МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Курганский государственный университет»

Кафедра «Информатика»

ВВЕДЕНИЕ В ИНФОРМАТИКУ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Часть 1

Методические указания
к выполнению лабораторных работ
по дисциплинам «Информатика», «Информационные технологии»
для студентов очной и заочной формы обучения
направлений 040400.62, 030900.62, 040100.62, 190700.62, 140400.62,
190600.62, 190109.65, 190110.65, 151900.62, 150700.62,
220700.62, 220400.62, 280700.62, 221700.62

Курган 2014

Кафедра: «Информатика»
Дисциплина: «Информатика»
(направления 030900.62, 140400.62, 190700.62, 190600.62, 151900.62, 151900.62, 280700.62, 221700.62;
специальности 190109.65, 190110.65);
«Информационные технологии»
(направления 150700.62, 220400.62, 220700.62, 040400.62, 040100.62).

Составили: ст. преподаватель Л.Г. Сысолятина, ст. преподаватель В.Я. Котликова, ст. преподаватель М.Б. Бекишева.

Утверждены на заседании кафедры «19» ноября 2013 г.
Рекомендованы методическим советом университета «30» апреля 2014 г.

1 Методы и модели оценки количества информации

Информация является одной из исходных категорий мироздания. Информацию наряду с веществом и энергией рассматривают в качестве важнейшей сущности мира, в котором мы живем. Определение информации невозможно свести к каким-то более простым, более исходным терминам. Аналогичными неопределяемыми понятиями, например, в математике, являются точка или прямая. Можно сделать некоторые утверждения, связанные с этими математическими понятиями, но определить их с помощью более элементарных понятий нельзя. На бытовом уровне и во многих научных дисциплинах термин «информация» ассоциируется с понятиями сведения, данные, знание, сообщения и др.

1.1 Объемный способ измерения информации

Объем информации в сообщении – это количество символов в сообщении.

Поскольку одно и то же число может быть записано разными способами (с использованием различных алфавитов): тридцать два, 32, XXXII, 100000_2, то этот способ чувствителен к форме представления сообщения. В вычислительной технике вся обрабатываемая и хранимая информация вне зависимости от ее природы (число, текст, звук и т.д.) представлена в двоичной форме. Такая стандартизация позволила ввести две стандартные единицы измерения: бит и байт.

1.2 Энтропийный способ измерения информации

Наука, изучающая количественные закономерности, связанные с получением, передачей, обработкой и хранением информации называется ***теорией информации***.

Любое сообщение, с которыми мы имеем дело в теории информации, представляет собой совокупность сведений о некоторой физической системе. Очевидно, если бы состояние физической системы было известно заранее, не было бы смысла передавать сообщение, т.е. сообщение приобретает смысл тогда, когда состояние системы заранее неизвестно, случайно.

Поэтому в качестве объекта, о котором передается информация, мы будем рассматривать некоторую физическую систему, которая случайным образом может оказаться в том или ином состоянии, т.е. систему, которой заведомо присуща какая-то степень неопределенности.

Возникает вопрос: что значит «большая» или «меньшая» степень неопределенности и чем можно ее измерить?

Сравним между собой две системы, каждой из которых присуща некоторая неопределенность:

1) монета, которая в результате бросания может оказаться в одном из двух состояний;
2) игральная кость, у которой 6 возможных состояний: 1, 2, 3, 4, 5, 6.

Неопределенность какой системы больше? Очевидно, второй, т.к. у нее больше возможных состояний, в каждом из которых система может оказаться с одинаковой вероятностью.

Может показаться, что степень неопределенности системы определяется числом возможных состояний системы. Но в общем случае это не так.

Рассмотрим, например, техническое устройство, которое может быть в двух состояниях: 1) исправно; 2) не исправно.

Пусть до получения сведений (априори) вероятность исправной работы устройства 0,99, а вероятность отказа – 0,01.

Такая система обладает очень малой степенью неопределенности: почти наверняка можно сказать, что устройство будет работать исправно.

При бросании монеты тоже имеется 2 возможных состояния, но степень неопределенности гораздо больше.

Таким образом степень неопределенности физической системы определяется не только числом ее возможных состояний, но и вероятностями состояний.

Перейдем к общему случаю.

Рассмотрим некоторую систему X, которая может принимать конечное множество состояний x_1, x_2, … x_n с вероятностями p_1, p_2, … p_n, где $p_i = P(X \sim x_i)$ – вероятность того, что система X примет состояние x_i. Очевидно, $\sum_{i=1}^{n} p_i = 1$.

Запишем эти данные в виде таблицы 1.1.

Таблица 1.1 – Вероятности появления значений состояний системы X

x_i	x_1		x_2		…	x_n
p_i	p_1		p_2			p_n

В качестве меры априорной неопределенности системы в теории информации применяется специальная характеристика, называемая **энтропией**.

Энтропией системы называется сумма произведений вероятностей различных состояний системы на логарифмы этих вероятностей, взятая с обратным знаком.

$$H(x) = -\sum_{i=1}^{n} p_i \cdot \log_a p_i \qquad (1.1)$$

(знак «–» поставлен для того, чтобы энтропия была положительной; числа $0<p_i<1$ и их логарифмы отрицательны).

Логарифм может быть взят при любом основании a>1. Перемена основания равносильна умножению энтропии на постоянное число, а выбор основания равносилен выбору определенной единицы измерения энтропии. Если за основание выбрано число 10, то говорят о «десятичных единицах» энтропии, если 2 – о «двоичных единицах».

На практике удобнее всего пользоваться логарифмами при основании 2 и измерять энтропию в двоичных единицах. Это хорошо согласуется с применяемой в компьютерах двоичной системой счисления.

Энтропия системы, которая имеет два равновозможных состояния (в качестве основания логарифма выберем 2) равна единице:

$$H(X) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.$$

Определенная таким образом единица энтропии называется двоичной единицей и обозначается **bit** (от английского binary digit – двоичный знак)

Это энтропия одного разряда двоичного числа, если он с одинаковой вероятностью может быть нулем или единицей.

Энтропия системы X, которая имеет n равновероятных состояний, измеренная в двоичных единицах, определяется по формуле (1.2)

$$H(X) = -\left(\frac{1}{n}\log_2\frac{1}{n} + \frac{1}{n}\log_2\frac{1}{n} + \ldots\right) = -n\cdot\frac{1}{n}\left(\log_2\frac{1}{n}\right) = -\log_2 1 + \log_2 n = \log_2 n \qquad (1.2)$$

Таким образом,

$$H(X) = \log_2 n, \qquad (1.3)$$

т.е. энтропия системы с равновозможными состояниями равна логарифму числа состояний.

Например, для системы с восемью состояниями H(X) = $\log_2 8 = 3$.

Свойства энтропии [1].

1) Энтропия обращается в ноль, когда одно из состояний системы достоверно, а другие невозможны (действительно, в этом случае все вероятности p_1, p_2, …, p_n обращаются в нуль, кроме одной, например, p_k, которая равна единице. $p_k \log p_k = 1\cdot\log 1 = 0$).

2) При заданном числе состояний энтропия обращается в максимум, когда эти состояния равновероятны. При увеличении числа состояний энтропия увеличивается.

3) Свойство аддитивности. Когда несколько независимых систем объединяются в одну, их энтропии складываются.

1.3 Энтропия и информация

Итак, мы определили энтропию как меру неопределенности состояния некоторой физической системы. Очевидно, что в результате получения сведений неопределенность системы может быть уменьшена. Чем больше объем полученных сведений, чем более содержательны сведения, тем больше информации о системе, тем менее неопределенным будет состояние системы.

Поэтому естественно количество информации измерять уменьшением энтропии той системы, для уточнения состояния которой предназначены сведения.

Рассмотрим некоторую систему X, над которой производится наблюдение, и оценим информацию, получаемую в результате того, что состояние системы X становится полностью известным. До получения информации энтропия системы была H(X). После получения сведений состояние системы полностью определилось, т.е. энтропия стала равной нулю.

Обозначим I_X информацию, получаемую в результате выяснения состояния системы X. Она равна уменьшению энтропии:

$$I_X = H(X) - 0$$

или

$$I_X = H(X), \tag{1.4}$$

т.е. количество информации, приобретаемое при полном выяснении состоянии некоторой физической системы, равно энтропии этой системы.

Учитывая (1.1), при a=2, получим

$$I_X = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i \,, \tag{1.5}$$

где $p_i = P(X \sim x_i)$.

Формула (1.5) означает, что информация I_X есть осредненное по всем состояниям системы значение логарифма вероятности состояния с обратным знаком.

Действительно, для получения I_x каждое значение $\log p_i$ (логарифм вероятности *i*-го состояния) со знаком минус множится на вероятность этого состояния и все такие произведения складываются. Естественно каждое отдельное слагаемое $-\log\ p_i$ рассматривать как частную информацию, получаемую от отдельного сообщения, состоящего в том, что система *X* находится в состоянии x_i. Обозначим эту информацию I_{x_i} :

$$I_{x_i} = -\log_2 p_i.$$

Тогда информация I_x представится как средняя (или полная) информация, получаемая от всех возможных отдельных сообщений с учетом их вероятностей.

