 л отлить половину, пользуясь двумя пустыми сосудами ёмкостью 8 и 5 л.

**алг** переливания

**нач**

наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л  
наполнить сосуд ёмкостью 5 л из сосуда ёмкостью 8 л  
вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л  
вылить всё из сосуда ёмкостью 8 л в сосуд ёмкостью 5 л  
наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л  
долить из сосуда ёмкостью 8 л в сосуд ёмкостью 5 л  
вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л

**кон**

Такая форма записи, ориентированная на исполнителя-человека, помогает понять или изложить сущность того или иного алгоритма. Запись на псевдокоде более формализована, чем другие словесные способы записи алгоритмов. Это позволяет стандартизировать, придать единую форму записи всем алгоритмам, с которыми вы будете иметь дело.

### САМОЕ ГЛАВНОЕ

Существуют различные способы записи алгоритмов: словесное описание, построчная запись, блок-схемы, языки программирования и др. Каждый из этих способов обладает своими достоинствами и недостатками.



### Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Что вы можете сказать о формах представления информации в презентации и в учебнике? Какими слайдами вы могли бы дополнить презентацию?

2. Каковы основные способы записи алгоритмов?
3. Чем вызвано существование многих способов записи алгоритмов?
4. Дайте словесное описание алгоритма сложения двух обыкновенных дробей  $a/b$  и  $c/d$ .
5. Представьте в виде построчной записи алгоритм решения следующей задачи: «Имеются четыре арбуза различной массы. Как, пользуясь чашечными весами без гирь, путём не более пяти взвешиваний расположить арбузы по возрастанию веса?».
6. Представьте с помощью блок-схемы алгоритм решения следующей задачи: «Из трёх монет одинакового достоинства одна фальшивая (более лёгкая). Как её найти с помощью одного взвешивания на чашечных весах без гирь?».
7. Запишите с помощью псевдокода алгоритм построения окружности заданного радиуса  $r$ , проходящей через заданные точки  $A$  и  $B$ .
8. В среде КуМир запишите и выполните алгоритм переливаний (пример 4) для исполнителя Водолей.
9. Сформулируйте основное отличие словесного описания алгоритма от описания на формальном языке.
10. Подготовьте краткую биографическую справку о Маркове А. А. (младшем).



## § 2.3

### Объекты алгоритмов

*Ключевые слова:*

- величина
- константа
- переменная
- тип
- имя
- присваивание
- выражение
- таблица

#### 2.3.1. Величины

Алгоритмы описывают последовательность действий, производимых над некоторыми объектами, определёнными условием задачи. Например, при решении задачи о начислении зарплаты сотрудникам предприятия такими объектами могут быть табельный номер сотрудника, его фамилия, имя, отчество, оклад, отработанное время и т. д.



---

В информатике отдельный информационный объект (число, символ, строка, таблица и др.) называется **величиной**.

---

Величины делятся на постоянные (константы) и переменные. **Постоянной** (константой) называется величина, значение которой указывается в тексте алгоритма и не меняется в процессе его исполнения. **Переменной** называется величина, значение которой может изменяться в процессе исполнения алгоритма. При испол-

нении алгоритма в каждый момент времени переменная обычно имеет значение, называемое текущим значением.

**Пример 1.** Величины, выражающие количество дней в неделе, ускорение свободного падения, количество дней в первой декаде месяца, являются константами. Величины, выражающие количество дней в месяце, пульс человека, количество дней в третьей декаде месяца, являются переменными.

В алгоритмах над величинами выполняются некоторые операции. Например:

- арифметические операции  $+$ ,  $-$ ,  $*$  (умножение),  $/$  (деление);
- операции отношения  $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $=$ ,  $;$ ;
- логические операции И, ИЛИ, НЕ.

Объекты, над которыми выполняются операции, называются операндами. Не всякий объект может быть операндом для выполнения любой операции. Например, текст не может быть объектом для выполнения арифметических операций; отрицательное число не может быть операндом для извлечения квадратного корня и т. д.

Множество величин, объединённых множеством допустимых значений и определённой совокупностью допустимых операций, называют величинами определённого **типа**. При составлении алгоритмов используют величины числового (целого и вещественного), символьного, литерного и логического типов.

В математике и физике оперируют числовыми величинами — натуральными, целыми, действительными числами. При составлении алгоритмов чаще всего используют числовые величины целого и вещественного<sup>1)</sup> типов, которые в школьном алгоритмическом языке обозначаются **цел** и **вещ** соответственно.

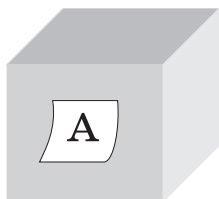
В задачах, возникающих в повседневной жизни, встречаются и нечисловые величины, значениями которых являются символы, слова, тексты и др. При составлении алгоритмов обработки текстовой информации используют величины символьного (**сим**) и литерного (**лит**) типов. Значением символьной величины является один символ: русская или латинская буква, цифра, знак препинания или другой символ. Значением литерной величины является последовательность символов. Иногда эту последовательность называют строкой или цепочкой. Литерные значения в

<sup>1)</sup> Термин «вещественный» принято использовать наряду с термином «действительный».

алгоритме записывают в кавычках, например: 'алгоритм', 'литерная величина', '2017'.

Величины логического (**лог**) типа могут принимать всего два значения:

- ДА (ИСТИНА, TRUE, 1);
- НЕТ (ЛОЖЬ, FALSE, 0).



Для ссылок на величины используют их **имена** (идентификаторы). Имя величины может состоять из одной или нескольких латинских букв, из латинских букв и цифр: *A1*, *M*, *AP*. Рекомендуется выбирать mnemonic имена, т. е. имена, отражающие суть объектов решаемой задачи, например: *SUMMA*, *PLAN*, *CENA* и т. д.

Если величину представить как ящик, содержимым которого является некоторое значение, то имя величины — это ярлык, повешенный на ящик.

### 2.3.2. Выражения

**Выражение** — языковая конструкция для вычисления значения с помощью одного или нескольких операндов.

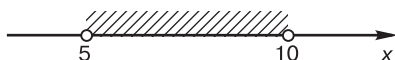
Выражения состоят из операндов (констант, переменных, функций), объединённых знаками операций. Выражения записываются в виде линейных последовательностей символов (без подстрочных и надстрочных символов, обыкновенных дробей и т. д.); знаки операций пропускать нельзя. Порядок выполнения операций определяется скобками и приоритетом (старшинством) операций; операции одинакового приоритета выполняются слева направо.

Различают арифметические, логические и строковые выражения.

**Арифметические выражения** служат для определения числового значения. Например,  $2 * x + 3$  — арифметическое выражение, значение которого при  $x = 1$  равно 5, а при  $x = -1$  — единице. Выражение  $\text{sqrt}(x)$  служит для обозначения операции извлечения квадратного корня из  $x$  ( $\sqrt{x}$ ).

**Логические выражения** описывают некоторые условия, которые могут выполняться или не выполняться. Логическое выражение может принимать одно из двух значений — ИСТИНА

или ЛОЖЬ. Например, логическое выражение  $(x > 5)$  и  $(x < 10)$  определяет принадлежность точки  $x$  интервалу  $(5; 10)$ :



При  $x = 6$  значение этого выражения — ИСТИНА, а при  $x = 12$  — ЛОЖЬ.

**Строковые выражения** состоят из величин (констант, переменных) символьного и литерного типов, соответствующих функций и операций сцепления (присоединения). Операция сцепления обозначается знаком «+» и позволяет соединить в одну последовательность несколько последовательностей символов. Значениями строковых выражений являются последовательности символов. Например, если  $A = \text{'том'}$ , то значение строкового выражения  $\text{'a'} + A$  есть  $\text{'атом'}$ .

### 2.3.3. Команда присваивания

Задать конкретное значение величины можно с помощью операции присваивания, которая записывается так:

$\langle \text{имя переменной} \rangle := \langle \text{выражение} \rangle$

Знак «:=» читается: «присвоить». Например, запись  $A := B + 5$  читается так: «переменной  $A$  присвоить значение выражения  $B$  плюс 5».

Знаки присваивания «:=» и равенства «=» — разные знаки:

- знак «=» означает равенство двух величин, записанных по обе стороны от этого знака;
- знак «:=» предписывает выполнение операции присваивания.

Например, запись  $A := A + 1$  выражает не равенство значений  $A$  и  $A + 1$ , а указание увеличить значение переменной  $A$  на единицу.

При выполнении команды присваивания сначала вычисляется значение выражения, стоящего справа от знака «:=», затем результат присваивается переменной, стоящей слева от знака «:=». При этом тип выражения должен быть совместим с типом соответствующей переменной.

Свойства присваивания:

- 1) пока переменной не присвоено значение, она остаётся неопределённой;
- 2) значение, присвоенное переменной, сохраняется в ней вплоть до выполнения следующего присваивания этой переменной нового значения;





- 3) если мы присваиваем некоторой переменной очередное значение, то предыдущее её значение теряется безвозвратно.



