

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиозлектроники

Е.С. Шандаров

ТЕОРИЯ ИНФОРМАЦИИ И ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания к лабораторным работам,
практическим занятиям и самостоятельной работе студентов

Томск
2024

УДК 519.72
ББК 32.811
Ш-20

Шандаров, Евгений Станиславович

Ш-20 Теория информации и информационных систем: методические указания к лабораторным работам, практическим занятиям и самостоятельной работе студентов/ Е.С. Шандаров. - Томск: Томский государственный университет систем управления и радиоэлектроники, 2024. - 41 с.

Методические указания предназначены для студентов, изучающих вопросы теории информации и информационных систем. Издание содержит все материалы, необходимые студентам для выполнения лабораторных работ, подготовки к практическим занятиям, а также для организации самостоятельной работы: описание теоретических сведений о способах кодирования и представления информации, описание заданий порядок выполнения лабораторных работ, задачи для практических работ, образец титульного листа и рекомендуемую структуру отчета по лабораторным работам.

Предназначено для студентов вузов, обучающихся по техническим специальностям.

Одобрено на заседании кафедры электронных приборов (ЭП), протокол № 05-24 от 08.05.2024 г.

УДК 519.72
ББК 32.811
© Шандаров Е. С., 2024
© Томский государственный университет
систем управления и радиоэлектроники, 2024

Оглавление

Введение.....	4
1 Теоретическая часть.....	5
1.1 Инструменты необходимые для выполнения лабораторных работ.....	5
1.2 Кодирование информации в компьютере.....	6
2 Лабораторные работы.....	13
2.1 Лабораторная работа №1. Исследование статистических характеристик текстового документа.....	13
2.2 Лабораторная работа №2. Работа с кодовыми таблицами русского языка.....	14
2.3 Лабораторная работа №3. Основы криптографии.....	15
2.4 Лабораторная работа №4. Изучение языка гипертекстовой разметки HTML.....	16
2.5 Отчет по лабораторным работам.....	32
3 Практические работы.....	33
3.1 Основные понятия теории информации. Количество информации. Энтропия Шеннона.....	33
3.2 Передача информации по каналам связи.....	35
3.3 Кодирование информации.....	37
Список рекомендованной литературы.....	40
Приложение А.....	41

Введение

Дисциплина “Теория информации и информационных систем” направлена на изучение таких разделов теории информации, как кодирование, количество информации, сжатие данных, передача информации по каналам связи, помехозащитное кодирование, криптография [1-3].

Цель дисциплины состоит в формировании у студентов понимания фундаментальных основ теории информации, методов кодирования и информационных систем.

Задачи изучения дисциплины определены как:

- Изучение методов кодирования информации.
- Формирование у студентов общих представлений о современных методах передачи, преобразования и приема информации в компьютерных и телекоммуникационных системах.
- Изучение методов сжатия информации.
- Овладение методами оптимального и помехоустойчивого кодирования в системах передачи и обработки информации.

Изучение данной дисциплины позволит студентам научиться решать задачи выбора оптимального способа кодирования данных для эффективного хранения и передачи по каналам связи, размещению информации в сети Интернет.

Данное издание содержит методические указания к выполнению лабораторных и практических работ по дисциплине «Теория информации и информационных систем».

В издании кратко изложена теория, необходимая для самостоятельного изучения и повторения материала, а также подготовки к практическим занятиям, даны задания для лабораторных и практических работ, а также рекомендации по оформлению отчетов по лабораторным работам.

Лабораторные работы по курсу проводятся с использованием программного обеспечения бесплатно распространяемого пакета OpenOffice.org и программного продукта FreePascal, также бесплатно-распространяемого.

В рамках данного курса студенты создают программы на языке Pascal для реализации базовых методов теории информации и кодирования.

Предполагается, что обучающиеся имеют математическую подготовку и хорошо владеют языком программирования Паскаль.