Если все возможные состояния системы одинаково вероятны ($p_1=p_2=\ldots=p_n=1/n$), то формула (1.5) может быть преобразована к виду:

$$\mathbf{I_X=log_2 n.} \tag{1.6}$$

Если информация выражена в двоичных единицах, то ей можно дать довольно наглядное истолкование: измеряя информацию в двоичных единицах, мы условно характеризуем ее числом ответов «да» или «нет», с помощью которых можно приобрести ту же информацию.

Сообщение, уменьшающее неопределенность знаний человека в два раза несет для него один бит информации.

Рассмотрим систему с двумя состояниями (таблица 1.2):

Таблица 1.2 – Вероятности появления значений системы 2-х состояний

x_i	x_1	x_2
p_i	p_1	p_2

Чтобы выяснить состояние этой системы, достаточно задать один вопрос: находится ли система в состоянии x_1?

Ответ «да» или «нет» доставляет некоторую информацию, которая достигает своего максимального значения 1, когда оба состояния равновероятны: $p_1=p_2=1/2$, таким образом максимальная информация, даваемая ответом «да» или «нет», равна одной двоичной единице.

Если информация от какого-то сообщения равна n двоичным единицам, то она равносильна информации, даваемыми n ответами «да» или «нет» на вопросы, поставленные так, что «да» и «нет» одинаково вероятны.

Если поставить вопросы так не удается, можно утверждать только, что минимальное число вопросов, необходимых для выяснения содержания данного сообщения, не меньше, чем информация, заключенная в сообщении.

Пример. Некто задумал любое целое число X от единицы до восьми

$$1 \le X \le 8,$$

а нам предлагается угадать его, поставив минимальное число вопросов на каждый из которых дается ответ «да» или «нет».

Решение: Определяем информацию, заключенную в сообщении, какое число задумано. Априори все значения X от 1 до 8 одинаково вероятны:

$p_1 = p_2 = ... = p_8 = \dfrac{1}{8}$, и формула (1.6) дает

$$I_x = \log_2 8 = 3.$$

Минимальное число вопросов, которые нужно поставить для выяснения задуманного числа, не меньше трех.

В данном случае можно действительно обойтись тремя вопросами, если сформулировать их так, чтобы вероятности ответов «да» или «нет» были равны.

Пусть, например, задумано число «пять», мы этого не знаем и задаем вопросы:

Вопрос 1. Число X меньше пяти?

Ответ: Нет.

(Вывод: X – одно из чисел 5,6,7,8.)

Вопрос 2. Число X меньше семи?

Ответ: Да.

(Вывод: X-одно из чисел 5,6.)

Вопрос 3. Число X меньше шести?

Ответ: Да.

(Вывод: число X равно пяти.)

Легко убедиться, что тремя такими (или аналогичными) вопросами можно установить любое задуманное число от 1 до 8)

1.4 Информация и алфавит

Алфавитный подход к измерению информации позволяет определить количество информации, заключенной в тексте. Алфавитный подход является **объективным**, т.е. он не зависит от субъекта (человека), воспринимающего текст.

Множество символов, используемых при записи текста, называется алфавитом. Полное количество символов в алфавите называется **мощностью** (размером) **алфавита**. Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой (равновероятно), то количество информации, которое несет каждый символ, вычисляется по формуле:

$$x = \log_2 N, \text{ (так как } N = 2^x\text{)},$$

где N – мощность алфавита. Следовательно, в 2-символьном алфавите каждый символ «весит» 1 бит ($\log_2 2 = 1$); в 4-символьном алфавите каждый символ несет 2 бита информации ($\log_2 4 = 2$); в 8-символьном – 3 бита ($\log_2 8 = 3$) и т.д.

Один символ из алфавита мощностью 256 (2^8) несет в тексте 8 бит информации. Такое количество информации называется **байт**. ***Алфавит из 256 символов используется для представления текстов в компьютере***.

1 байт = 8 бит.

Если весь текст состоит из К символов, то при алфавитном подходе размер содержащейся в нем информации равен:

$$V = k \cdot x,$$

где x – информационный вес одного символа в используемом алфавите.

Для измерения информации используются и более крупные единицы:

1 Кбайт (килобайт) = 2^{10} байт = 1024 байта;

1 Мбайт (мегабайт) = 2^{20} байт = 1024 Кбайта;

1 Гбайт (гигабайт) = 2^{30} байт = 1024 Мбайта;

1 Тбайт (терабайт) = 2^{40} байт = 1024 Гбайта.

Пример 4. Книга, набранная с помощью компьютера, содержит 150 страниц; на каждой странице – 40 строк, в каждой строке – 60 символов. Каков объем информации в книге?

Решение: Мощность компьютерного алфавита равна 256. Один символ несет 1 байт информации. Значит, страница содержит 40 х 60 = 2400 байт информации. Объем информации в книге (в разных единицах):

2400 х 150 = 3600 байт.

360000/1024 = 351,5625 Кбайт.

351,5625/1024 = 0,34332275 Мбайт.

2 Системы счисления

2.1 Понятие о системах счисления

Любое число имеет значение (содержание) и форму представления. Значение числа задает его отношение к значениям других чисел («больше», «меньше», «равно») и, следовательно, порядок расположения чисел на числовой оси. Форма представления определяет порядок записи числа с помощью предназначенных для этого знаков. При этом значение числа не зависит от способа его представления. Число с одним и тем же значением может быть записано по-разному, т.е. отсутствует взаимно однозначное соответствие между представлением числа и его значением. В связи с этим возникают вопросы о формах представления чисел и о способах перехода от одной формы к другой.

Способ представления числа определяется системой счисления.

Система счисления – это совокупность приемов и правил для обозначения и наименования чисел.

• ***Алфавит системы счисления*** – это множество всех символов (знаков), используемых для записи чисел в данной системе счисления.

• ***Цифра*** – это любой символ (знак), входящий в алфавит данной системы счисления.

Системы счисления делятся на следующие виды:

1) унарные системы;
2) кодовые (непозиционные) системы;
3) позиционные системы.

Простейшая и самая древняя – так называемая унарная система счисления. В ней для записи любых чисел используется всего один символ. Длина записи числа при таком кодировании прямо связана с его величиной.

Система счисления называется непозиционной, если количественный эквивалент (значение) каждого символа не зависит от его положения (места, позиции) в коде числа.

Примером непозиционной системы, которая сохранилась до наших дней, может служить система счисления, которая применялась более двух с половиной тысяч лет назад в Древнем Риме.

В основе римской системы счисления лежали знаки I (один палец) для числа 1, V (раскрытая ладонь) для числа 5, X (две сложенные ладони) для 10, а обозначения C для 100 и M для 1000 – это первые буквы соответствующих латинских слов (centum – сто, mille – тысяча).

Чтобы записать число, римляне разлагали его на сумму тысяч, полутысяч, сотен, полусотен, десятков, пятерок, единиц.

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Например, десятичное число 28 представляется следующим образом:

XXVIII =X+X + V+I + I + I

(два десятка, пятерка, три единицы).

Римская система счисления сегодня используется в основном для наименования знаменательных дат, томов, разделов и глав в книгах.

Конечно, непозиционные системы счисления гораздо удобнее, чем унарная, но и они имеют ряд недостатков:

1) постоянная потребность введения новых знаков для записи больших чисел. Например, имея знаки I, V, X, трудно изобразить тысячу. И всегда можно придумать число, которое трудно изобразить даже вновь введенными знаками;

2) невозможно представлять дробные числа;

3) сложно выполнять арифметические операции, так как нет определенных правил действий над числами.

• ***Система счисления называется позиционной, если количественный эквивалент (значение) символа зависит от его положения (места, позиции) в записи числа.***

В десятичной позиционной системе счисления для записи чисел используются десять различных знаков (цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Не только сама цифра, но и ее место, ее позиция имеют определяющее значение: из двух написанных рядом цифр левая выражает единицы, в десять раз большие, чем правая, т.е. 10 единиц какого-либо разряда образуют единицу следующего, старшего разряда. Другими словами, единицы различных разрядов представляют собой различные степени числа 10.

В десятичном числе

$$A_{10} = 255 = 2 \cdot 10^2 + 5 \cdot 10^1 + 5 \cdot 10^0$$

цифры 5, находящиеся на разных позициях, имеют различные количественные значения – 5 десятков и 5 единиц. При перемещении цифры на соседнюю позицию ее вес (количественный эквивалент) изменяется в 10 раз. Поэтому данную систему называют позиционной.

Общим для унарной и римской систем счисления является то, что значение числа в них определяется посредством операций сложения и вычитания базисных цифр, из которых составлено число, независимо от их позиции в числе. Такие системы получили название **аддитивных**. В отличие от них позиционное представление следует считать **аддитивно-мультипликативным**, поскольку значение числа определяется операциями умножения и сложения.

Основные достоинства любой позиционной системы счисления – простота выполнения арифметических операций и ограниченное количество символов, необходимых для записи любых чисел.

2.2 Представление чисел в позиционных системах счисления

Основание позиционной системы счисления – это количество единиц младшего разряда числа, объединяемых в одну единицу старшего разряда.

Возможно множество позиционных систем, так как за основание системы счисления можно принять любое число, не меньшее 2. Наименование системы счисления соответствует ее основанию (десятичная, двоичная, пятеричная и т.д.).

Для записи чисел в позиционной системе с основанием q нужно иметь алфавит из q цифр, изображающих 0, 1, …, q – 1.

Обычно при $q<10$ используют q первых арабских цифр, а при $q>10$ к десяти арабским цифрам добавляют буквы. Примеры алфавитов нескольких систем счисления представлены в таблице 2.1.