**Пример 2.** Составим алгоритм, в результате которого переменные  $A$  и  $B$  литерного типа обменяются своими значениями.

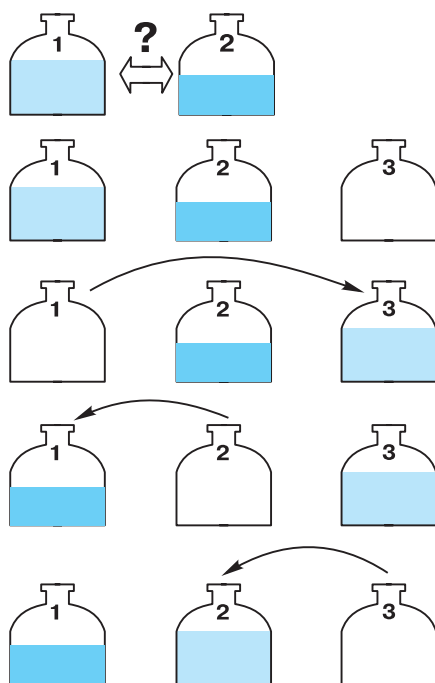
Решение вида

$A := B$

$B := A$

неверно, так как после выполнения первой команды присваивания первоначальное значение переменной  $A$  будет безвозвратно утеряно. Вторая команда присвоит переменной  $B$  текущее значение переменной  $A$ . В результате обе переменные получат одно и то же значение.

Для поиска правильного решения воспользуемся аналогией. Если требуется перелить жидкость из сосуда 1 в сосуд 2, а из сосуда 2 — в сосуд 1, то без дополнительного сосуда 3 здесь не обойтись. Алгоритм переливаний представлен на рис. 2.4.



**Рис. 2.4.** Алгоритм переливаний жидкостей

Для решения исходной задачи введём промежуточную переменную  $M$ . Алгоритм обмена значениями переменных  $A$  и  $B$  запишем так:

**нач**

$M := A$

$A := B$

$B := M$

**кон**

Если  $A$  и  $B$  — числовые величины, то обмен их значениями можно организовать и без промежуточной переменной, например так:

$A := A + B$

$B := A - B$

$A := A - B$

### 2.3.4. Табличные величины

В практической деятельности человек часто использует всевозможные таблицы. Это, например, список учащихся в классном журнале, табель успеваемости, таблица результатов спортивных соревнований и т. д. Чаще всего встречаются линейные и прямоугольные таблицы.

**Линейная таблица (одномерный массив)** представляет собой набор однотипных данных, записанных в одну строку или один столбец. Элементы строки (столбца) всегда нумеруются. Например, с помощью линейной таблицы могут быть представлены дни недели (рис. 2.5, *а*) или количество уроков, пропущенных учеником в течение 5-дневной учебной недели (рис. 2.5, *б*).

1	Понедельник
2	Вторник
3	Среда
4	Четверг
5	Пятница
6	Суббота
7	Воскресенье

*а*

	1	2	3	4	5
Васечкин	6	6	1	0	0

*б*

**Рис. 2.5.** Примеры линейных таблиц

**Прямоугольная таблица (двумерный массив)** — это упорядоченный некоторым образом набор строк (столбцов), содержащих одинаковое количество элементов. Строки прямоугольных таблиц имеют свою нумерацию, столбцы — свою. Например, с помощью прямоугольной таблицы можно представить количество уроков, пропущенных всеми учениками 8 класса в течение 5-дневной учебной недели (рис. 2.6).

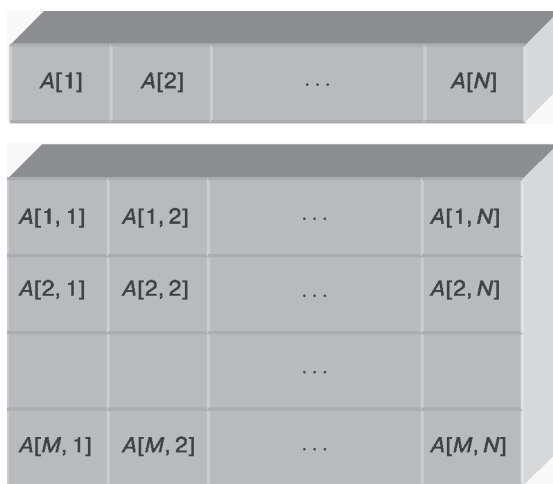
	1	2	3	4	5
1. Васечкин	6	6	1	0	0
2. Ионов	0	0	0	0	6
3. Радугина	0	0	1	0	0
...	...	...	...	...	...
19. Чабанюк	0	0	0	0	0

**Рис. 2.6.** Пример прямоугольной таблицы

Всей совокупности элементов табличной величины даётся одно имя. Элементы различают по их номерам, называемым **индексами**. Индекс записывается в квадратных скобках сразу за именем таблицы.

Если первую из рассмотренных нами таблиц (см. рис. 2.5, *a*) назвать *WEEK*, то  $WEEK[1] = \text{'понедельник'}$ ,  $WEEK[6] = \text{'суббота'}$ . Назовём третью из рассмотренных таблиц *LES*. Тогда  $LES[1, 1] = 6$ ,  $LES[2, 5] = 6$ ,  $LES[3, 4] = 0$ .

Образно линейная и прямоугольная таблицы изображены на рис. 2.7.



**Рис. 2.7.** Образное представление линейной и прямоугольной таблиц

## САМОЕ ГЛАВНОЕ

В информатике отдельный информационный объект (число, символ, строка, таблица и др.) называется величиной.

Величины делятся на постоянные (их значения указываются в тексте алгоритма и не меняются в процессе его исполнения) и переменные (их значения могут изменяться в процессе исполнения алгоритма). При составлении алгоритмов используют величины целого, вещественного, логического, символьного и литерного типов.

Для ссылок на величины используют их имена (идентификаторы). Имя величины может состоять из одной или нескольких латинских букв, из латинских букв и цифр.

Таблица (массив) — набор некоторого числа однотипных элементов, которым присвоено одно имя. Положение элемента в таблице однозначно определяется его индексом (индексами).

## Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Используйте эти материалы при подготовке ответов на вопросы и выполнении заданий.
2. Что такое величина? Чем различаются постоянные и переменные величины?
3. Величины каких типов используются при записи алгоритмов?
4. Укажите тип величины, если её значение равно: 2020; 14.48; 'ДА'; FALSE, -125; '142';  $1,4 \cdot 10^5$ ; .123E-2; 'пять'.
5. Определите типы следующих величин:
  - а) вес человека;
  - б) марка автомобиля;
  - в) год вашего рождения;
  - г) площадь фигуры;
  - д) название месяца года;
  - е) количество мест в самолёте.



6. Работая в группе, приведите примеры допустимых и недопустимых значений для каждой из величин:
  - а) температура человека;
  - б) скорость автомашины;
  - в) площадь страны;
  - г) название дня недели.
7. Для чего предназначена команда присваивания? Каковы её основные свойства?
8. Какая команда присваивания составлена правильно?
  - а)  $A := B$
  - б)  $A = B$
  - в)  $A = B + 1$
  - г)  $A + 1 := A$
9. Придумайте свой алгоритм обмена значениями числовых переменных  $A$  и  $B$ .
10. Имеются числовые переменные  $A$ ,  $B$  и  $C$ . Сколько промежуточных переменных потребуется для того, чтобы переменной  $A$  было присвоено значение переменной  $B$ , переменной  $B$  — значение переменной  $C$ , а переменной  $C$  — значение переменной  $A$ ? Запишите соответствующий алгоритм.
11. После выполнения команды присваивания  $x := x + y$  значение переменной  $x$  равно 3, а значение переменной  $y$  равно 5. Чему были равны значения переменных  $x$  и  $y$  до выполнения указанной команды присваивания?
12. Что называют выражением? Каковы основные правила записи выражений?
13. Переведите из линейной записи в общепринятую:
  - а)  $a * b / c$ ;
  - б)  $a / b * c$ ;
  - в)  $a + b / c$ ;
  - г)  $(a + b) / c$ ;
  - д)  $a + b / c + d$ ;
  - е)  $(a + b) / (c + d)$ .
14. Запишите на школьном алгоритмическом языке:
  - а)  $ax^2 + bx + c$ ;
  - б)  $v_0 + \frac{at^2}{2}$ ;



в)  $\frac{1}{2}(a+b)h$ ;

г)  $\frac{1+x_1x_2}{b^2c}$ ;

д)  $\sqrt{a^2+b^2}$ .