1 Теоретическая часть

1.1 Инструменты необходимые для выполнения лабораторных работ

Free Pascal (FPC) компилятор с открытыми исходными кодами с двумя важными особенностями: высокая совместимость с Delphi и доступность для большого количества платформ, включая Windows, Mac OS X, и Linux.

Совместимость Free Pascal с Delphi включает не только поддержку того же языка программирования Object Pascal который использует Delphi, но также библиотеки конструкции и классы, которые "понимает" Delphi, например System, SysUtils, StrUtils, DateUtils, Classes, Variants, Math, IniFiles and Registry, которые включены в Free Pascal и поддерживаются всеми платформами. Free Pascal также включает модули Windows, ShellAPI, BaseUnix, Unix и DynLibs для доступа к специфичным для конкретной ОС функциям. Около дюжины модулей составляют ядро обычно упоминающееся как Free Pascal run-time library (RTL).

Консольные приложения не имеют графического интерфейса (GUI). Обычно они запускаются в консоли, читают информацию с консоли и выводят результаты на консоль. В Windows консоли обычно соответствует окно командной строки. В OS X и Linux консоли соответствует окно терминала. Консольные приложения это, как правило, небольшие утилиты, такие как программа FC (file compare - сравнение файлов) в Windows или команды cd и cp в Unix. Консольные приложения могут быть также мощными программами для тестирования, моделирования или управления данными, которые не нуждаются в причудливом GUI, потому как они запускаются другими программами или из командных файлов. Компилятор Free Pascal и прилагающийся набор утилит являются консольными приложениями, то есть могут быть запущены в консоли, из командных файлов или из IDE Lazarus.

Для создания консольного приложения необходим только текстовый редактор и компилятор FreePascal.

1.2 Кодирование информации в компьютере

Кодовая таблица ASCII

ASCII — это таблица кодировки символов, в которой каждой букве, числу или знаку соответствует определенное число. В стандартной таблице ASCII 128 символов, пронумерованных от 0 до 127. В них входят латинские буквы, цифры, знаки препинания и управляющие символы.

DEC	HEX	CHAR
0	00	NUL (Null)
1	01	SOH (Start of Heading)
2	02	STX (Start of Text)
3	03	ETX (End of Text)
4	04	EOT (End of Transmission)
5	05	ENQ (Enquiry)
6	06	ACK (Acknowledgment)
7	07	BEL (Bell)
8	08	BS (Backspace)
9	09	TAB (Horizontal Tab)
10	0A	LF (Line Feed)
11	0B	VT (Vertical Tab)
12	0C	FF (Form Feed)
13	0D	CR (Carriage Return)
14	0E	SO (Shift Out)
15	0F	SI (Shift In)
16	10	DLE (Data Link Escape)
17	11	DC1 (Device Control 1)
18	12	DC2 (Device Control 2)
19	13	DC3 (Device Control 3)
20	14	DC4 (Device Control 4)
21	15	NAK (Negative Acknowledgement)
22	16	SYN (Synchronous Idle)
23	17	ETB (End of Transmission Block)
24	18	CAN (Cancel)
25	19	EM (End of Medium)
26	1A	SUB (Substitute)
27	1B	ESC (Escape)

28	1C	FS	(File Separator)
29	1D	GS	(Group Separator)
30	1E	RS	(Record Separator)
31	1F	US	(Unit Separator)
32	20	SPACE	(Space)
33	21	!	(Exclamation Mark)
34	22	"	(Double Quote)
35	23	#	(Number Sign)
36	24	\$	(Dollar Sign)
37	25	%	(Percent Sign)
38	26	&	(Ampersand)
39	27	'	(Single Quote)
40	28	((Left Parenthesis)
41	29)	(Right Parenthesis)
42	2A	*	(Asterisk)
43	2B	+	(Plus Sign)
44	2C	,	(Comma)
45	2D	-	(Hyphen-Minus)
46	2E	.	(Period)
47	2F	/	(Slash)
48	30	0	(Digit Zero)
49	31	1	(Digit One)
50	32	2	(Digit Two)
51	33	3	(Digit Three)
52	34	4	(Digit Four)
53	35	5	(Digit Five)
54	36	6	(Digit Six)
55	37	7	(Digit Seven)
56	38	8	(Digit Eight)
57	39	9	(Digit Nine)
58	3A	:	(Colon)
59	3B	;	(Semicolon)
60	3C	<	(Less-Than Sign)
61	3D	=	(Equal Sign)
62	3E	>	(Greater-Than Sign)
63	3F	?	(Question Mark)
64	40	@	(Commercial At)
65	41	A	