Таблица 2.1 – Алфавиты систем счисления с различными основаниями

Основание	Название	Алфавит
q=2	двоичная	0 1
q=3	троичная	0 1 2
q=8	восьмеричная	0 1 2 3 4 5 6 7
q=16	шестнадцатеричная	0 1 2 3 4 5 6 7 8 9 A B C D E F

Если требуется указать основание системы, к которой относится число, то оно приписывается нижним индексом к этому числу. Например:

$$101101_2,\ 3671_8,\ 3B8F_{16}.$$

В системе счисления с основанием q (q-ичная система счисления) единицами разрядов служат последовательные степени числа q, иначе говоря, q единиц какого-либо разряда образуют единицу следующего разряда.

Основание позиционной системы счисления равно количеству символов (знаков) в ее алфавите. Основание в любой системе изображается как 10, но имеет разное количественное значение, оно показывает, во сколько раз изменяется количественное значение цифры при перемещении ее на соседнюю позицию.

Развернутая и свернутая формы записи чисел

В позиционной системе счисления любое вещественное число может быть представлено в следующем виде:

$$A_q=\pm(a_{n-1}\cdot q^{n-1}+a_{n-2}\cdot q^{n-2}+\ldots+a_0 q^0+a_{-1} q^{-1}+a_{-2} q^{-2}+\ldots+a_{-m} q^{-m}), \quad (2.1)$$

или

$$A_q=\pm\sum_{i=-m}^{n-1} a_i q^i \quad (2.2)$$

где A – число;

q – основание системы счисления;

a_i – цифры, принадлежащие алфавиту данной системы счисления;

n – количество целых разрядов чисел;

m – количество дробных разрядов чисел.

Разложение числа по формуле (2.1) называется ***развернутой*** формой записи.

Пример 1

Получить развернутую форму десятичных чисел 4538,62; 53428.

Решение:

$4538{,}62=4*10^3+5*10^2+3*10^1+8*10^0+6*10^{-1}+2*10^{-2}$.

$53428=5*10^4+3*10^3+4*10^2+2*10^1+8*10^0$.

Пример 2

Получить развернутую форму чисел 112_3, 100101_2,$14FC_{16}$, $101{,}01_2$.

Решение:

$112_3 = 1*10^2+1*10^1+2*10^0$

$100101_2 = 1*10^{101}+0*10^{100}+0*10^{11}+1*10^{10}+0*10^1+1*10^0$

$14FC_{16} = 1*10^3+4*10^2+F*10^1+C*10^0$

$101{,}01_2 = 1*10^{10}+0*10^1+1*10^0+0*10^{-1} +1*10^{-10}$

Обратите внимание на то, что в любой системе счисления ее основание записывается как 10.

Свернутой формой записи числа называется его представление в виде

$$A = a_{n-1}a_{n-2}\ldots a_1a_0a_{-1}\ldots a_{-m}.$$

Именно такой формой записи чисел мы и пользуемся в повседневной жизни, иначе свернутую форму записи называют ***естественной*** или ***цифровой***.

2.3 Перевод целых чисел из одной системы счисления в другую

Поскольку одно и тоже число может быть записано в различных системах счисления, возникает вопрос о переводе представления числа из одной системы (p) в другую (q). Будем обозначать такое представление $A_p \rightarrow A_q$. Теоретически можно произвести перевод для любых p и q. Однако подобный прямой перевод будет затруднен тем, что придется выполнять операции по правилам арифметики не десятичных систем счисления. По этой причине более удобным можно считать вариант преобразования с промежуточным переводом $A_p \to A_r \to A_q$ с основанием r, для которого арифметические операции выполнить легко. Такими удобными основаниями являются r=1 и r=10, т.е. перевод осуществляется через унарную или десятичную систему счисления.

Преобразование $A_p \to A_1 \to A_q$

Идея алгоритма перевода следующая: положим начальное значение $A_q=0$. Из числа A_p вычтем единицу по правилам вычитания системы p и добавим ее к A_q по правилам сложения системы q; будем выполнять эту последовательность действий пока не достигнем $A_p=0$.

Пример 3

Выполнить преобразование $22_3 \to A_6$.

Решение:

Последовательность действий и промежуточные результаты для наглядности представим в виде следующей таблицы 2.2.

Таблица 2.2 – Преобразование $A_3 \to A_1 \to A_6$

Шаг	0	1	2	3	4	5	6	7	8
$A_3 - 1$	22	21	20	12	11	10	2	1	0
$A_6 + 1$	0	1	2	3	4	5	10	11	12

Следовательно, $22_3 \to 12_6$.

Преобразование $A_p \to A_{10} \to A_q$

Очевидно, первая и вторая части преобразования не связаны друг с другом, таким образом можно рассматривать их по отдельности.

Преобразование $A_{10} \to A_q$

Поскольку A_{10} целое десятичное число в его разложении отсутствуют коэффициенты с отрицательными индексами и его можно представить в виде.

$A_{10} = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \ldots + a_2 \cdot q^2 + a_1 \cdot q^1 + a_0 \, q^0$.

Разделим число A_{10} на q. Частное будет равно

$$a_{n-1} \cdot q^{n-2} + \ldots + a_2 \cdot q + a_1,$$

а остаток будет равен a_0.

Полученное неполное частное опять разделим на q, остаток от деления будет равен a_1.

Если продолжить этот процесс деления, то на n-м шаге получим набор цифр a_0, a_1, a_2,…, a_{n-1}, которые входят в q-ичное представление числа A_{10} и совпадают с остатками при последовательном делении данного числа на q. Но мы их получили в порядке, обратном порядку расположения в q-ичном представлении числа A.

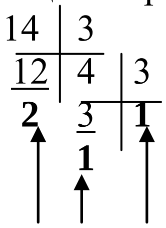
$A_{10} = (a_{n-1} a_{n-2} \ldots a_1 a_0)_q$.

Пример 4

Перевести $14_{10} \to A_3$.

Решение:

Рассмотренную выше последовательность действий удобнее изображать следующим образом:



a_0 a_1 a_2

Записывая остатки от деления в обратном направлении получим число 112_3.

$14_{10} = 112_3$.

Преобразование $A_p \rightarrow A_{10}$

Данное преобразование явно вытекает из формулы (2.1): **если все слагаемые в развернутой форме не десятичного числа представить в десятичной системе и вычислить полученное выражение по правилам десятичной арифметики, то получится число в десятичной системе равное данному** (это правило распространяется как на целые, так и на дробные числа).

Пример 5

Перевести $112_3 \rightarrow A_{10}$, $100101_2 = \rightarrow A_{10}$, $14FC_{16} \rightarrow A_{10}$, $101{,}01_2 \rightarrow A_{10}$

Решение:

$112_3 = (1*10^2+1*10^1+2*10^0)_3 = (1*3^2+1*3^1+2*3^0)_{10}=14_{10}$

$100101_2=(1*10^{101}+0*10^{100}+0*10^{11}+1*10^{10}+0*10^1+1*10^0)_2=(1*2^5+0*2^4+0*2^3+1*2^2+$
$0*\ *2^1+1*2^0)_{10}=37_{10}$

$14FC_{16} = (1*10^3+4*10^2+F*10^1+C*10^0)_{16}=(1*16^3+4*16^2+15*16^1+12*16^0)_{10}=5372_{10}$

$101{,}01_2=(1*10^{10}+0*10^1+1*10^0+0*10^{-1}+1*10^{-10})_2=(1*2^2+0*2^1+1*2^0+0*2^{-1}+1*2^{-2})_{10}=$
$=5{,}25_{10}$

2.4 Перевод дробных чисел из одной системы счисления в другую

Вещественное число, в общем случае содержащее целую и дробную части, всегда можно представить в виде суммы целого числа и правильной дроби. Поскольку ранее проблема записи натуральных чисел в различных системах счисления была решена, рассмотрим правило перевода правильных дробей.

$A_{10} \rightarrow A_q$

Пусть A_{10} – десятичная дробь, тогда в ее разложении отсутствуют коэффициенты с положительными индексами.

$$A_{10}= a_{-1}\cdot q^{-1}+ a_{-2}\cdot q^{-2}+ a_{-3}\cdot q^{-3} \qquad (2.3)$$

Таким образом, необходимо найти коэффициенты a_{-1}, a_{-2},…, входящие в запись числа в q-ичной системе счисления.

Умножим правую и левую части выражения (2.3) на q. В результате в правой части получим:

$a_{-1}+ a_{-2}\cdot q^{-1}+ a_{-3}\cdot q^{-2}+\ldots$

Целая часть здесь равна a_{-1}, она и даст нам старший коэффициент в разложении числа A_{10}.

Оставшуюся дробную часть умножим на q: $a_{-2}\cdot + a_{-3}\cdot q^{-1}+\ldots$.

Цифра a_{-2} представляет собой второй коэффициент после запятой в двоичном представлении исходного числа.

Описанный процесс необходимо продолжать до тех пор, пока в правой части не получим нуль или пока не будет достигнута необходимая точность вычислений.

Пример 6

Перевести десятичную дробь 0,5625 в двоичную систему счисления.

Решение:

Вычисления лучше всего оформлять по следующей схеме:

0,	5625 x 2
1	1250 x 2
0	2500 x 2
0	5000 x 2
1	0000

$0{,}5625_{10}=0{,}1001_2$.