15. Запишите логическое выражение, истинное при выполнении указанного условия и ложное в противном случае:

а)  $x$  принадлежит отрезку  $[0, 1]$ ;

б)  $x$  лежит вне отрезка  $[0, 1]$ ;

в) каждое из чисел  $x, y$  положительно;

г) хотя бы одно из чисел  $x, y$  положительно;

д) ни одно из чисел  $x, y$  не является положительным;

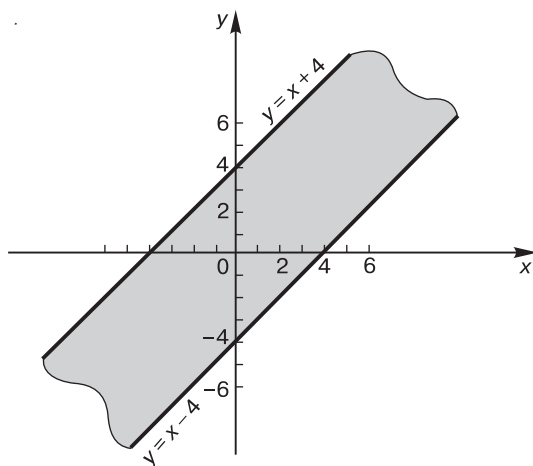
е) только одно из чисел  $x, y$  положительно.

16. Изобразите в декартовой прямоугольной системе координат область, в которой и только в которой истинно следующее логическое выражение:

а)  $(x \geq -1)$  и  $(x \leq 1)$  и  $(y \geq -1)$  и  $(y \leq 1)$ ;

б)  $(y > x)$  и  $(y > -x)$  и  $(y < 1)$ .

17. Запишите логическое выражение, принимающее значение TRUE, когда точка с координатами  $(x, y)$  принадлежит закрашенной области.



18. Запишите команду присваивания, в результате выполнения которой логическая переменная  $t$  получает значение TRUE, если выполняется указанное условие, и значение FALSE в противном случае:

- а)  $x$  — положительное число;
- б) хотя бы одно из чисел  $x$ ,  $y$ ,  $z$  равно нулю;
- в) числа  $x$ ,  $y$ ,  $z$  равны между собой.



19. Какие из приведённых ниже величин целесообразно представлять с помощью таблиц? Обсудите этот вопрос в группе.  
*Величины:* список учеников класса, рост учеников класса, средний рост учеников класса, оценка ученика по физике, средний балл ученика по физике, оценки учеников за контрольную работу по информатике, длины сторон треугольника, длины сторон нескольких треугольников, названия дней недели, имя человека, площадь фигуры, периметры нескольких прямоугольников, самая низкая температура воздуха в январе, количество девочек в классе, самая дождливая декада июня.

## § 2.4

# Основные алгоритмические конструкции

### *Ключевые слова:*

- следование
- ветвление
- повторение
- линейные алгоритмы
- разветвляющиеся алгоритмы
- циклические алгоритмы

Человеку в жизни приходится решать множество различных задач. Решение каждой из них описывается своим алгоритмом, и разнообразие этих алгоритмов очень велико. Вместе с тем для записи любого алгоритма достаточно трёх основных алгоритмических конструкций (структур): следования, ветвления, повторения. Это положение выдвинул и доказал Э. Дейкстра в 70-х гг. прошлого века.



Эдсгер Вибе Дейкстра (1930–2002) — выдающийся нидерландский учёный, идеи которого оказали огромное влияние на развитие компьютерной индустрии.



### 2.4.1. Следование



**Следование** — алгоритмическая конструкция, отображающая естественный, последовательный порядок действий. Алгоритмы, в которых используется только структура «следование», называются **линейными алгоритмами**.

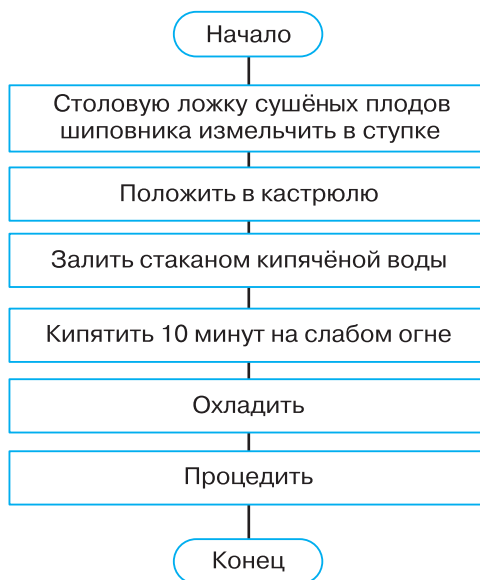
Графическое представление алгоритмической конструкции «следование» приведено на рис. 2.8.



**Рис. 2.8.** Алгоритмическая конструкция «следование»



**Пример 1.** Линейный алгоритм приготовления отвара шиповника.



Обратите внимание, что многие из предписаний этого алгоритма могут потребовать детализации — представления в виде некоторой совокупности более мелких предписаний.

**Пример 2.** У исполнителя Робот есть четыре команды перемещения (вверх, вниз, влево и вправо), при выполнении каждой из них Робот перемещается на одну клетку в соответствующем направлении. По команде закрасить Робот закрашивает клетку, в которой он находится. Запишем линейный алгоритм, исполняя который Робот нарисует на клетчатом поле следующий узор и вернётся в исходное положение, обозначенное звёздочкой:

*		

```

алг узор
нач
    закрасить
    вправо
    вправо
    закрасить
    вниз
    влево
    закрасить
    вверх
    влево
кон
    
```

**Пример 3.** Дан фрагмент линейного алгоритма:

```

x:=2
y:=x*x
y:=y*y
x:=y*x
s:=x+y
    
```

Выясним, какое значение получит переменная *s* после выполнения этого фрагмента алгоритма. Для этого составим таблицу значений переменных, задействованных в алгоритме:

Шаг алгоритма	Переменные		
	<i>x</i>	<i>y</i>	<i>s</i>
1	2	–	–
2	2	4	–
3	2	16	–
4	32	16	–
5	32	16	48



Составленная нами таблица значений переменных моделирует работу исполнителя этого алгоритма.



**Пример 4.** Некоторый исполнитель может выполнять над целыми числами кроме операций сложения, вычитания, умножения и деления ещё две операции: с помощью операции `div` вычисляется целое частное, с помощью операции `mod` — остаток.

Например:  $5 \text{ div } 2 = 2$ ;  $5 \text{ mod } 2 = 1$ ;  $2 \text{ div } 5 = 0$ ;  $2 \text{ mod } 5 = 2$ .

Покажем, как с помощью этих операций можно реализовать алгоритм работы кассира, выдающего покупателю сдачу ( $s$ ) наименьшим количеством банкнот по 1000 ( $k1000$ ), 500 ( $k500$ ), 100 ( $k100$ ) и 50 ( $k50$ ).

```
k1000:=s div 1000
s:=s mod 1000
k500:=s div 500
s:=s mod 500
k100:=s div 100
s:=s mod 100
k50:=s div 50
```



Исполните алгоритм для  $s = 745$  и  $s = 1864$ . Составьте соответствующие таблицы значений переменных.

### 2.4.2. Ветвление



**Ветвление** — алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной из двух последовательностей действий (ветвей). Алгоритмы, в основе которых лежит структура «ветвление», называют **разветвляющимися**.

Блок-схема ветвления представлена на рис. 2.9. Каждая ветвь может быть любой степени сложности (рис. 2.9, а), а может вообще не содержать предписаний (рис. 2.9, б).



**Рис. 2.9.** Структура «ветвление»: *a* — полная форма ветвления; *б* — неполная форма ветвления

На школьном алгоритмическом языке команда ветвления записывается так.

Полная форма ветвления:

```
если <условие>
    то <действия 1>
    иначе <действия 2>
все
```

**Пример 5**

```
алг правописание приставок НЕ, НИ
нач
    если приставка под ударением
        то писать НЕ
        иначе писать НИ
    все
кон
```

Неполная форма ветвления:

```
если <условие>
    то <действия 1>
все
```

**Пример 6**

```
алг сборы на прогулку
нач
    если на улице дождь
        то взять зонтик
    все
кон
```

Для записи условий, в зависимости от результатов проверки которых выбирается та или иная последовательность действий, используются **операции сравнения**:

- $A < B$  —  $A$  меньше  $B$ ;
- $A \leq B$  —  $A$  меньше или равно  $B$ ;
- $A = B$  —  $A$  равно  $B$ ;
- $A > B$  —  $A$  больше  $B$ ;
- $A \geq B$  —  $A$  больше или равно  $B$ ;

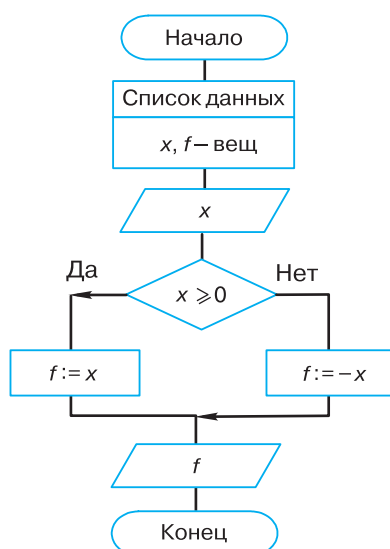


$A \neq B$  —  $A$  не равно  $B$ .

