

1 ПОНЯТИЕ ИНФОРМАЦИИ

Определение	С точки зрения информатики, <i>информация</i> — это связанные между собой сведения, изменяющие наши представления о явлении или объекте окружающего мира										
Свойства	Понятность	Этим свойством обладает только та информация, которая выражена в форме, понятной тем, кому она предназначена, в противном случае информация становится бесполезной									
	Полезность (ценность)	Это комплексный показатель качества информации. Зависит от того, какие задачи можно решить, используя эту информацию. Однакая ценность информации — это понятие субъективное, т. к. информация, полезная для одного человека, может быть совершенно бесполезной для другого									
	Достоверность	Информация достоверна, если она содержит сведения, отражающие истинное положение дел. Часто из-за искажений информации это свойство утрачивается. Кроме того, достоверная информация со временем может стать недостоверной, так как она обладает свойством устаревать									
	Актуальность	Определяется степенью сохранения ценности информации в момент ее использования. Актуальную информацию очень важно иметь при работе в изменяющихся условиях, т. к. в таком случае только вовремя полученная (или обновленная) информация может принести пользу (примером может служить прогноз погоды)									
	Полнота и точность	Полнота информации означает, что она содержит минимальный, но достаточный для принятия правильного решения состав. Нарушение полноты информации сдерживает принятие решений и может повлечь ошибки									
Формы	числовая, текстовая, графическая, звуковая, видеoinформация										
Единицы	Бит, байт, Килобайт, Мегабайт, Гигабайт, Терабайт и т.д. Бит – это минимальная единица измерения информации, соответствующая одной двоичной цифре («0» или «1»). $2^3=8$ $2^{10}=1024$ $2^{10}=1024$ $2^{10}=1024$ бит → байт → Кбайт → Мбайт → Гбайт										
Информационные процессы	— получение информации; — хранение информации; — обработка информации; — передача информации										
Кодирование звука	$V = B \cdot f \cdot t$, где V – размер цифрового моноаудиофайла (бит, байт и т.д), t – время (с), f – частота дискретизации (Гц), B – глубина кодирования (бит). <i>При двухканальной записи (стерео) объем памяти, необходимый для хранения данных одного канала, умножается на 2</i>										
Информационный объем сообщения	$I = N \cdot K$, где V – объем текста, N – количество символов (отсчетов), K – число бит на символ (отсчет). $Q = M^N$, Q – количество всех возможных «слов» (символьных цепочек), M – мощность алфавита (количество символов в этом алфавите)										
Определение скорости передачи информации	$V = q \cdot t$, V – объем переданной информации, q – пропускная способность канала (в битах в секунду или подобных единицах), t – время передачи.										
Компьютерные сети. Адресация в Интернете	IP-адрес состоит из четырех чисел, разделенных точками; каждое из этих чисел находится в интервале 0...255, например: 192.168.85.210 Адрес документа в Интернете (URL = <i>Uniform Resource Locator</i>) состоит из следующих частей: — протокол, чаще всего http (для Web-страниц) или ftp (для файловых архивов); — знаки //, отделяющие протокол от остальной части адреса; — доменное имя (или IP-адрес) сайта; — каталог на сервере, где находится файл; — имя файла										

Таблица степеней двойки

К, бит	0	1	2	3	4	5	6	7	8	9	10
Q, вариантов	1	2	4	8	16	32	64	128	256	512	1024

2 ЭЛЕКТРОННЫЕ ТАБЛИЦЫ EXCEL

2.1 Встроенные функции Excel

Функции	Вид записи	Назначение
Математические	КОРЕНЬ(...)	Вычисление квадратного корня
	ABS(...)	Вычисление абсолютного значения (модуля) числа
	ОКРУГЛ(...)	Округляет число до указанного количества десятичных разрядов
	ПИ() *	Значение математической константы «ПИ» (3,1415926...)
	НОД(...)	Наибольший общий делитель нескольких чисел
	НОК(...)	Наименьшее общее кратное нескольких чисел
Статистические	СЛУСИС() *	Вычисление случайного числа в промежутке между 0 и 1
	МИН(...)	Определение минимального из указанных чисел
	МАКС(...)	Определение максимального из указанных чисел
	СРЗНАЧ(...)	Определение среднего значения указанных чисел
Логические	СУММ(...)	Определение суммы указанных чисел
	И(условие1; условие2;...)	Вычисление значения (ИСТИНА, ЛОЖЬ) логической операции И
	ИЛИ(условие1; условие2;...)	Вычисление значения (ИСТИНА, ЛОЖЬ) логической операции ИЛИ
	ЕСЛИ(условие; знач ИСТИНА; знач ЛОЖЬ)	Вычисление значения в зависимости от выполнения условия