Пример 7

Перевести в двоичную систему счисления десятичную дробь 0,7.

Решение:

0,	7 x 2
1	4 x 2
0	8 x 2
1	6 x 2
1	2

…….

Очевидно, что этот процесс может продолжаться бесконечно, давая все новые и новые знаки в изображении двоичного эквивалента числа $0{,}7_{10}$. Так, за четыре шага мы получаем число $0{,}1011_2$, а за семь шагов число $0{,}1011001_2$, которое является более точным представлением числа $0{,}7_{10}$ в двоичной системе счисления. Такой бесконечный процесс обрывают на некотором шаге, когда считают, что получена требуемая точность представления числа.

2.5 Перевод смешанных чисел

Перевод смешанных чисел, содержащих целую и дробную части, осуществляется в два этапа. Отдельно переводится целая часть, отдельно – дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой (точкой).

Пример 8

Перевести число $17{,}25_{10}$ в двоичную систему счисления.

Решение:

Переводим целую часть:

```
17 | 2
 1 | 8 | 2
     0 | 4 | 2
         0 | 2 | 2
             0 | 1
```

Переводим дробную часть:

0,	25
	x 2
0	50
	x 2
1	00

Ответ: $17{,}25_{10}=10001{,}01_2$.

ОБЩЕЕ ПРАВИЛО
перевода целых чисел из системы счисления с основанием 10 в систему счисления с основанием q:

1) основание новой системы счисления выразить цифрами исходной системы счисления и все последующие действия производить в исходной системе счисления;
2) последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получим частное, меньшее делителя;
3) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствии с алфавитом новой системы счисления;
4) составить число в новой системе счисления, записывая его начиная с последнего остатка.

Здесь и далее под исходной подразумевается система счисления с основанием 10, под новой – с основанием q.

ОБЩЕЕ ПРАВИЛО
перевода правильной дроби: из системы счисления с основанием 10 в систему счисления с основанием q:

1) основание новой системы счисления выразить цифрами исходной системы счисления и все последующие действия производить в исходной системе счисления;
2) последовательно умножать данное число и получаемые дробные части произведений на основание новой системы счисления до тех пор, пока дробная часть произведения не станет равной нулю или пока не будет достигнута требуемая точность представления числа;
3) полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствии с алфавитом новой системы счисления;
4) составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

2.6 Перевод чисел между системами счисления 2↔8↔16

Интерес к двоичной системе счисления вызван тем, что именно эта система используется для представления чисел в компьютере. Однако двоичная запись оказывается громоздкой, поскольку содержит много цифр, и плохо воспринимается и запоминается человеком из-за зрительной однородности (все число состоит из нулей и единиц). Поэтому в нумерации ячеек памяти компьютера, записи кодов команд, нумерации регистров и устройств используются системы счисления с основаниями 8 и 16; выбор именно этих систем счисления обусловлен тем, что переход от них к двоичной системе и обратно осуществляется, как будет показано ниже, весьма простым образом.

Пользуясь алгоритмами, сформулированными в предыдущих разделах, можно заполнить таблицу 2.3 и таблицу 2.4. Примем без доказательства две теоремы.

Теорема 1. Для преобразования целого числа $A_p \to A_q$ в том случае, если основания систем счисления связаны соотношением $q=p^r$, где r – целое число большее 1, достаточно A_p разбить справа налево на группы по r цифр и каждую из них независимо перевести в систему q. (При необходимости исходное число следует дополнить незначащими нулями слева до группы в r цифр.)

Пример 9

Выполнить преобразование $110001_2 \to A_8$.

Решение:

Исходное число разбивается на группы по три разряда справа налево ($8=2^3$, следовательно, r=3) и каждая тройка в соответствии с таблицей 2.3 переводится в 8-ричную систему счисления независимо от остальных троек:

$$\underbrace{110}_{6}\underbrace{001}_{1}$$

Следовательно, $110001_2=61_8$. Аналогично, разбивая A_2 на группы по 4 двоичные цифры и дополняя старшую группу незначащими нулями слева, получим $110001_2=31_{16}$.

Таблица 2.3 – Двоично-восьмеричная таблица

8	**2**
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Таблица 2.4 – Двоично-шестнадцатеричная таблица

16	**2**	**16**	**2**
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Теорема 2. Для преобразования целого числа $A_p \to A_q$ в том случае, если системы счисления связаны соотношением $p=q^r$, где r – целое число большее 1, достаточно каждую цифру A_p заменить соответствующим r-разрядным числом в системе счисления q.

Пример 10

Выполнить преобразование $D3_{16} \rightarrow A_2$.

Решение:

$D3_{16}=\underbrace{1101}_{D}\underbrace{0011}_{3}{}_2$

Переходы $A_8 \rightarrow A_{16}$ и $A_{16} \rightarrow A_8$, очевидно, удобнее осуществлять через промежуточный переход к двоичной системе. Например, $123_8=001010011_2=53_{16}$.

2.7 Арифметические операции

Сложение

Таблицы сложения для двоичной, восьмеричной и шестнадцатеричной систем счисления представлены на рисунке 2.1.

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

Сложение в двоичной системе

+	**0**	**1**
0	0	1
1	1	10

Сложение в восьмеричной системе

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Сложение в шестнадцатеричной счисления

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Рисунок 2.1 – Таблицы сложения для двоичной, восьмеричной и шестнадцатеричной систем счисления

Пример 11. Сложить десятичные числа 15 и 6 в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Решение:

Двоичная: 1111_2+110_2 **Восьмеричная**: 17_8+6_8 **Шестнадцатеричная** $F_{16}+6_{16}$

$$\begin{array}{r} 1111 \\ +\,\underline{0110} \\ 10101 \end{array} \qquad \begin{array}{r} 17 \\ +\,\underline{06} \\ 25 \end{array} \qquad \begin{array}{r} F \\ +\,\underline{\ 6} \\ 15 \end{array}$$

Ответ: $15+6=21_{10}=10101_2=25_8=15_{16}$.

Проверка. Преобразуем полученные суммы к десятичному виду:

$10101_2=2^4+2^2+2^0=16+4+1=21_{10}$,

$25_8=2*8^1+5*8^0=16+5=21_{10}$,

$15_{16}=1*6^1+5*16^0=16+5=21_{10}$.

Пример 12. Сложить десятичные числа 141,5 и 59,75 в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Решение:

Двоичная: $10001101{,}1_2+111011{,}11_2$ **Восьмеричная**: $215{,}4_8+73{,}6_8$

$$\begin{array}{r} 1000110{,}10_2 \\ +\ \underline{\ \ 11101{,}11_2} \\ 11001001{,}01_2 \end{array} \qquad \begin{array}{r} 215{,}4_8 \\ +\,\underline{\ 73{,}6_8} \\ 311{,}2_8 \end{array}$$

Шестнадцатеричная $8D{,}8_{16}+3B{,}C_{16}$

$$\begin{array}{r} 8D{,}8_{16} \\ +\,\underline{3B{,}C_{16}} \\ C9{,}4_{16} \end{array}$$

Ответ: $141{,}5_{10}+59{,}75_{10}=201{,}25_{10}=11001001{,}01_2=311{,}2_8=C9{,}4_{16}$.

Проверка. Преобразуем полученные суммы к десятичному виду:

$11001001{,}01_2=2^7+2^6+2^3+2^0+2^{-2}=201{,}25_{10}$;

$311{,}2_8=3*8^2+1*8^1+1*8^0+2*8^{-1}=201{,}25_{10}$;

$C9{,}4_{16}=12*16^1+9*16^0+4*16^{-1}=201{,}25_{10}$.

Вычитание

Пример 13. Вычесть единицу из чисел 10_2, 10_8 и 10_{16}.

Решение:

Двоичная: 10_2-1_2 **Восьмеричная**: 10_8-1_8 **Шестнадцатеричная** $10_{16}-1_{16}$

$$\begin{array}{r} 10_2 \\ -\,\underline{\ 1_2} \\ 1_2 \end{array} \qquad \begin{array}{r} 10_8 \\ -\,\underline{\ 1_8} \\ 7_8 \end{array} \qquad \begin{array}{r} 10_{16} \\ -\,\underline{\ 1_{16}} \\ F_{16} \end{array}$$

Пример 14. Вычесть число 59,75 из числа 201,25 в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Решение:

Двоичная: $11001001{,}01_2-111011{,}11_2$ **Восьмеричная**: $311{,}2_8-73{,}6_8$

$$\begin{array}{r} 11001001{,}01 \\ -\,\underline{00111011{,}11} \\ 10001101{,}10 \end{array} \qquad \begin{array}{r} 311{,}2 \\ -\,\underline{\ 73{,}6} \\ 215{,}4 \end{array}$$

Шестнадцатеричная $C9{,}4_{16}+3B{,}C_{16}$

$$\begin{array}{r} C9{,}4 \\ -\,\underline{3B{,}C} \\ 8D{,}8 \end{array}$$

Ответ: $201{,}25_{10}-59{,}75_{10}=141{,}5_{10}=10001101{,}1_2=215{,}4_8=8D{,}8_{16}$.

Проверка. Преобразуем полученные суммы к десятичному виду:
$10001101,1_2=2^7+2^3+2^2+2^0+2^{-1}=141,5_{10}$;
$215,4_8=2*8^2+1*8^1+5*8^0+4*8^{-1}=141,5_{10}$;
$8D,8_{16}=8*16^1+D*16^0+8*16^{-1}=141,5_{10}$.

Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо брать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Таблицы умножения для двоичной, восьмеричной и шестнадцатеричной систем счисления представлены на рисунке 2.2.

Умножение в двоичной системе

x	**0**	**1**
0	0	0
1	0	1

Умножение в восьмеричной системе

x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Умножение в шестнадцатеричной системе

x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1a	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1c	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1e	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Рисунок 2.2 – Таблицы умножения для двоичной, восьмеричной и шестнадцатеричной систем счисления

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе умножение сводится лишь к сдвигам множимого и сложениям.

Пример 15. Перемножить числа 5 и 6 в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Решение:

Двоичная: $101_2 * 110_2$ **Восьмеричная**: $5_8 * 6_8$ **Шестнадцатеричная** $5_{16} * 6_{16}$

$$\begin{array}{r} 101 \\ \times\ \underline{110} \\ 101 \\ +\underline{101} \\ 11110 \end{array} \qquad \begin{array}{r} 5 \\ \times\ \underline{6} \\ 36 \end{array} \qquad \begin{array}{r} 5 \\ \times\ \underline{6} \\ 1E \end{array}$$

Ответ: $5 * 6 = 30_{10} = 11110_2 = 36_8 = 1E_{16}$.

Проверка. Преобразуем полученные суммы к десятичному виду:

$11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30_{10}$;

$36_8 = 3*8^1 + 6*8^0 = 30_{10}$;

$1E_{16} = 1*16^1 + E*16^0 = 30_{10}$.

Деление

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление в десятичной позиционной системе. В двоичной системе деление выполняется особенно просто, так как очередная цифра частного может быть только нолем или единицей.

Пример 16. Разделим число 5865_{10} на число 115_{10}, представив исходные данные в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Решение:

Двоичная: $1011011101001_2 : 1110011_2$ **Восьмеричная**: $13351_8 : 163_8$

$$\begin{array}{r|l} -\,1011011101001 & \underline{1110011} \\ \underline{1110011} & 110011 \\ -\,10001000 & \\ \underline{1110011} & \\ -\,10101100 & \\ \underline{1110011} & \\ -\,1110011 & \\ \underline{1111011} & \\ 0 & \end{array} \qquad \begin{array}{r|l} -\,13351 & \underline{163} \\ \underline{1262} & 63 \\ -\,531 & \\ \underline{531} & \\ 0 & \end{array}$$

Шестнадцатеричная $16E9_{16} : 73_{16}$

$$\begin{array}{r|l} -\,16E9 & \underline{73} \\ \underline{159} & 33 \\ -\,159 & \\ \underline{159} & \\ 0 & \end{array}$$

Ответ: $5865_{10} : 115_{10} = 51_{10} = 110011_2 = 63_8 = 33_{16}$.

Проверка. Преобразуем полученные суммы к десятичному виду:

$110011_2 = 2^5 + 2^4 + 2^1 + 2^1 = 51_{10}$;

$63_8 = 6*8^1 + 3*8^0 = 51_{10}$;

$33_{16} = 3*16^1 + 3*16^0 = 51_{10}$.

3 Основы алгоритмизации

3.1 Алгоритм и его свойства

Традиционно считается, что самый первый алгоритм был придуман древнегреческим математиком Евклидом. В современной математике понятие алгоритма является ключевым понятием, которое восходит к работам выдающегося узбекского математика IX века Аль-Хорезми.

Примеры алгоритмов в широком смысле этого слова можно встретить и в повседневной жизни. Так, поваренная книга является сборником алгоритмов, описывающих процессы приготовления пищи.

Компьютерные программы представляют собой алгоритмы, записанные средствами языков программирования.

Исчерпывающее определение ***алгоритма*** дано нашим соотечественником А. А Марковым. Алгоритм – это точное общепринятое предписание, определяющее процесс преобразования ***исходных данных в искомый результат***.

Исполнитель алгоритма – это тот объект или субъект, для управления которым составлен алгоритм.

Система команд исполнителя (СКИ) – это вся совокупность команд, которые исполнитель умеет выполнять.

Алгоритм должен обладать следующими ***свойствами***: понятностью, точностью, конечностью.

Понятность: алгоритм составляется только из команд, входящих в СКИ исполнителя.

Точность: каждая команда алгоритма управления определяет однозначное действие исполнителя.

Конечность (или результативность): выполнение алгоритма должно приводить к результату за конечное число шагов.

Среда исполнителя: обстановка, в которой функционирует исполнитель.

Определенная последовательность действий исполнителя всегда применяется к некоторым ***исходным данным.*** Например, для приготовления блюда по кулинарному рецепту нужны соответствующие продукты (данные). ***Полный набор данных*: необходимый и достаточный набор данных для решения поставленной задачи (получения искомого результата).

Для того чтобы сделать алгоритм более наглядным, часто используют ***блок-схемы***.

Блок-схема алгоритма представляет собой систему блоков, соединенных между собой линиями, которые указывают последовательность выполнения блоков. Направление на линии связи между блоками указывается стрелкой в конце линии. Если линия связи направлена сверху вниз и слева направо, то стрелка на этой линии не ставится.

Блоки изображаются геометрическими фигурами, внутри которых дается текстоформульная информация. Блок, организующий ввод или вывод данных, изображается параллелограммом. Для изображения разветвления используется ромб, внутри которого условие разветвления. Для обозначения начала и конца алгоритма используются прямоугольники с закругленными углами. Допускается нумерация блоков путем разрыва верхней стороны их контура и записи на месте разрыва номера блока.

При составлении блок-схемы алгоритма рекомендуется располагать ее на одной странице и придерживаться правил, установленных ГОСТ 19.002-80 «Схемы алгоритмов и программ. Правила выполнения», ГОСТ 19.003-80 «Схемы алгоритмов и программ. Обозначения условные графические» и ГОСТ 19.701-90 «Единая система программной документации. СХЕМЫ АЛГОРИТМОВ, ПРОГРАММ, ДАННЫХ И СИСТЕМ. Обозначения условные и правила выполнения», дата введения которого 01.01.92. ГОСТ 19.701-90 имеет статус действующего в настоящее время, разработан на основе ГОСТ 19.002-80 и ГОСТ 19.003-80.

ОБЩИЕ ТРЕБОВАНИЯ:

– схемы алгоритмов, программ данных и систем состоят из имеющих заданное значение символов, краткого пояснительного текста и соединяющих линий;

– схемы могут использоваться на различных уровнях детализации, причем число уровней зависит от размеров и сложности задачи обработки данных. Уровень детализации должен быть таким, чтобы различные части и взаимосвязь между ними были понятны в целом;

– схема – графическое представление определения, анализа или метода решения задачи, в котором используются символы для отображения операций, данных, потока, оборудования т.д.

3.2 Правила выполнения схем

3.2.1 Для облегчения вычерчивания и нахождения на схеме зон устанавливают с учетом минимальных размеров символов, изображенных на данном листе. Допускается один символ размещать в двух и более зонах, если размер символа превышает размер зоны.

3.2.2 Координаты зоны представляют:

- по горизонтали – арабскими цифрами слева направо в верхней части листа;
- по вертикали – прописными буквами латинского алфавита сверху вниз в левой части листа.

3.2.3 Координаты зон в виде сочетания букв и цифр присваиванию символам, вписанным в поля этих зон, например, *A1*, *A2*, *A3*, *B1*, *B2*, *B3* и т.д.

При выполнении схем от руки, если поле листа не разбито на зоны, символам присваивают порядковые номера.

3.2.4 В пределах одной схемы, при выполнении ее от руки, допускается применять не более двух смежных размеров ряда чисел, кратных 5.

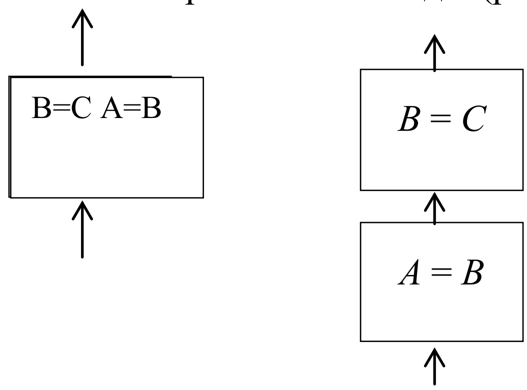
3.2.5 Линии потока должны быть параллельны линиям внешней рамки схемы.

3.2.6 Расстояние между параллельными линиями потока должны быть не менее 3 мм, между остальными символами схемы – не менее 5 мм.

3.2.7 Записи внутри символа или рядом с ним должны быть краткими. Сокращения слов и аббревиатуры, за исключением установленных государственными стандартами, должны быть расшифрованы в нижней части поля схемы или в документе, к которому эта схема относится.

3.2.8 Для удобства детализации программы должны быть использованы символы «Процесс», «Решение», «Модификация», «Ввод-вывод» и «Пуск-останов», при этом внутри символов на расстоянии не менее 0,25 *а* проводят тонкую линию (размер *а* по ГОСТ 19.003-80)

3.2.9 Записи внутри символов должны быть представлены так, чтобы их можно было читать слева направо и сверху вниз, независимо от направления потока. Вид *а* должен быть прочитан как вид *б* (рисунок 3.1).