Здесь буквы  $A$  и  $B$  можно заменять на любые переменные, числа и арифметические выражения. Приведённые операции сравнения допускаются и для символьных переменных.



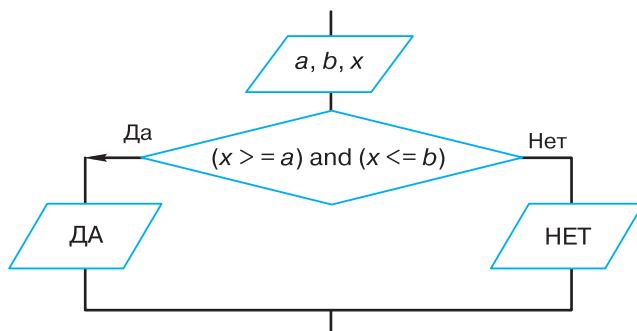
**Пример 7.** Алгоритм вычисления функции  $f(x) = |x|$  для произвольного числа  $x$ .



Обратите внимание на второй блок этой блок-схемы. В нём представлены имена и типы величин (данных), обрабатываемых в алгоритме.

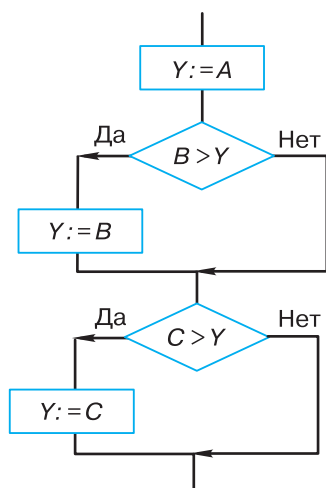
Условия, состоящие из одной операции сравнения, называются **простыми**. В качестве условий при организации ветвлений можно использовать и составные условия. **Составные условия** получаются из простых с помощью логических связок **and** (**и**), **or** (**или**), **not** (**не**): **and** означает одновременное выполнение всех условий, **or** — выполнение хотя бы одного условия, а **not** означает отрицание условия, записанного после слова **not**.

**Пример 8.** Алгоритм определения принадлежности точки  $x$  отрезку  $[a, b]$ . Если точка  $x$  принадлежит данному отрезку, то выводится ответ ДА, в противном случае — НЕТ.



Существует достаточно много ситуаций, в которых приходится выбирать не из двух, а из трёх и более вариантов. Есть разные способы построения соответствующих алгоритмов. Один из них — составить комбинацию из нескольких ветвлений.

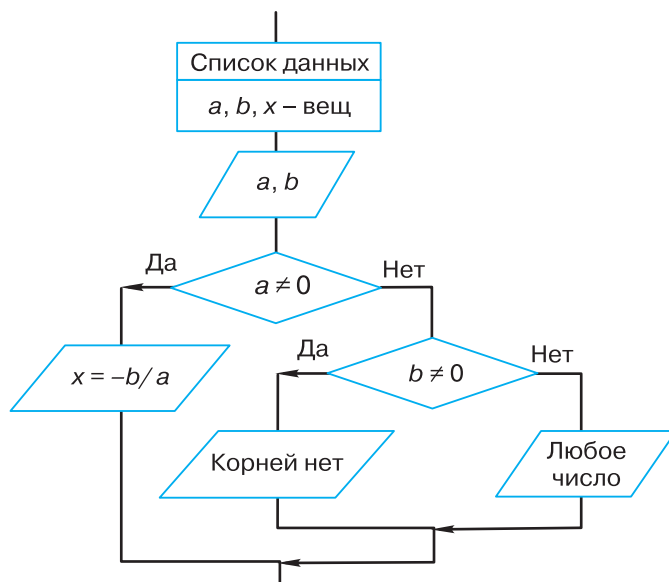
**Пример 9.** Алгоритм, в котором переменной  $Y$  присваивается значение наибольшей из трёх величин  $A$ ,  $B$  и  $C$ .



Пусть  $A = 10$ ,  $B = 30$  и  $C = 20$ . Тогда процесс выполнения алгоритма можно представить в следующей таблице:

Шаг алгоритма	Константы			Переменная	Условие
	A	B	C		
	10	30	20	Y	
1				10	
2					$30 > 10$ (Да)
3				30	
4					$20 > 30$ (Нет)

**Пример 10.** Алгоритм решения линейного уравнения  $ax + b = 0$ .



**Пример 11.** Исполнитель Робот может выполнять ту или иную последовательность действий в зависимости от выполнения следующих простых условий:

справа свободно  
слева свободно  
сверху свободно  
снизу свободно  
клетка чистая

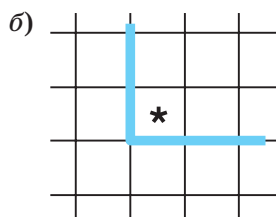
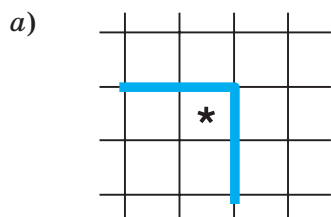
справа стена  
слева стена  
сверху стена  
снизу стена  
клетка закрашена

Также Робот может действовать в зависимости от выполнения составных условий.

Подумайте, в какую клетку переместится Робот из клетки, обозначенной звёздочкой, при выполнении следующего фрагмента алгоритма.

```

если справа свободно или снизу свободно
    то закрасить
все
если справа стена
    то влево
все
если слева стена
    то вправо
все
    
```



### 2.4.3. Повторение

**Повторение** — алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно. Алгоритмы, содержащие конструкцию повторения, называют **циклическими** или **циклами**. Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.

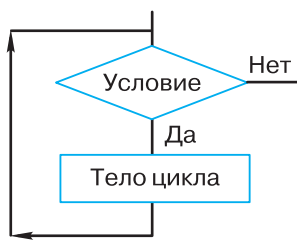
В зависимости от способа организации повторений можно выделить следующие типы циклов:

- 1) цикл с заданным условием продолжения работы;
- 2) цикл с заданным условием окончания работы;
- 3) цикл с фиксированным числом повторений.

**Цикл с заданным условием продолжения работы (цикл-ПОКА, цикл с предусловием)**. Логика работы этой конструкции описывается схемой, показанной на рис. 2.10.







**Рис. 2.10.** Цикл с предусловием

На школьном алгоритмическом языке эта конструкция записывается так:

```
нц пока <условие>  
    <тело цикла (последовательность действий)>  
кц
```

Выполняется цикл-ПОКА следующим образом: 1) проверяется условие (вычисляется значение логического выражения); 2) если условие выполняется (Да), то выполняется тело цикла и снова осуществляется переход к проверке условия; если же условие не выполняется (Нет), то выполнение цикла заканчивается. Возможны случаи, когда тело цикла не будет выполнено ни разу.



**Пример 12.** Алгоритм, по которому из всех имеющихся кирпичей отбираются целые кирпичи и складываются в машину.

```
алг отбор  
нач  
    нц пока есть кирпичи  
        взять один кирпич  
        если кирпич целый  
            то положить кирпич в машину  
            иначе отложить кирпич в сторону  
        все  
    кц  
кон
```

**Пример 13.** Правее Робота (клетка со звёздочкой) расположен коридор неизвестной длины. Необходимо, чтобы Робот закрасил все клетки этого коридора.



Пока будет выполняться условие справа свободно, Роботу следует выполнять команды:

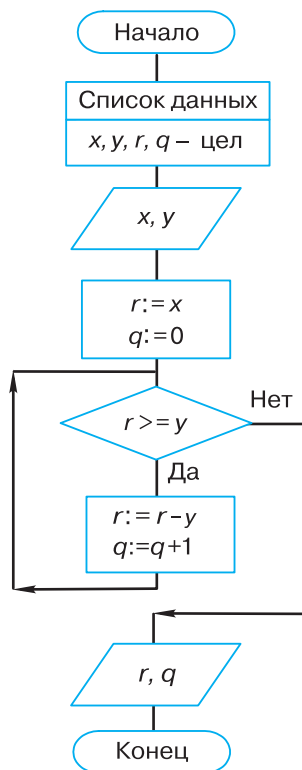
вправо  
закрась

Соответствующий алгоритм для Робота будет иметь вид:

**нц пока** справа свободно  
    вправо  
    закрась  
**кц**

**Пример 14.** Требуется, не пользуясь операцией деления, получить частное  $q$  и остаток  $r$  от деления натурального числа  $x$  на натуральное число  $y$ .

Представим операцию деления как последовательные вычитания делителя из делимого. Причём вычитать будем до тех пор, пока результат вычитания не станет меньше вычитаемого (делителя). В этом случае количество вычитаний будет равно частному от деления  $q$ , а последняя разность — остатку от деления  $r$ . Блок-схему алгоритма см. справа.