* Записывается без аргументов

2.2 Виды ссылок

Название	Запись	При копировании
Относительная	C3	Меняется в соответствии с новым положением ячейки
Абсолютная	\$C\$3	Не меняется
Смешанная	C\$3	Не меняется номер строки
	\$C3	Не меняется имя столбца

2.3 Операторы

Арифметические операторы			Операторы сравнения		
Оператор	Значение	Пример	Оператор	Значение	Пример
+ (знак плюс)	Сложение	A1+A2	= (знак равенства)	Равно	(A1=B1)
- (знак минус)	Вычитание Отрицание	A1-A2 -A1	> (знак больше)	Больше	(A1>B1)
* (звездочка)	Умножение	A1*A2	< (знак меньше)	Меньше	(A1<B1)
/ (косая черта)	Деление	A1/A2	>= (знак больше и равно)	Больше или равно	(A1>=B1)
% (знак процента)	Процент	50%	<= (знак меньше и равно)	Меньше или равно	(A1<=B1)
^ (крышка)	Возведение в степень	A1^2	<> (знак меньше и больше)	Не равно	(A1<>B1)
Текстовый оператор конкатенации					
& (амперсанд)	Объединение последовательностей знаков в одну последовательность		«Фамилия»&«Имя»&«Отчество»		
Операторы ссылок					
: (двоеточие)	Ставится между ссылками на первую и последнюю ячейки диапазона		B5:B15		
; (точка с запятой)	Оператор объединения		B5:B15;D5:D15		
(пробел)	Оператор пересечения множеств, служит для ссылки на общие ячейки двух диапазонов		B7:D7 C6:C8		

3 СИСТЕМЫ СЧИСЛЕНИЯ

Система счисления – способ наименования и представления чисел с помощью символов. Такие символы в любой системе счисления называются цифрами.

Алфавит системы счисления – это совокупность символов, используемых в данной системе счисления.

Система счисления	Основание	Алфавит цифр
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двоичная	2	0, 1
Восьмеричная	8	0, 1, 2, 3, 4, 5, 6, 7
Шестнадцатеричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15)

Для восьмеричной системы:

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

Для шестнадцатеричной системы:

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

3.1 Перевод целых чисел из одной системы счисления в другую (на примере числа 46)

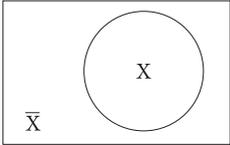
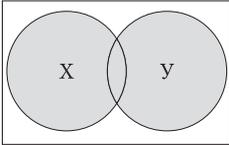
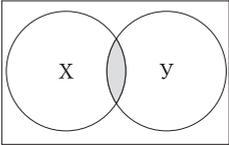
Перевод из 10-ой системы в другие		Перевод из 2-ой системы в другие	
10 → 2	$\begin{array}{r} 46 \mid 2 \\ 0 \ 23 \mid 2 \\ \swarrow \quad \downarrow \\ 1 \ 11 \mid 2 \\ \swarrow \quad \downarrow \\ 1 \ 5 \mid 2 \\ \swarrow \quad \downarrow \\ 1 \ 2 \mid 2 \\ \swarrow \quad \downarrow \\ 0 \ 1 \end{array}$ <p>Ответ: 101110₂</p>	2 → 8	<p>(разбиваем число на двоичные триады справа налево и дополняем слева нолями, при необходимости)</p> $101110_2 = \underbrace{101}_5 \underbrace{110}_6 = 56_8$ <p>Ответ: 56₈</p>
10 → 8	$\begin{array}{r} 46 \mid 8 \\ \swarrow \quad \downarrow \\ 6 \ 5 \end{array}$ <p>Ответ: 56₈</p>	2 → 10	$101110 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 46_{10}$ <p>Ответ: 46₁₀</p>
10 → 16	$\begin{array}{r} 46 \mid 16 \\ \swarrow \quad \downarrow \\ 14 \ 2 \end{array}$ <p>Ответ: 2E₁₆</p>	2 → 16	<p>(разбиваем число на двоичные тетрады справа налево и дополняем слева нолями, при необходимости)</p> $101110_2 = \underbrace{0010}_2 \underbrace{1110}_E = 2E_{16}$ <p>Ответ: 2E₁₆</p>
Перевод из 8-ой системы в другие		Перевод из 16-ой системы в другие	
8 → 2	$56_8 = 101110_2$ <p>Ответ: 101110₂</p>	16 → 2	$2E_{16} = \underbrace{0010}_2 \underbrace{1110}_E = 101110_2$ <p>Ответ: 101110₂</p>
8 → 10	$56_8 = 5 \cdot 8^1 + 6 \cdot 8^0 = 46_{10}$ <p>Ответ: 46₁₀</p>	16 → 8	$2E_{16} = 101110_2 = \underbrace{101}_5 \underbrace{110}_6 = 56_8$ <p>Ответ: 56₈</p>
8 → 16	$56_8 = \underbrace{101}_5 \underbrace{110}_6 = \underbrace{0010}_2 \underbrace{1110}_E = 2E_{16}$ <p>Ответ: 2E₁₆</p>	16 → 10	$2E_{16} = 2 \cdot 16^1 + E \cdot 16^0 = 32 + 14 = 46_{10}$ <p>Ответ: 46₁₀</p>