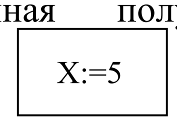
а *б*

Рисунок 3.1 – Способы изображений блоков присваивания

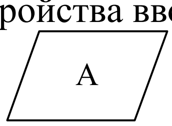
3.3 Основные типы алгоритмических структур

3.3.1 Линейный алгоритм

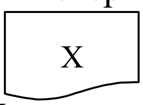
Алгоритм, в котором команды выполняются последовательно одна за другой, называется ***линейным алгоритмом***. Линейный алгоритм может быть построен из команд присваивания, ввода и вывода.

Команда присваивания – команда исполнителя, в результате которой переменная получает новое значение, графическое изображение которой 

Пример: X:=5 Переменной X присваивается значение, равное 5.

Команда ввода – команда, по которой значения переменных задаются через устройства ввода (например, клавиатуру), графическое изображение которой 

Пример: *ввод* A – ввод значения переменной A с клавиатуры компьютера.

Команда вывода: команда, по которой значение величины отражается на устройстве вывода компьютера (например, экран дисплея, бумага), графическое изображение которой 

Пример: *вывод* X – значение переменной X выводится на экран.

На блок-схеме (рисунок 3.2) изображена структура линейного алгоритма.

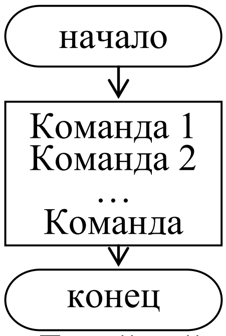


Рисунок 3.2 – Линейный алгоритм

3.3.2 Алгоритмическая структура «Ветвление»

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в алгоритмическую структуру «***ветвление***» входит ***условие***, в зависимости от выполнения или невыполнения которого реализуется та или иная последовательность команд (серия).

Будем называть условием высказывание, которое может быть либо истинным, либо ложным. Условие, записанное на формальном языке, называется условным или логическим выражением.

Условные выражения могут быть простыми и сложными. Простое условие включает в себя два числа, две переменных или два арифметических выражения, которые сравниваются между собой с использованием операций сравнения (равно, больше, меньше и пр.). Например: 5>3, 2* 8=4 * 4 и т.д.

Сложное условие – это последовательность простых условий, объединенных между собой знаками логических операций.

Например: (5>3) **and** (2* 8=4 * 4); (a>=b) **or** (c<d).

Алгоритмическая структура «*ветвление*» может быть зафиксирована различными способами:

- графически, с помощью блок-схемы;
- на языке программирования, например, на Паскале с использованием специальной инструкции ветвления (оператора условного перехода).

Ветвление бывает полное и неполное. Описание ветвления в блок-схемах и на алгоритмическом языке (**кв** – конец ветвления) представлено на рисунке 3.3:

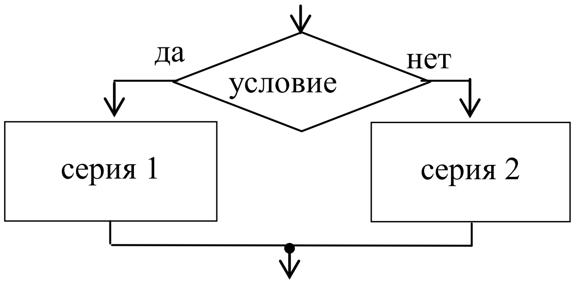
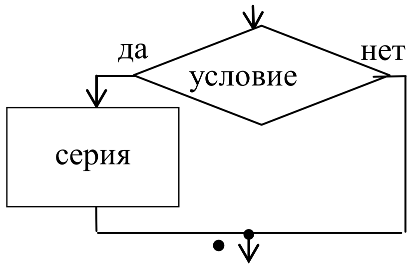
Полное ответвление	**Неполное ответвление**
	
если <условие> **то** <серия 1> **иначе** <серия 2> **кв**	**если** <условие> **то** <серия> **кв**

Рисунок 3.3 – Виды ветвлений

Алгоритм может иметь несколько ветвлений – последовательных или вложенных.

3.3.3 Алгоритмическая структура «Выбор»

Алгоритмическая структура «выбор» применяется для реализации ветвления со многими вариантами серий команд. В структуру выбора входят несколько условий, проверка которых осуществляется в строгой последовательности их записи в команде выбора. При истинности одного из условий выполняется соответствующая последовательность команд.

3.3.4 Алгоритмическая структура «Цикл»

В алгоритмическую структуру «цикл» входит серия команд, выполняемая *многократно*. Такая последовательность команд называется *телом цикла*.

Циклические алгоритмические структуры бывают двух типов:

• *циклы со счетчиком*, в которых тело цикла выполняется определенное количество раз;

• *циклы с условием*, в которых тело цикла выполняется, пока условие истинно или ложно.

Алгоритмическая структура «цикл» может быть зафиксирована различными способами:

• графически – с помощью блок-схемы;

• на языке программирования, например, на языке Паскаль с использованием специальных инструкций, реализующих циклы различного типа.

Цикл со счетчиком. Когда заранее известно, какое число повторений тела цикла необходимо выполнить, можно воспользоваться циклической инструкцией (рисунок 3.4).

В начале выполнения цикла значение переменной **i** устанавливается равным **In**. При каждом проходе цикла переменная **i** увеличивается на величину шага. Если она достигает величины, большей **Ik**, то цикл завершается и выполняются следующие за ним операторы (**нц** – начало цикла, **кц** – конец цикла) .

для i от In до Ik, повторять

нц

<тело цикла>

кц

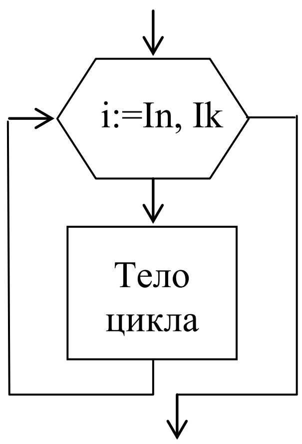


Рисунок 3.4 – Цикл со счетчиком

Циклы с условием. Часто бывает так, что необходимо повторить тело цикла, но заранее неизвестно, какое количество раз это надо сделать. В таких случаях количество повторений зависит от некоторого условия.

Условие выхода из цикла можно поставить в начале, перед телом цикла. Такой цикл называется ***циклом с предусловием***.

Цикл выполняется, пока условие имеет значение «истина». Как только условие примет значение «ложь», выполнение цикла закончится. В этом случае условие является *условием продолжения цикла* (рисунок 3.5).

пока <условие>, повторять
нц
    <тело цикла>
кц

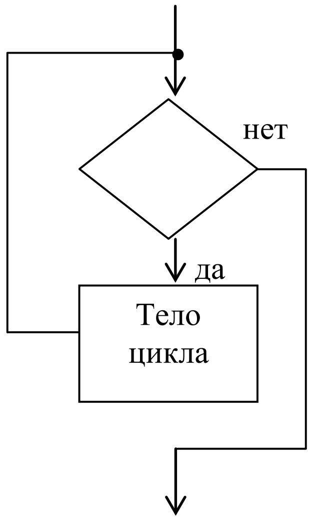


Рисунок 3.5 – Цикл с предусловием

Условие выхода из цикла можно поставить в конце, после тела цикла. Такой цикл называется «***циклом с постусловием***». Цикл выполняется, пока условие имеет значение «ложь». Как только условие примет значение «истина», выполнение цикла закончится. В этом случае условие является *условием завершения «цикла»* (рисунок 3.6).

Цикл с постусловием, в отличие от цикла с предусловием, выполняется обязательно как минимум один раз, независимо от того, выполняется условие или нет.

повторять

<тело цикла>

до тех пор, пока не выполнится <условие>

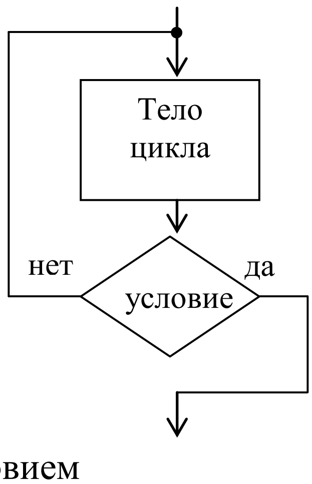


Рисунок 3.6 – Цикл с постусловием

ПРИЛОЖЕНИЯ

Приложение А

Пример А1

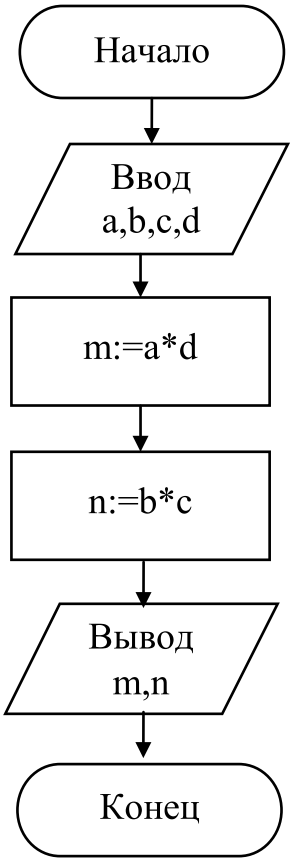
Даны две простые дроби. Составить алгоритм получения дроби, являющейся результатом их деления.

Решение представлено на рисунке А1.