## Глава 2. Основы алгоритмизации

Исполним этот алгоритм для  $x = 23$  и  $y = 5$ .

Шаг алгоритма	Операция	Переменная				Условие $r \geq y$
		$x$	$y$	$r$	$q$	
1	ввод $x$	23	–	–	–	
2	ввод $y$		5	–	–	
3	$r := x$			23	–	
4	$q := 0$				0	
5	$r \geq y$					23 $\geq$ 5 (Да)
6	$r := r - y$			18		
7	$q := q + 1$				1	
8	$r \geq y$					18 $\geq$ 5 (Да)
9	$r := r - y$			13		
10	$q := q + 1$				2	
11	$r \geq y$					13 $\geq$ 5 (Да)
12	$r := r - y$			8		
13	$q := q + 1$				3	
14	$r \geq y$					8 $\geq$ 5 (Да)
15	$r := r - y$			3		
16	$q := q + 1$				4	
17	$r \geq y$					3 $\geq$ 5 (Нет)
18	вывод $r$			3		
19	вывод $q$				4	

Цикл с заданным условием окончания работы (цикл-ДО, цикл с постусловием). Логика работы этой конструкции описывается схемой, показанной на рис. 2.11.

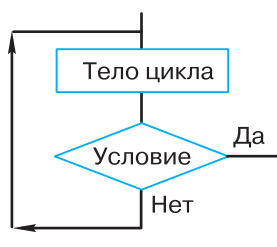


Рис. 2.11. Цикл с постусловием

На школьном алгоритмическом языке эта конструкция записывается так:

```

нц
  <тело_цикла (последовательность действий)>
кц при <условие>
  
```

Выполняется цикл-ДО следующим образом: 1) выполняется тело цикла; 2) проверяется условие (вычисляется значение логического выражения); если условие не выполняется («Нет»), то снова выполняется тело цикла и осуществляется переход к проверке условия; если же условие выполняется, то выполнение цикла заканчивается. В любом случае тело цикла будет выполнено хотя бы один раз.

**Пример 15.** Алгоритм по выучиванию наизусть четверостишия.

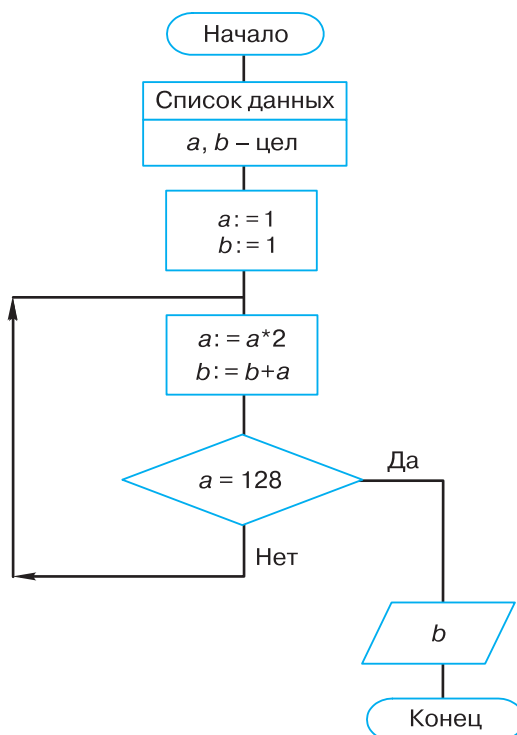
```

алг четверостишие
нач
  нц
    прочитать четверостишие по книге 1 раз
    рассказать четверостишие
  кц при не сделал ошибку
кон
  
```





**Пример 16.** Вычислим значение переменной  $b$  согласно следующему алгоритму:



Составим таблицу значений переменных, задействованных в алгоритме:

Шаг алгоритма	Операция	Переменные		Условие
		$a$	$b$	$a = 128$
1	$a := 1$	1	–	
2	$b := 1$		1	
3	$a := a * 2$	2		
4	$b := b + a$		3	
5	$a = 128$			$2 = 128$ (Нет)
6	$a := a * 2$	4		

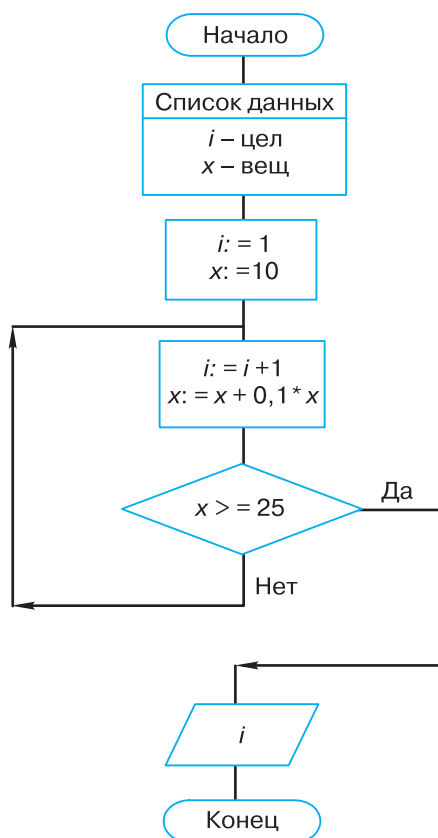
Шаг алгоритма	Операция	Переменные		Условие
		<i>a</i>	<i>b</i>	$a = 128$
7	$b := b + a$		7	
8	$a = 128$			$4 = 128$ (Нет)
9	$a := a * 2$	8		
10	$b := b + a$		15	
11	$a = 128$			$8 = 128$ (Нет)
12	$a := a * 2$	16		
13	$b := b + a$		31	
14	$a = 128$			$16 = 128$ (Нет)
15	$a := a * 2$	32		
16	$b := b + a$		63	
17	$a = 128$			$32 = 128$ (Нет)
18	$a := a * 2$	64		
19	$b := b + a$		127	
20	$a = 128$			$64 = 128$ (Нет)
21	$a := a * 2$	128		
22	$b := b + a$		255	
23	$a = 128$			$128 = 128$ (Да)
24	вывод $b$		255	

Ответ:  $b = 255$ .

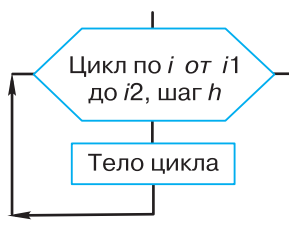
**Пример 17.** Спортсмен приступает к тренировкам по следующему графику: в первый день он должен пробежать 10 км; каждый следующий день следует увеличивать дистанцию на 10% от нормы предыдущего дня. Как только дневная норма достигнет или превысит 25 км, необходимо прекратить её увеличение и далее пробегать ежедневно ровно 25 км. Начиная с какого дня спортсмен будет пробегать 25 км?

Пусть  $x$  — количество километров, которое спортсмен пробежит в некоторый  $i$ -й день. Тогда в следующий  $(i + 1)$ -й день он пробежит  $x + 0,1x$  километров ( $0,1x$  — это 10% от  $x$ ).





Цикл с фиксированным числом повторений (цикл-ДЛЯ, цикл с параметром). Логика работы этой конструкции описывается схемой, показанной на рис. 2.12.



**Рис. 2.12.** Цикл с параметром

На школьном алгоритмическом языке эта конструкция записывается так:

```

нц для i от i1 до i2 шаг h
    <тело цикла (последовательность действий)>
кц
    
```

В цикле-ДЛЯ всегда есть **параметр цикла** — величина целого типа, изменяющаяся в ходе выполнения цикла от своего начального значения  $i1$  до конечного значения  $i2$  с шагом  $h$ .

Выполняется цикл-ДЛЯ следующим образом: 1) параметру цикла присваивается начальное значение; 2) параметр цикла сравнивается с конечным значением; если он не превышает конечного значения, то выполняется тело цикла, увеличивается значение параметра цикла на шаг и снова осуществляется проверка параметра цикла; если же параметр цикла превышает конечное значение, то выполнение цикла заканчивается.

Если величина шага в цикле с параметром равна единице, то шаг не указывают. Мы ограничимся рассмотрением именно таких циклов.

В отличие от двух предыдущих конструкций (цикл-ПОКА, цикл-ДО) цикл-ДЛЯ имеет строго фиксированное число повторений, что позволяет избежать *заикливания*, т. е. ситуации, когда тело цикла выполняется бесконечно.

**Пример 18.** Алгоритм переправы через реку воинского отряда из пяти человек. Солдаты могут воспользоваться помощью двух мальчиков — хозяев небольшой лодки, в которой может переправиться или один солдат, или один либо два мальчика.

**алг** переправа

**нач**

**нц** для  $i$  от 1 до 5

два мальчика переправляются на противоположный берег  
 один мальчик высаживается на берег, другой плывёт обратно  
 солдат переправляется через реку  
 мальчик возвращается на исходную позицию

**кц**

**кон**

**Пример 19.** Составим алгоритм вычисления степени с натуральным показателем  $n$  для любого вещественного числа  $a$ .