66	42	B	
67	43	C	
68	44	D	
69	45	E	
70	46	F	
71	47	G	
72	48	H	
73	49	I	
74	4A	J	
75	4B	K	
76	4C	L	
77	4D	M	
78	4E	N	
79	4F	O	
80	50	P	
81	51	Q	
82	52	R	
83	53	S	
84	54	T	
85	55	U	
86	56	V	
87	57	W	
88	58	X	
89	59	Y	
90	5A	Z	
91	5B	[(Left Square Bracket)
92	5C	\	(Backslash)
93	5D]	(Right Square Bracket)
94	5E	^	(Caret / Circumflex)
95	5F	_	(Underscore)
96	60	`	(Grave Accent)
97	61	a	
98	62	b	
99	63	c	
100	64	d	
101	65	e	
102	66	f	
103	67	g	

104	68	h	
105	69	i	
106	6A	j	
107	6B	k	
108	6C	l	
109	6D	m	
110	6E	n	
111	6F	o	
112	70	p	
113	71	q	
114	72	r	
115	73	s	
116	74	t	
117	75	u	
118	76	v	
119	77	w	
120	78	x	
121	79	y	
122	7A	z	
123	7B	{	(Left Curly Brace)
124	7C		(Vertical Bar)
125	7D	}	(Right Curly Brace)
126	7E	~	(Tilde)
127	7F	DEL	(Delete)

Кодировка букв русского алфавита

Кодировка букв русского алфавита может осуществляться несколькими способами:

- ISO 8859-5 — международный стандарт;
- Кодовая страница 866 (Microsoft CP866) — используется в MS-DOS;
- Кодовая страница 1251 (Microsoft CP1251) для Microsoft Windows;
- На базе ГОСТ КОИ-8, koi8-r — применяется в мире Unix;
- Unicode — используется в Microsoft Windows, Unix и клонах Unix.

Конечно, отсутствие строгой стандартизации не могло способствовать повышению эффективности. Тем не менее, это очень поучительный урок того, как делать не следует.