В алгебраической форме решение задачи выглядит следующим образом:

$$\frac{a}{b} \div \frac{c}{d} = \frac{a * d}{b * c} = \frac{m}{n}.$$

Исходными данными являются четыре целые величины: a, b, c, d. Результат – два целых числа m и n.



алг Деление дробей
вещ a, b, c, m, m, d
нач **ввод** a, b, c, d
m:=a*d
n:=b*c
вывод “Числитель”, m
вывод “Знаменатель”, n
кон

Рисунок А1 – Блок-схема решения примера А1

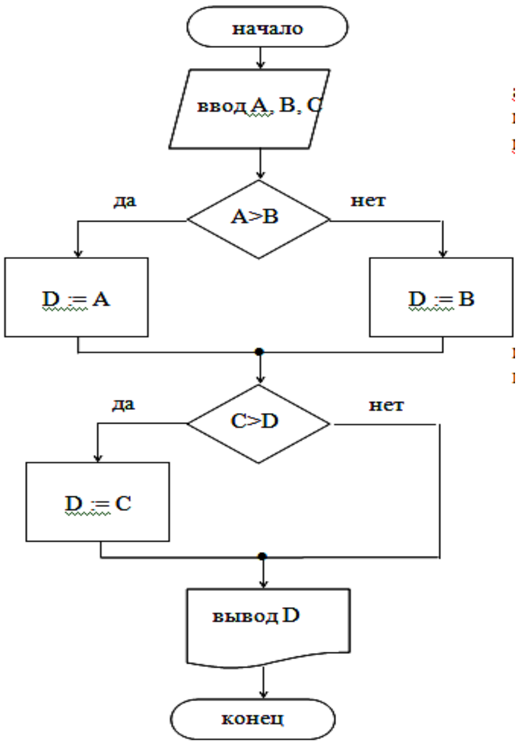
Тест: a = 3, b=4, c=5, d=6. Результат: m = 18, n = 20.

Пример А2

Даны три вещественных числа A, B, C. Найти наибольшее среди них.

Решение представлено на рисунке А2.

Сначала определяется большее среди двух значений A и B, затем большее между найденным значением и величиной C. Алгоритм имеет структуру двух последовательных ветвлений.




```
алг  БИТ1
вещ  A, B, C, D
нач  ввод A, B, C
       если A>B
           то  D:=A
           иначе D:=B
       кв
       если C > D
           то  D:=C
       кв
вывод D
кон
```

Рисунок А2 – Блок-схема и программа решения примера А2

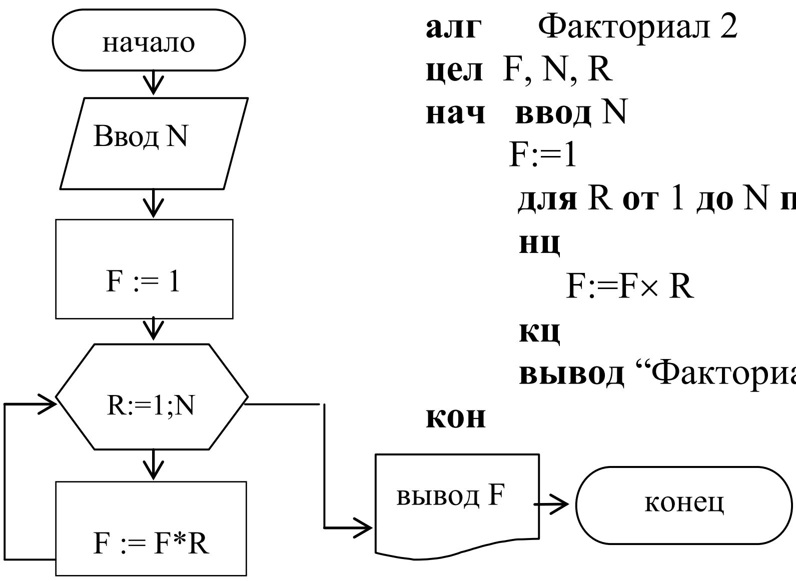
Тест: A = *3,* B = *6,* C = *2. Результат:* D = *6.*

Пример А3

Дано целое положительное число N. Вычислить факториал этого числа: N! = 1 * 2 * 3 * ... * N

Решение представлено на рисунке А3.

Задача решается с помощью циклического алгоритма: цикла с параметром.




```
алг     Факториал 2
цел  F, N, R
нач  ввод N
     F:=1
     для R от 1 до N повторять
     нц
         F:=F× R
     кц
     вывод “Факториал=”, F
кон
```

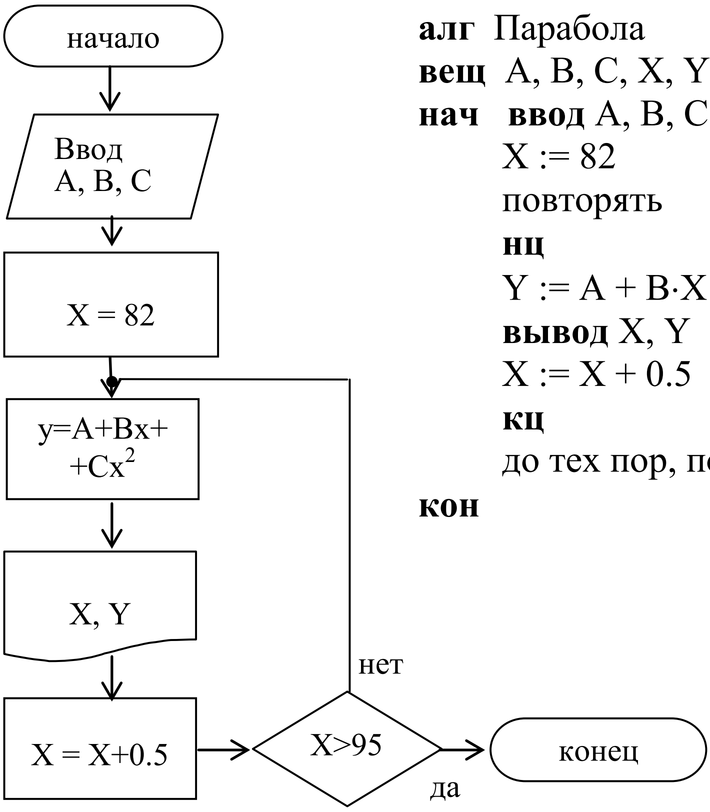
Рисунок А3 – Блок-схема решения примера А3

Тест: N = *5. Результат: факториал* F = *120.*

Пример А4

Вычислить значение функции Y = A = BX + CX^2 для $X \in [82; 95]$ изменяющимся с шагом 0,5.

Решение отображено на рисунке А4.



алг Парабола
вещ A, B, C, X, Y
нач **ввод** A, B, C
X := 82
повторять
нц
Y := A + B·X + C·X^2
вывод X, Y
X := X + 0.5
кц
до тех пор, пока не выполнится X>95
кон

Рисунок А4 – Блок-схема и программа примера А4

Пример А5

Использование вложенных циклов на примере вычисления двойной суммы

$$C = \sum_{n=1}^{8} \sum_{k=1}^{9} A_{n\ k} B_k^2,$$

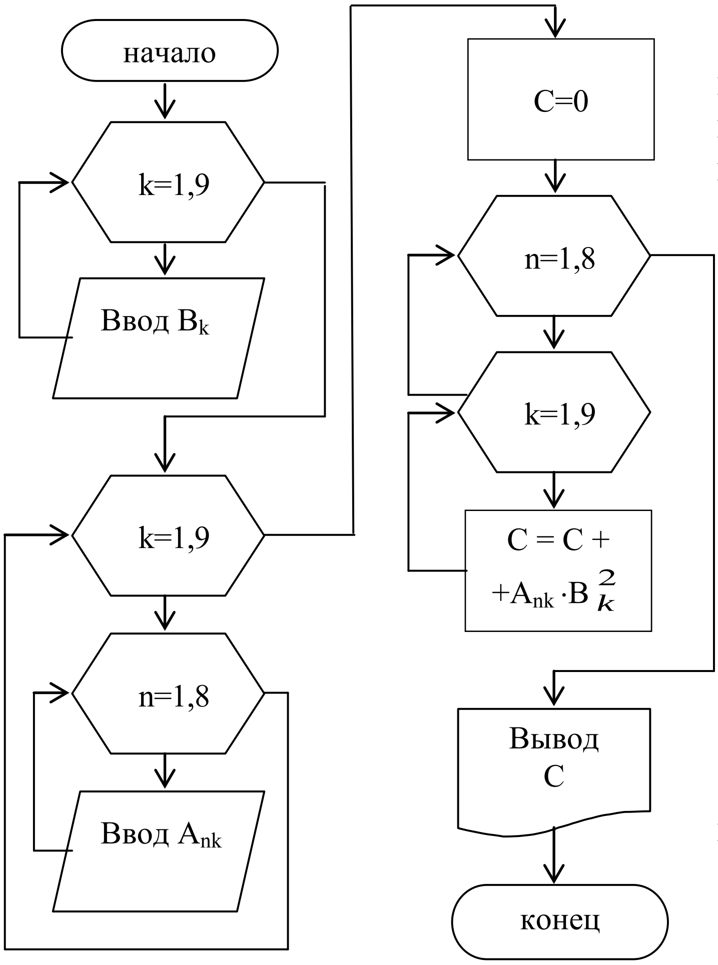
где A= $\begin{pmatrix} a_{1\,1} & a_{1\,2} & \dots & a_{1k} \\ a_{2\,1} & a_{2\,2} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{n\,k} \end{pmatrix}$ – матрица или двумерный массив данных одного и того же типа.