По определению:

$$a^1 = a, \quad a^n = \underbrace{a \cdot a \cdot \dots \cdot a}_n, \quad a \in R, \quad n \in N, \quad n \geq 2.$$

$n$  сомножителей

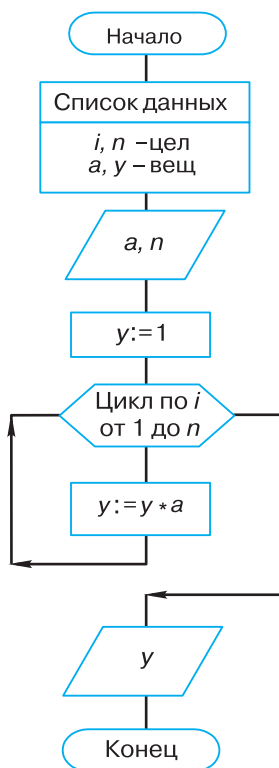
При составлении алгоритма воспользуемся единой формулой, в которой число умножений равно показателю степени:

$$a^n = \underbrace{1 \cdot a \cdot a \cdot \dots \cdot a}_n$$

$n$  умножений







Исполним этот алгоритм для  $a = 4$  и  $n = 3$ .

Шаг алгоритма	Операция	Переменные				Условие
		$a$	$n$	$y$	$i$	
1	ввод $a, n$	4	3	–	–	
2	$y:=1$			1	–	
3	$i:=1$				1	
4	$i \leq n$					$1 \leq 3$ (Да)
5	$y:=y*a$			4		
6	$i:=i+1$				2	
7	$i \leq n$					$2 \leq 3$ (Да)
8	$y:=y*a$			16		
9	$i:=i+1$				3	
10	$i \leq n$					$3 \leq 3$ (Да)
11	$y:=y*a$			64		

Шаг алгоритма	Операция	Переменные				Условие
		<i>a</i>	<i>n</i>	<i>y</i>	<i>i</i>	
12	$i := i + 1$				4	
13	$i \leq n$					$4 \leq 3$ (Нет)
14	вывод $y$			64		

**Пример 20.** Для исполнителя Робот цикл с фиксированным числом повторений реализуется с помощью следующей конструкции:

```

нц <число повторений> раз
    <тело цикла>
кц
    
```

Так, если правее Робота не встретится препятствий, то, выполнив приведённый ниже алгоритм, он переместится на пять клеток вправо и закрасит эти клетки:

```

алг
нач
    нц 5 раз
        вправо; закрасить
    кц
кон
    
```

**Пример 21.** Исполнитель Чертёжник предназначен для построения рисунков на координатной плоскости. Исходное положение исполнителя — начало координат. Система команд Чертёжника:

Команда	Действие
поднять перо	Чертёжник поднимает перо
опустить перо	Чертёжник опускает перо
сместиться на вектор $(a, b)$	Чертёжник из текущей точки с координатами $(x, y)$ перемещается в точку с координатами $(x + a, y + b)$ . Если числа $a, b$ положительные, то значение соответствующей координаты увеличивается; если отрицательные, то уменьшается
<b>нц</b> <число повторений> <b>раз</b> <тело цикла> <b>кц</b>	Команды, образующие тело цикла, повторяются указанное число раз



Пусть Чертёжник находится в точке с координатами  $(x_0, y_0)$ . Выясним, в какой точке он окажется после выполнения программы.

```

нц k раз
    сместиться на вектор (a, b)
кц
    
```

На основании данных о начальном положении Чертёжника и алгоритма его перемещений можно записать:

$x:=x_0$ $y:=y_0$	Исходное положение
<b>нц</b> k <b>раз</b> $x:=x+a$ $y:=y+b$ <b>кц</b>	Преобразование координат в теле цикла

Работу этого алгоритма можно представить так:

$k$	$x$	$y$
—	$x_0$	$y_0$
1	$x_0 + a = x_0 + 1 \cdot a$	$y_0 + b = y_0 + 1 \cdot b$
2	$x_0 + a + a = x_0 + 2 \cdot a$	$y_0 + b + a = y_0 + 2 \cdot b$
3	$x_0 + a + a + a = x_0 + 3 \cdot a$	$y_0 + b + a + a = y_0 + 3 \cdot b$
	...	...
$k$	$x_0 + a + a + \dots + a = x_0 + k \cdot a$	$y_0 + b + b + \dots + b = y_0 + k \cdot b$

Из таблицы видно, что вместо имеющегося циклического алгоритма можно использовать линейный алгоритм, приводящий к такому же результату:

```

 $x:=x_0+k \cdot a$ 
 $y:=y_0+k \cdot b$ 
    
```

Таким образом, Чертёжник, находившийся в точке с координатами  $(x_0, y_0)$ , после выполнения программы:

```

нц k раз
    сместиться на вектор (a, b)
кц
    
```

окажется в точке с координатами  $(x_0 + k \cdot a, y_0 + k \cdot b)$ .

Пусть Чертёжник находится в точке с координатами (1, 2). Выясним, в какой точке он окажется после выполнения программы:

```

нц 5 раз
    сместиться на вектор (3,3)
    сместиться на вектор (1,-1)
кц
    
```

С учётом исходных данных и алгоритма перемещений исполнителя запишем:

<pre> x:=1 y:=2                 </pre>	Исходное положение
<pre> <b>нц</b> 5 <b>раз</b>     x:=x+3; y:=y+3     x:=x+1; y:=y-1 <b>кц</b>                 </pre>	Преобразование координат в теле цикла

В теле цикла изменение координат точки происходит дважды. Выполнив соответствующие вычисления, тело цикла перепишем так:

```

нц 5 раз
    x:=x+4; y:=y+2
кц
    
```

Заменим циклический алгоритм линейным:

```

x:=1+5*4
y:=2+5*2
    
```

Таким образом, Чертёжник после выполнения программы окажется в точке с координатами (21, 12).

Подумайте, какой командой можно дополнить исходную программу, чтобы после выполнения программы Чертёжник:

- а) оказался в начале координат;
- б) вернулся в исходную точку.

### САМОЕ ГЛАВНОЕ

Для записи любого алгоритма достаточно трёх основных алгоритмических конструкций (структур): следования, ветвления, повторения.

Следование — алгоритмическая конструкция, отображающая естественный, последовательный порядок действий. Алгоритмы, в которых используется только структура «следование», называются линейными.



Ветвление — алгоритмическая конструкция, в которой в зависимости от результата проверки условия («Да» или «Нет») предусмотрен выбор одной из двух последовательностей действий (ветвей). Алгоритмы, в основе которых лежит структура «ветвление», называют разветвляющимися.

Повторение — алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно. Алгоритмы, содержащие конструкцию «повторение», называют циклическими или циклами. Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется телом цикла. В зависимости от способа организации повторений различают три типа циклов:

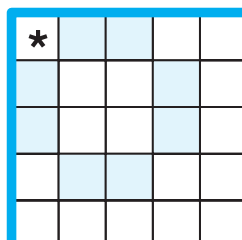
- 1) цикл с заданным условием продолжения работы;
- 2) цикл с заданным условием окончания работы;
- 3) цикл с фиксированным числом повторений.



### Вопросы и задания



1. Ознакомьтесь с материалами презентации к параграфу, содержащейся в электронном приложении к учебнику. Используйте эти материалы при подготовке ответов на вопросы и выполнении заданий.
2. Какие алгоритмы называются линейными? Зависит ли в линейном алгоритме последовательность выполняемых действий от исходных данных?
3. Приведите пример линейного алгоритма:
  - а) из повседневной жизни;
  - б) из литературного произведения;
  - в) из любой предметной области, изучаемой в школе.
4. Запишите линейный алгоритм, исполняя который Робот нарисует на клетчатом поле следующий узор и вернётся в исходное положение:



5. По алгоритму восстановите формулу.

```
a1:=1/x
a2:=a1/x
a3:=a2/x
a4:=a3/x
y:=a1+a2
y:=y+a3
y:=y+a4
```

6. Какое значение получит переменная  $y$  после выполнения алгоритма?

```
x:=1
y:=2*x
y:=y+3
y:=y*x
y:=y+4
y:=y*x
y:=y+5
```

Восстановите формулу вычисления  $y$  для произвольного значения  $x$ .

7. Для заданного количества суток ( $t$  в  $fh$ ) требуется определить количество часов ( $h$ ), минут ( $m$ ) и секунд ( $s$ ).
8. Известно, что 1 миля = 7 вёрст, 1 верста = 500 сажень, 1 сажень = 3 аршина, 1 аршин = 28 дюймов, 1 дюйм = 25,4 мм. Пользуясь этой информацией, составьте линейный алгоритм перевода расстояния  $X$  миль в километры.
9. Исходное данное — целое трёхзначное число  $x$ . Выполните для  $x = 125$  следующий алгоритм.