1	2	I		II		III		IV		V		VI
а	A	208	D0	160	A0	224	E0	193	C1	1072	0430	A
б	BE	209	D1	161	A1	225	E1	194	C2	1073	0431	B
в	VE	210	D2	162	A2	226	E2	215	D7	1074	0432	W
г	GHE	211	D3	163	A3	227	E3	199	C7	1075	0433	G
д	DE	212	D4	164	A4	228	E4	196	C4	1076	0434	D
е	IE	213	D5	165	A5	229	E5	197	C5	1077	0435	E
ё	IO	241	F1	241	F1	184	B8	163	A3	1105	0451	#
ж	ZHE	214	D6	166	A6	230	E6	214	D6	1078	0436	V
з	ZE	215	D7	167	A7	231	E7	218	DA	1079	0437	Z
и	I	216	D8	168	A8	232	E8	201	C9	1080	0438	I
й	SHORT I	217	D9	169	A9	233	E9	202	CA	1081	0439	J
к	KA	218	DA	170	AA	234	EA	203	CB	1082	043A	K
л	EL	219	DB	171	AB	235	EB	204	CC	1083	043B	L
м	EM	220	DC	172	AC	236	EC	205	CD	1084	043C	M
н	EN	221	DD	173	AD	237	ED	206	CE	1085	043D	N
о	O	222	DE	174	AE	238	EE	207	CF	1086	043E	O
п	PE	223	DF	175	AF	239	EF	208	D0	1087	043F	P
р	ER	224	E0	224	E0	240	F0	210	D2	1088	0440	R
с	ES	225	E1	225	E1	241	F1	211	D3	1089	0441	S
т	TE	226	E2	226	E2	242	F2	212	D4	1090	0442	T
у	U	227	E3	227	E3	243	F3	213	D5	1091	0443	U
ф	EF	228	E4	228	E4	244	F4	198	C6	1092	0444	F
х	HA	229	E5	229	E5	245	F5	200	C8	1093	0445	H
ц	TSE	230	E6	230	E6	246	F6	195	C3	1094	0446	C
ч	CHE	231	E7	231	E7	247	F7	222	DE	1095	0447	^
ш	SHA	232	E8	232	E8	248	F8	219	DB	1096	0448	[
щ	SHCHA	233	E9	233	E9	249	F9	221	DD	1097	0449]
ъ	HARD SIGN	234	EA	234	EA	250	FA	223	DF	1098	044A	_
ы	YERU	235	EB	235	EB	251	FB	217	D9	1099	044B	Y
ь	SOFT SIGN	236	EC	236	EC	252	FC	216	D8	1100	044C	X
э	E	237	ED	237	ED	253	FD	220	DC	1101	044D	\
ю	YU	238	EE	238	EE	254	FE	192	C0	1102	044E	@
я	YA	239	EF	239	EF	255	FF	209	D1	1103	044F	Q

1	2	I		II		III		IV		V		VI
А	А	176	B0	128	80	192	C0	225	E1	1040	0410	a
Б	BE	177	B1	129	81	193	C1	226	E2	1041	0411	b
В	VE	178	B2	130	82	194	C2	247	F7	1042	0412	w
Г	GHE	179	B3	131	83	195	C3	231	E7	1043	0413	g
Д	DE	180	B4	132	84	196	C4	228	E4	1044	0414	d
Е	IE	181	B5	133	85	197	C5	229	E5	1045	0415	e
Ё	Ю	161	A1	240	F0	168	A8	179	B3	1025	0401	3
Ж	ZHE	182	B6	134	86	198	C6	246	F6	1046	0416	v
З	ZE	183	B7	135	87	199	C7	250	FA	1047	0417	z
И	I	184	B8	136	88	200	C8	233	E9	1048	0418	i
Й	SHORT I	185	B9	137	89	201	C9	234	EA	1049	0419	j
К	KA	186	BA	138	8A	202	CA	235	EB	1050	041A	k
Л	EL	187	BB	139	8B	203	CB	236	EC	1051	041B	l
М	EM	188	BC	140	8C	204	CC	237	ED	1052	041C	m
Н	EN	189	BD	141	8D	205	CD	238	EE	1053	041D	n
О	О	190	BE	142	8E	206	CE	239	EF	1054	041E	o
П	PE	191	BF	143	8F	207	CF	240	F0	1055	041F	p
Р	ER	192	C0	144	90	208	D0	242	F2	1056	0420	r
С	ES	193	C1	145	91	209	D1	243	F3	1057	0421	s
Т	TE	194	C2	146	92	210	D2	244	F4	1058	0422	t
У	U	195	C3	147	93	211	D3	245	F5	1059	0423	u
Ф	EF	196	C4	148	94	212	D4	230	E6	1060	0424	f
Х	HA	197	C5	149	95	213	D5	232	E8	1061	0425	h
Ц	TSE	198	C6	150	96	214	D6	227	E3	1062	0426	c
Ч	CHE	199	C7	151	97	215	D7	254	FE	1063	0427	
Ш	SHA	200	C8	152	98	216	D8	251	FB	1064	0428	{
Щ	SHCHA	201	C9	153	99	217	D9	253	FD	1065	0429	}
Ъ	HARD SIGN	202	CA	154	9A	218	DA	255	FF	1066	042A	
Ы	YERU	203	CB	155	9B	219	DB	249	F9	1067	042B	y
Ь	SOFT SIGN	204	CC	156	9C	220	DC	248	F8	1068	042C	x
Э	Е	205	CD	157	9D	221	DD	252	FC	1069	042D	
Ю	YU	206	CE	158	9E	222	DE	224	E0	1070	042E	'
Я	YA	207	CF	159	9F	223	DF	241	F1	1071	042F	q