B = (b_1, b_2, b_3,…, b_k) – одномерный массив (вектор) данных одного и того же типа.

Решение представлено на рисунке А5.

Замечание. В соответствии с ГОСТ 19.701-90 границы цикла можно изображать так как они отображены на рисунке А6.

Начало цикла и конец цикла – оба символа имеют один и тот же идентификатор. Условия инициализации, приращения, завершения и т.д. помещаются внутри графического символа в начале или в конце в зависимости от расположения операции, проверяющей условие.



алг Массив
вещ B_k, A_{nk}, C
цел k, n
нач для k от 1 до 9 повторять
нц ввод b_k
кц
для k от 1 до 9 повторять
для n от 1 до 8 повторять
нц ввод A_{nk}
кц
C:=0
для n от 1 до 8 повторять
нц
для k от 1 до 9 повторять
нц $C:=C+A_{nk} \cdot B_k^2$
кц
кц
вывод C
кон

Рисунок А5 – Вложенные циклы

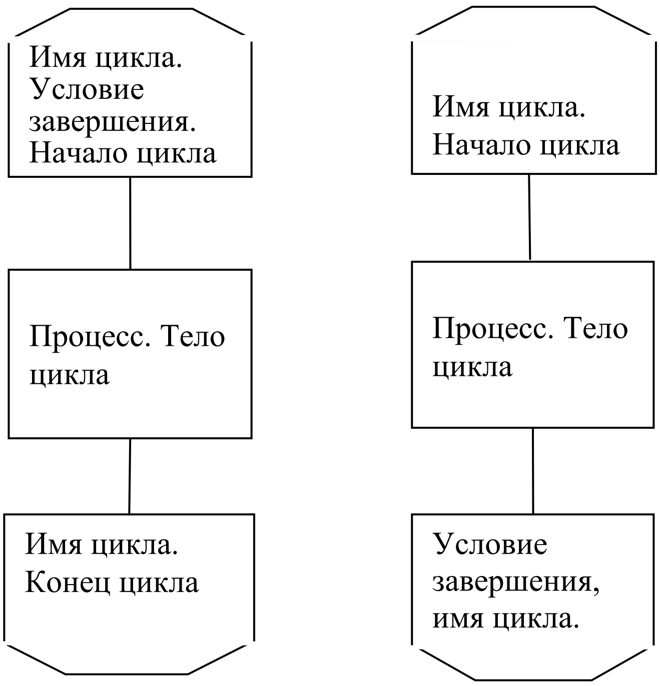


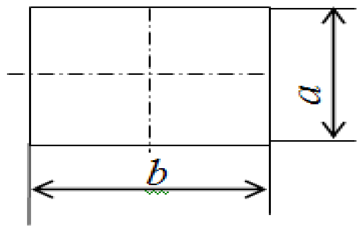
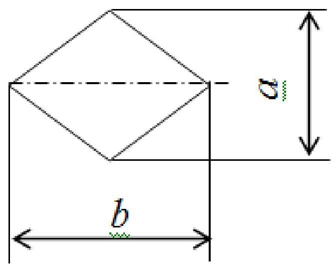
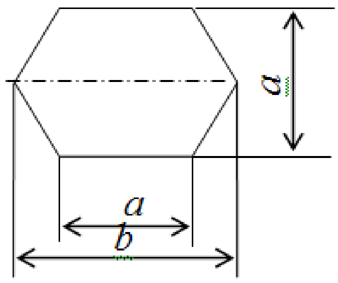
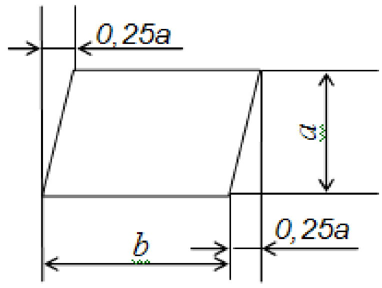
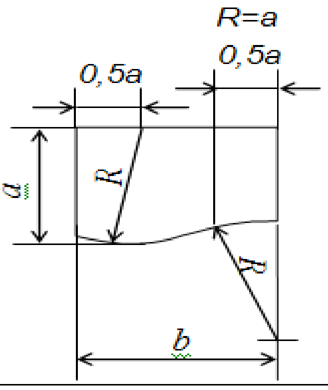
Рисунок А6 – Образец изображения границ цикла

Приложение Б

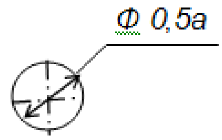
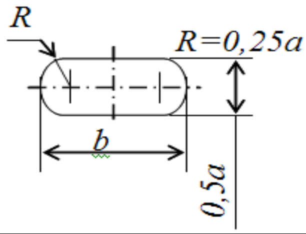
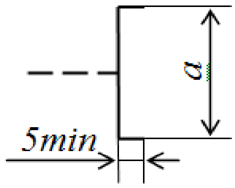
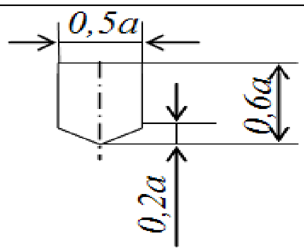
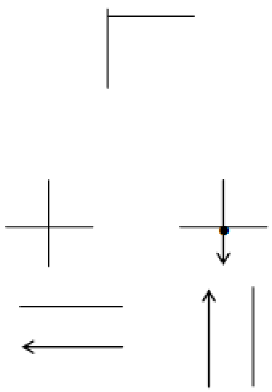
Фрагменты ГОСТ 19.003-80 Перечень, наименование, обозначение символов и отображаемые ими функции

1 Перечень, наименование, обозначение и размеры обязательных символов и отображаемые ими функции в алгоритме и программе обработки данных должны соответствовать параметрам, указанным в таблице Б1.

Таблица Б1 – Фрагменты ГОСТ 19.003-80

Фрагмент схемы	Содержание обозначения	Правила применения
1 Процесс		Выполнение операций, в результате которых изменяется значение, форма представления или расположение данных
2 Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий
3 Модификация		Выполнение операций, меняющих команды или группы команд, изменяющих программу
4 Ввод – вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод)
5 Документ		Ввод – вывод данных, носителем которых служит бумага

Продолжение таблицы Б1

6 Соединитель		Указание связи между прерванными линиями потока, связывающими символы
7 Пуск-остановка		Начало, конец, прерывание процесса обработки данных выполнения программы
8 Комментарий		Связь между элементами схемы и пояснением. Пунктирные линии в символе комментария могут обводить группу графических символов. Текст пояснений должен быть помешен около ограничивающей фигуры
9 Межстраничный соединитель		Указание связи между разъединенными частями схем алгоритмов и программ, расположенных на разных листах
10 Линии потока. Излом линий потока изображают под углом 90°. Место слияний линий потока обозначается точкой или цифрой 0		Применяют для указания направления линии потока: можно без стрелки, если линия направлена слева направо и сверху вниз; со стрелкой – в остальных случаях. Слияние линий потока: каждая из них направлена к одному и тому же символу на схеме.

2 Соотношение геометрических элементов символов

Размер *a* должен выбираться из ряда 10, 15, 20 мм. Допускается увеличивать размер *a* на число, кратное 5. Размер *b* равен 1,5*a*.

Список литературы

1 Венцель, Е. С. Теория вероятности [Текст] / Е. С. Венцель. – М. : Высшая школа, 1999. – 575 с.
2 Информатика. Базовый курс / под ред. С. В. Симоновича. – 2-е изд. – СПб. : Питер, 2013. – 640 с.
3 Угринович, Н. Информатика и информационные технологии [Текст] / Н. Угринович. – М. : Бином, Лаборатория знаний, 2003. – 512 с.
4 Семакин, И. Г. Информатика. Структурированный конспект базового курса [Текст] / И. Г. Семакин, Г. С. Вараксин. – М. : Лаборатория базовых знаний, 2004. – 168 с.
5 Семакин, И. Г. Информатика. [Текст] : Задачник-практикум / И. Г. Семакин, Е. Хеннер. – Т. 1. – М. : Лаборатория базовых знаний, 2002. – 304 с.
6 ГОСТ 19.003-80. Единая система программной документации. Схемы алгоритмов и программ. Обозначения условные графические. (Утратил силу).

Содержание

Сысолятина Лидия Геннадьевна
Котликова Вера Яковлевна
Бекишева Марина Борисовна

ВВЕДЕНИЕ В ИНФОРМАТИКУ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Часть 1

Методические указания
к выполнению лабораторных работ
по дисциплинам «Информатика», «Информационные технологии»
для студентов очной и заочной формы обучения
направлений 040400.62, 030900.62, 040100.62, 190700.62, 140400.62, 190600.62, 190109.65, 190110.65, 151900.62, 150700.62, 220700.62, 220400.62, 280700.62, 221700.62

Редактор Е. А. Могутова

…………………………………………………………………………………

Подписано в печать 10.07.14	Формат 60*84 1/16.	Бумага 65 г/м^2
Печать цифровая	Усл. печ. л. 2,25	Уч.-изд. л. 2,25
Заказ 210	Тираж 25	Не для продажи

…………………………………………………………………………………

РИЦ Курганского государственного университета.
640000, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.