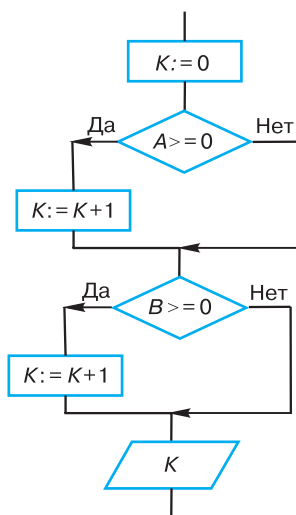
```
a:=x div 100
b:=x mod 100 div 10
c:=x mod 10
s:=a+b+c
```

Какой смысл имеет результат  $s$  этого алгоритма?

10. Определите значения целочисленных переменных  $x$  и  $y$  после выполнения алгоритма.

```
x:=336
y:=8
x:=x div y
y:=x mod y
```

11. Какие алгоритмы называют разветвляющимися? Согласны ли вы с утверждением, что в разветвляющемся алгоритме при любых исходных данных выполняются все действия, предусмотренные алгоритмом?
12. Приведите пример разветвляющегося алгоритма:
  - а) из повседневной жизни;
  - б) из литературного произведения;
  - в) из любой предметной области, изучаемой в школе.
13. Дополните алгоритм из примера 9 так, чтобы с его помощью можно было найти наибольшую из четырёх величин  $A$ ,  $B$ ,  $C$  и  $D$ .
14. Составьте алгоритм, с помощью которого можно определить, существует ли треугольник с длинами сторон  $a$ ,  $b$ ,  $c$ .
15. Составьте алгоритм, с помощью которого можно определить, является ли треугольник с заданными длинами сторон  $a$ ,  $b$ ,  $c$  равносторонним.
16. Составьте алгоритм возведения чётного числа в квадрат, а нечётного — в куб.
17. Какая задача решается с помощью следующего алгоритма?

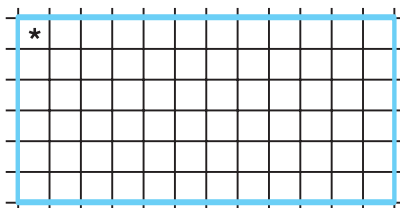


18. Составьте блок-схему алгоритма определения количества чётных чисел среди заданных целых чисел  $A$ ,  $B$  и  $C$ .
19. Составьте блок-схему алгоритма определения принадлежности точки  $x$  отрезку  $[a, b]$  (пример 8) с использованием комбинации из двух ветвлений.

20. Составьте блок-схему алгоритма правописания приставок, оканчивающихся на букву «з».
21. Известно, что 31 января 2011 года было понедельником. Какие значения должны быть присвоены литерной переменной  $y$  в алгоритме, определяющем день недели для произвольного числа (*chislo*) января 2011 года?

```
chislo:= chislo mod 7
если chislo=3 то y:='...'
если chislo=4 то y:='...'
если chislo=5 то y:='...'
если chislo=6 то y:='...'
если chislo=0 то y:='...'
если chislo=1 то y:='...'
если chislo=2 то y:='...'
```

22. Даны две точки на плоскости. Определите, какая из них находится ближе к началу координат.
23. Определите, есть ли среди цифр заданного целого трёхзначного числа одинаковые.
24. Приведите пример циклического алгоритма:  
 а) из повседневной жизни;  
 б) из литературного произведения;  
 в) из любой предметной области, изучаемой в школе.
25. Напишите алгоритм, под управлением которого Робот обойдёт прямоугольную область, обнесённую стеной, по периметру и закрасит угловые клетки. Размеры области неизвестны.

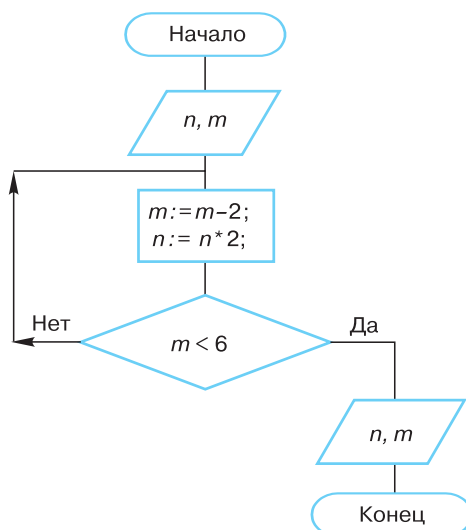


26. Запас рыбы в пруду оценён в  $A$  тонн. Ежегодный прирост рыбы составляет 15%. Ежегодный план отлова —  $B$  тонн. Наименьший запас рыбы составляет  $C$  тонн. (Запас ниже  $C$  тонн уже не восстанавливается.) Составьте блок-схему алгоритма для подсчёта количества лет, в течение которых можно выдерживать заданный план.





27. Дана последовательность 5, 9, 13, 17, ... . Составьте блок-схему алгоритма для определения числа слагаемых (первых членов последовательности), сумма которых равна 324.
28. Составьте алгоритм для определения количества цифр в записи произвольного натурального числа.
29. Сумма 10 000 рублей положена в банк, при этом прирост составляет 5% годовых. Составьте алгоритм, определяющий, через какой промежуток времени первоначальная сумма увеличится в два раза.
30. Одноклеточная амёба каждые три часа делится на 2 клетки. Составьте алгоритм вычисления времени, через которое будет  $X$  амёб.
31. Определите значения переменных  $n$  и  $m$  после выполнения алгоритма.



32. Исполнитель Чертёжник находится в произвольной точке координатной плоскости.

а) Выясните, где окажется Чертёжник после выполнения следующего алгоритма:

**алг**

**нач**

**нц** 6 **раз**

    сместиться на вектор (4, 6)

    сместиться на вектор (-2, -4)

**кц**

    сместиться на вектор (-12, -12)

**кон**

- б) После выполнения следующего алгоритма Чертёжник вернулся в исходную точку:

```

алг
нач
  нц 7 раз
    сместиться на вектор (0, 2)
    сместиться на вектор (a, 0)
    сместиться на вектор (0, b)
    сместиться на вектор (-1, 0)
  кц
  сместиться на вектор (-7, -7)
кон
    
```

Какие числа надо записать вместо  $a$  и  $b$ ?

33. Составьте алгоритм нахождения произведения  $z$  двух натуральных чисел  $x$  и  $y$  без использования операции умножения.
34. Население города  $N$  увеличивается на 5% ежегодно. В текущем году оно составляет 40 000 человек. Составьте блок-схему алгоритма вычисления предполагаемой численности населения города через 3 года. Составьте таблицу значений переменных, задействованных в алгоритме.
35. Каждая бактерия делится на две в течение 1 минуты. В начальный момент имеется одна бактерия. Составьте блок-схему алгоритма вычисления количества бактерий через 10 минут. Исполните алгоритм, фиксируя каждый его шаг в таблице значений переменных.
36. Согласны ли вы со следующими утверждениями:
- короткие алгоритмы могут описывать длинные последовательности действий;
  - краткость алгоритма и скорость его выполнения совпадают?
- Обсудите эти вопросы в группе. Приведите примеры, подтверждающие вашу точку зрения.





## Тестовые задания для самоконтроля

1. Алгоритмом можно считать:
  - а) описание процесса решения квадратного уравнения
  - б) расписание уроков в школе
  - в) технический паспорт автомобиля
  - г) список класса в журнале
2. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач?
  - а) понятность
  - б) определённость
  - в) результативность
  - г) массовость
3. Как называется свойство алгоритма, означающее, что он всегда приводит к результату через конечное, возможно, очень большое, число шагов?
  - а) дискретность
  - б) понятность
  - в) результативность
  - г) массовость
4. Как называется свойство алгоритма, означающее, что он задан с помощью таких предписаний, которые исполнитель может воспринимать и по которым может выполнять требуемые действия?
  - а) дискретность
  - б) понятность
  - в) определённость
  - г) массовость
5. Как называется свойство алгоритма, означающее, что путь решения задачи разделён на отдельные шаги?
  - а) дискретность
  - б) определённость
  - в) результативность
  - г) массовость

6. Как называется свойство алгоритма, означающее, что путь решения задачи определён вполне однозначно, на любом шаге не допускаются никакие двусмысленности и недомолвки?
- а) дискретность
  - б) понятность
  - в) определённость
  - г) результативность

7. Исполнителю Черепаха был дан для исполнения следующий алгоритм:

Повтори 10 [Вперед 10 Направо 72]

Какая фигура появится на экране?

- а) незамкнутая ломаная линия
- б) правильный десятиугольник
- в) фигура, внутренние углы которой равны  $72^\circ$
- г) правильный пятиугольник

8. Исполнитель Робот передвигается по клетчатому полю, выполняя команды, которым присвоены номера: 1 — на клетку вверх, 2 — на клетку вниз, 3 — на клетку вправо, 4 — на клетку влево. Между соседними клетками поля могут стоять стены. Если при выполнении очередного шага Робот сталкивается со стеной, то он разрушается. В результате выполнения программы 3242332411 Робот успешно прошёл из точки А в точку В. Какую программу необходимо выполнить, чтобы Робот вернулся из точки В в точку А по кратчайшему пути и не подвергнулся риску разрушения?