4 ОСНОВЫ ЛОГИКИ

4.1 Таблицы истинности логических функций

Таблицей истинности логической функции принято называть табличное представление логической операции, в котором присутствуют все возможные сочетания значений входных переменных и получаемые при этом значения выходных переменных (результатов логической операции).

Таблица истинности функции логического

отрицания (инверсии) (операция НЕ – NOT)	сложения (дизъюнкции) (операция ИЛИ – OR)	умножения (конъюнкции) (операция И – AND)																																				
<table border="1"> <tr><td>X</td><td>\bar{X}</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	X	\bar{X}	0	1	1	0	<table border="1"> <tr><td>X</td><td>Y</td><td>$X = X \vee Y$</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	$X = X \vee Y$	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>X</td><td>Y</td><td>$X = X \& Y$</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	$X = X \& Y$	0	0	0	0	1	0	1	0	0	1	1	1
X	\bar{X}																																					
0	1																																					
1	0																																					
X	Y	$X = X \vee Y$																																				
0	0	0																																				
0	1	1																																				
1	0	1																																				
1	1	1																																				
X	Y	$X = X \& Y$																																				
0	0	0																																				
0	1	0																																				
1	0	0																																				
1	1	1																																				
																																						

где X, Y – значения аргументов; 0 – тождественно ложное высказывание; 1 – тождественно истинное высказывание

Свойства

дизъюнкции

$$\begin{aligned}
 X \vee Y &\equiv Y \vee X; \\
 (X \vee Y) \vee Z &\equiv X \vee (Y \vee Z); \\
 X \vee 0 &\equiv X; \\
 X \vee \bar{X} &\equiv 1; \\
 X \vee 1 &\equiv 1; \\
 X \vee X &\equiv X
 \end{aligned}$$

конъюнкции

$$\begin{aligned}
 X \& X &\equiv X; \\
 X \& Y &\equiv Y \& X; \\
 (X \& Y) \& Z &\equiv X \& (Y \& Z); \\
 X \& \bar{X} &\equiv 0; \\
 X \& 0 &\equiv 0; \\
 X \& 1 &\equiv X
 \end{aligned}$$

4.2 Основные законы алгебры логики

Закон непротиворечия	$A \& \bar{A} = 0$		
Закон исключенного третьего	$A \vee \bar{A} = 1$		
Законы де Моргана	$\overline{A \vee B} = \bar{A} \& \bar{B}, \overline{A \& B} = \bar{A} \vee \bar{B}$		
Закон двойного отрицания	$A = \overline{\bar{A}}$		
Закон коммутативности	$A \& B = B \& A,$ $A \vee B = B \vee A$		
Закон ассоциативности	$(A \& B) \& C = A \& (B \& C),$ $(A \vee B) \vee C = A \vee (B \vee C)$		
Закон дистрибутивности	За скобки можно выносить как общие множители, так и общие слагаемые: $(A \& B) \vee (A \& C) = A \& (B \vee C),$ $(A \vee B) \& (A \vee C) = A \vee (B \& C)$		
Закон поглощения	$A \vee (A \& B) = A,$ $A \& (A \vee B) = A$		
Закон исключения (склеивания)	$(A \& B) \vee (\bar{A} \& B) = B,$ $(A \vee B) \& (\bar{A} \vee B) = B$		
$A \vee A = A$ $A \vee 0 = A$	$A \& A = A$ $A \& 1 = A$	$A \vee 1 = 1$ $A \& 0 = 0$	$A \rightarrow B = \bar{A} \vee B$ $A \leftrightarrow B = \bar{A} \& \bar{B} \vee A \& B$