- а) 41
  - б) 4131441322
  - в) 2231441314
  - г) 241314
  - д) 14
9. Система команд исполнителя Вычислитель состоит из двух команд, которым присвоены номера:

1 — вычти 2

2 — умножь на 3

Первая из них уменьшает число на 2, вторая увеличивает число в 3 раза. При записи алгоритмов для краткости указываются лишь номера команд. Запишите алгоритм, содержащий не более пяти команд, с помощью которого из числа 11 будет получено число 13.



10. Некоторый алгоритм строит цепочки символов следующим образом:

- первая цепочка состоит из одного символа — цифры 1;
- в начало каждой из последующих цепочек записывается число — номер строки по порядку, далее дважды подряд записывается предыдущая строка.

Вот первые 3 строки, созданные по этому правилу:

- (1) 1
- (2) 211
- (3) 3211211

Сколько символов будет в седьмой цепочке, созданной по этому алгоритму?

11. К четырёхзначному натуральному числу применяется следующий алгоритм:

- 1) вычислить сумму первых двух цифр;
- 2) вычислить сумму последних двух цифр;
- 3) записать полученные два числа друг за другом в порядке убывания (невозрастания).

Укажите число, которое может получиться в результате работы этого алгоритма:

- а) 1918
- б) 218
- в) 1212
- г) 1218

12. Наибольшей наглядностью обладает следующая форма записи алгоритмов:

- а) словесная
- б) рекурсивная
- в) графическая
- г) построчная

13. Величины, значения которых могут изменяться в процессе исполнения алгоритма, называются:

- а) постоянными
- б) константами
- в) переменными
- г) табличными

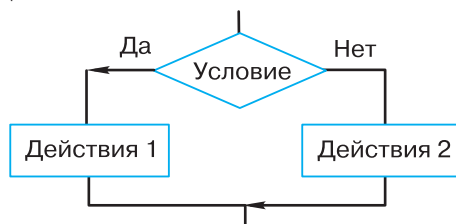
14. Величиной целого типа является:

- а) количество мест в зрительном зале
- б) рост человека
- в) марка автомобиля
- г) площадь государства

15. Укажите логическое выражение, определяющее принадлежность точки  $x$  отрезку  $[-10, 10]$ .
- $(x > 10)$  и  $(x < -10)$
  - $(x > 10)$  или  $(x < -10)$
  - $(x < 10)$  или  $(x \geq -10)$
  - $(x \geq -10)$  и  $(x \leq 10)$
16. Укажите правильный вариант записи условия « $x$  — двузначное число»:
- $x \text{ div } 10 \leq 9$
  - $(x \geq 10)$  и  $(x < 100)$
  - $x \text{ div } 100 = 0$
  - $x \text{ mod } 100 = 99$
17. Какая команда присваивания должна следовать за командами  $A := A + B$  и  $B := A - B$ , чтобы последовательное выполнение всех трёх команд вело к обмену значениями переменных  $A$  и  $B$ ?
- $A := A + B$
  - $A := A - B$
  - $B := A + B$
  - $B := B - A$
18. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?



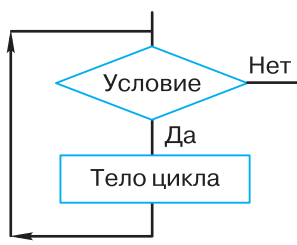
- линейный
  - разветвляющийся
  - циклический
  - вспомогательный
19. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?



- а) линейный
- б) разветвляющийся с неполным ветвлением
- в) разветвляющийся с полным ветвлением
- г) циклический

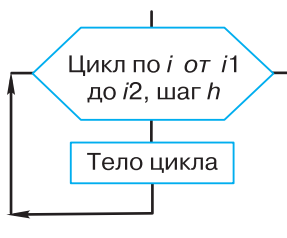
20. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?

- а) цикл с параметром
- б) цикл с заданным условием продолжения работы
- в) цикл с заданным условием окончания работы
- г) цикл с фиксированным числом повторений



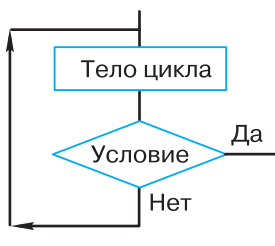
21. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?

- а) цикл с заданным условием продолжения работы
- б) цикл с заданным условием окончания работы
- в) цикл с постусловием
- г) цикл с фиксированным числом повторений



22. К какому виду алгоритмов можно отнести алгоритм, схема которого представлена ниже?

- а) цикл с заданным условием продолжения работы
- б) цикл с заданным условием окончания работы
- в) цикл с фиксированным числом повторений
- г) цикл с предусловием



23. Сергей, Антон, Таня и Надя, гуляя по лесу, наткнулись на овраг, который можно перейти по шаткому мосту. Сергей может перейти его за минуту, Антон — за две, Таня — за три, Надя — за четыре. Фонарик у группы только один, и он обязательно нужен для перехода по мосту, который выдерживает только двоих человек. Когда два человека вместе идут по мосту, то идут они со скоростью более медленного из них. Ребята смогли разработать алгоритм перехода на другую сторону оврага за минимально возможное время. Какое время она затратили на его исполнение?

- а) 10 минут
- б) 11 минут
- в) 12 минут
- г) 13 минут

24. Дан фрагмент линейного алгоритма.

a:=8  
b:=6+3\*a  
a:=b/3\*a

Чему равно значение переменной  $a$  после его исполнения?

25. Исполните следующий фрагмент линейного алгоритм для  $a = x, b = y$ .

a:=a+b  
b:=b-a  
a:=a+b  
b:=-b

Какие значения присвоены переменным  $a$  и  $b$ ?

- а)  $y, x$
- б)  $x + y, x - y$
- в)  $x, y$
- г)  $-y, x$

26. Определите значения целочисленных переменных  $x$  и  $y$  после выполнения фрагмента алгоритма.





```

x:=11
y:=5
t:=y
y:=x mod y
x:=t
y:=y+2*t
    
```

а)  $x = 11, y = 5$

б)  $x = 5, y = 11$

в)  $x = 10, y = 5$

г)  $x = 5, y = 10$

27. Среди четырёх монет есть одна фальшивая. Неизвестно, легче она или тяжелее настоящей. Какое минимальное количество взвешиваний необходимо сделать на весах с двумя чашками без гирь, чтобы определить фальшивую монету?

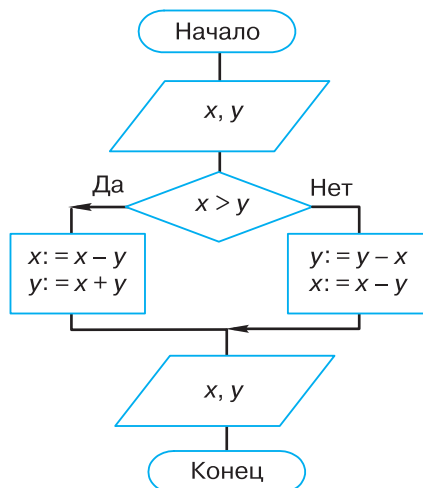
а) 2

б) 3

в) 4

г) 5

28. Исполните алгоритм при  $x = 10$  и  $y = 15$ .



Какие значения будут получены в результате его работы?

а)  $-5, 10$

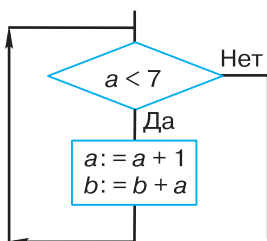
б)  $5, 20$

в)  $10, 15$

г)  $5, 5$

д)  $-5, 5$

29. Исполните фрагмент алгоритма при  $a = 2$  и  $b = 0$ .



Определите значение переменной  $b$  после выполнения фрагмента алгоритма.

30. Определите значение переменной  $f$  после выполнения фрагмента алгоритма.

```

f:=1
нц для i от 1 до 5
  f:=f*i
кц
  
```

31. Определите значение переменной  $s$  после выполнения фрагмента алгоритма.

```

s:=0
нц для i от 1 до 5
  s:=s+i*i
кц
  
```

32. Исполнитель Чертёжник выполнил следующий алгоритм:

```

нц 5 раз
  сместиться на вектор (0, 2)
  сместиться на вектор (4, 0)
кц
  
```

На какую одну команду можно заменить этот алгоритм, чтобы Чертёжник оказался в той же точке, что и после его выполнения?

- а) сместиться на вектор  $(-4, 0)$
- б) сместиться на вектор  $(0, -2)$
- в) сместиться на вектор  $(20, 10)$
- г) сместиться на вектор  $(-20, -10)$

Для проверки знаний и умений по теме «Основы алгоритмизации» вы можете воспользоваться интерактивным тестом к главе 2, содержащимся в электронном приложении к учебнику.