5 АЛГОРИТМЫ

5.1 Понятие алгоритма и его свойства

Алгоритм — это понятное и точное указание исполнителю совершить последовательность действий, направленных на решение поставленной задачи

Свойства алгоритма

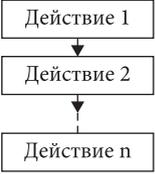
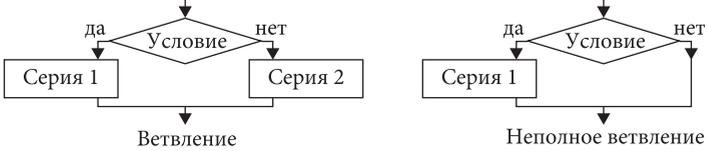
Дискретность	Разделение информационного процесса в алгоритме на отдельные команды
Понятность	Каждый алгоритм должен состояться в расчете на конкретного исполнителя с учетом его возможностей. У каждого исполнителя имеется перечень команд, которые он может исполнить. Такой перечень (список) называется системой команд исполнителя (СКИ). Понятными исполнителю будут являться только те команды, которые попадают в этот список
Определенность (детерминированность)	Команды, образующие алгоритм, должны быть предельно четкими и однозначными. Их результат не может зависеть от какой-либо дополнительной информации извне алгоритма. При наличии ошибок в алгоритме это свойство может иногда нарушаться
Результативность	Результат выполнения алгоритма должен быть обязательно получен, т.е. правильный алгоритм не может обрываться безрезультатно из-за какого-либо непреодолимого препятствия в ходе выполнения. Кроме того, любой алгоритм должен завершиться за конечное число шагов. Большинство алгоритмов данным требованиям удовлетворяют, но при наличии ошибок возможны нарушения результативности
Корректность	Любой алгоритм создан для решения той или иной задачи, поэтому необходима уверенность, что это решение будет правильным для любых допустимых исходных данных
Массовость	Алгоритм имеет смысл разрабатывать только в том случае, когда он будет применяться многократно для различных наборов исходных данных. Массовость алгоритма в отдельных случаях может нарушаться: к числу подобных исключений можно отнести алгоритмы пользования некоторыми простыми автоматами (для них входными данными служит единственный тип монет)

5.2 Условные обозначения в блок-схемах алгоритмов

Схема алгоритма представляет собой систему связанных геометрических фигур, каждая из которых обозначает один этап процесса решения задачи и называется блоком. Порядок выполнения блоков указывается стрелками, соединяющими блоки. В схеме блоки размещаются сверху вниз в порядке их выполнения

Наименование	Обозначение	Комментарии
Пуск-останов		Начало или конец алгоритма, вход/выход в подпрограмму
Процесс		Действия, вычисления или группа операций
Предопределенный процесс (вспомогательный алгоритм)		Использование ранее разработанного алгоритма как составную часть данного алгоритма
Принятие решения		Разветвление в алгоритме, проверка условия
Модификация (заголовок цикла)		Многократное выполнение некоторой последовательности действий, которая называется телом цикла
Ввод/вывод		Ввод/вывод данных в общем виде
Комментарии		Комментарии

5.3 Базовые алгоритмические структуры

<p>Линейная структура</p>	<p>Разветвляющаяся структура (альтернатива «если-то-иначе» или «если-то»)</p>	
<p>действия выполняются друг за другом</p> 	<p>обеспечивает альтернативный выбор в зависимости от заданного условия</p> 	
<p>Циклическая структура (или повторение)</p>		
<p>предусматривает повторное выполнение некоторого набора действий</p>		
<p>цикл «пока» (с предусловием)</p> 	<p>цикл «до» (с постусловием)</p> 	<p>цикл с параметром</p> 

5.4 Основные понятия алгоритмического программирования

<p>Данные</p>	<p>Величины, обрабатываемые программой \Leftrightarrow переменные константы</p>	
<p>Имена (идентификаторы)</p>	<p>Используются для обозначения объектов в программе (переменных, массивов, функций и др.). Каждая переменная имеет свое уникальное имя и характеризуется идентификатором, типом и значением</p>	
<p>Операции</p>	<ul style="list-style-type: none"> — арифметические операции: +, -, *, / и др. ; — логические операции: и, или, не; — операции отношения: < (меньше), > (больше), <= (меньше или равно), >= (больше или равно), = (равно), <> (не равно); — операция сцепки («присоединения», «конкатенации») символьных значений друг с другом с образованием одной длинной строки; изображается знаком «+» 	
<p>Выражения</p>	<p>Предназначаются для выполнения необходимых вычислений, состоят из констант, переменных, указателей функций (например, exp(x)), объединенных знаками операций</p>	
<p>Операторы (команды)</p>	<p>Как правило, оператор содержит имя и данные и указывает, какую операцию и над какими величинами надо выполнить. Одна строка программы может содержать один или несколько операторов. В состав операторов входят:</p> <ul style="list-style-type: none"> — ключевые слова; — данные; — выражения и т. д. 	
<p>Функции</p>	<p>Для наиболее употребительных функций программы их вычисления записаны в память компьютера, в библиотеки программ, а сами функции включены в состав языков программирования. Такие функции называются встроенными (или стандартными). Для вычисления таких функций в программе достаточно указать имя функции и значение ее аргумента. Каждый язык программирования имеет свой набор стандартных функций</p>	
<p>Процедуры</p>	<p>Процедура — это самостоятельная программная единица, которая выполняется по команде из другой программы или процедуры. Процедура оформляется определенным образом, к ней можно обращаться из разных точек программы любое число раз. При этом такая процедура может решать каждый раз одну и ту же задачу с разными значениями исходных данных</p>	

5.5 Операторы языков Бейсик, Паскаль и Си

Бейсик	Паскаль	Си
Описание массива		
DIM <имя массива> <максимальные размеры индексов>	<имя массива> :array <мин.индекс>..<макс.индекс> of <тип>;	[класс_памяти] <тип элемента> <имя> [размер]...;
DIM A(10)	a:array[1.. 10] of real;	int a[10];
Оператор присваивания		
<имя переменной> = <выражение>	<имя переменной> := <выражение>;	<имя переменной> = <выражение>;
X = 5	x := 5;	x = 5;
Условный оператор		
используется 2 варианта: 1) оператор «в строчку»: IF <условие> THEN <операторы «да»> [ELSE <операторы «нет»>;] 2) оператор «в столбец»: IF <условие> THEN <операторы «да»> [ELSE <операторы «нет»>] ENDIF	if <условие> then <оператор «да»> [else <оператор «нет»>];	if (условие) <оператор «да»>; else <оператор «нет»>;
IF A <> 0 THEN A = 0 ELSE A = -1 ENDIF	if a <> 0 then a := 0 else a := -1;	if(a <> 0) a = 0; else a = -1;
Операторы цикла		
1) цикл с предусловием (цикл «до»)		
WHILE <условие цикла> <операторы в цикле> WEND	While <условие> do <оператор в цикле>;	while (условие) <оператор в цикле>;
WHILE X < 100 X = X + 1 WEND	While x < 100 do x := x + 1;	while (x < 100) x = x + 1;
2) цикл с постусловием (цикл «пока»)		
DO <оператор цикла> LOOP UNTIL <условие>	Repeat <операторы цикла>; Loop Until <условие>;	do <оператор цикла>; while (условие);
DO X = X + 1 LOOP UNTIL X >=100	Repeat x := x + 1; Loop Until x >=100;	do x = x + 1; while (x >=100);
3) цикл с параметром		
FOR <упр.> = <н.з.> TO <к.з.> STEP <шаг> <операторы цикла> NEXT [<упр.>], где упр. – управляющая переменная, у которой имеется начальное (н.з.) и конечное (к.з.) значение.	1) for <упр.> := <н.з.> to <к.з.> do <оператор>; 2) for <упр.> := <н.з.> Downto <к.з.> do <оператор>;	for (инициализация_цикла; выражение; реинициализация_цикла) <оператор>;
FOR I = 1 TO 5 STEP 1 F = F + I*2 NEXT I	for i = 1 to 5 do f := f + i*2;	for (i = 1; i <= 5; i ++) f = f + i*2;
Оператор безусловного перехода		
GOTO <метка>	GOTO <метка>;	goto <метка>;
GOTO K	goto k;	goto k;
Оператор ввода-вывода		
<i>ввод:</i> INPUT [<пояснительный текст>], <список вводимых переменных> <i>вывод:</i> PRINT <список выводимых величин>	<i>ввод:</i> Read (<список вводимых переменных>;) <i>вывод:</i> Write (<список выводимых величин>;)	<i>ввод:</i> printf (<форматная_строка>, <список_вывода>;) <i>вывод:</i> scanf (<форматная_строка>, <список_ввода>;)
<i>Пример вывода на экран — «Введите число:(курсор ввода числа)»</i>		
PRINT «Введите число: « INPUT A	Write («Введите число: »); Read (a);	printf («Введите число: »); scanf (a);