Кодовая таблица UTF-8

UTF-8 — это кодировка символов переменной длины, что, в данном случае, означает длину от 1 до 4 байт на символ. Первый байт UTF-8 используется для кодирования ASCII, что означает, что данный набор символов полностью обратно совместим с ASCII. UTF-8

означает, что символы ASCII и Latin полностью взаимозаменяемы с небольшим увеличением размера данных, так как используется только первый байт. Пользователи восточных алфавитов, например, японского, которым назначили диапазон с большим числом байт несчастливы, так как это приводит к 50-процентной избыточности в их данных.

Кодовые таблицы и принцип кодирования символов национальных алфавитов вы можете найти в сети Интернет.

2 Лабораторные работы

2.1 Лабораторная работа №1. Исследование статистических характеристик текстового документа

Цель работы

Исследовать с точки зрения статистики некий осмысленный текстовый документ. Освоение инструментов Pascal для работы с текстовыми файлами, сортировка массивов.

Задание и порядок выполнения работы

Исследовать статистические характеристики текстового документа. Получить таблицу частоты использования символов алфавита отсортированную по убыванию частоты.

Определитесь с текстовым документом, характеристики которого вы будете исследовать. Требования такие:

- документ на английском языке
- размер текстового файла - 20-60кБ
- документ должен быть осмысленным

В программной оболочке FreePascal создайте программу со следующим функционалом:

- на вход программы поступает текстовый файл
 - в результате обработки программа формирует массив из двух элементов: символ или его ASCII-код и количество появлений данного символа в тексте
 - производится сортировка массива любым из доступных вам методов
 - на экран выводится таблица состоящая из следующих столбцов: код символа, сам символ в кавычках, количество появлений в тексте, частота нормированная к 1
- В отчет необходимо поместить таблицу с частотой появления символов в тексте.

Отчет по проделанной работе готовится в текстовом редакторе OpenOffice.org Write и предоставляется для проверки в электронном виде в формате электронных документов PDF.

Отчет должен состоять из следующих частей:

- введение
- постановка задачи
- основная часть
- заключение
- приложение

В приложении приводится исходный код программы на языке Pascal.

2.2 Лабораторная работа №2. Работа с кодовыми таблицами русского языка

Цель работы

Познакомиться с разнообразием кодовых таблиц для русских символов, научиться производить конвертацию текста из одной кодировки в другую.

Задание и порядок выполнения работы

Исходный текст - файл с русским текстом. Кодировка файла - CP1251. Произвести кодирование текста в кодировки KOI8-R, ISO 8859-5, Microsoft CP866, Unicode.

Выберите исходный файл на русском языке, может содержать цифры, знаки и буквы латинского алфавита. Кодировка файла CP1251. Размер файла - от 200 байт до 1 килобайта. Имя файла: <filename>.cp1251

На языке Pascal написать программу которая производит преобразование файла в другие кодировки. Выбор кодировки производится с помощью меню. Возможен также выбор кодировки с помощью ключей командной строки, например:

```
transcode -koi8r <filename>
```

```
transcode -cp866 <filename>
```

Результат работы программы записывается в файлы с именами: <filename>.cp866, <filename>.koi8r и т.д.

Проверку правильности преобразования можно произвести с помощью программы kwrite.

Кодовые таблицы можно найти либо в учебном пособии, в разделе приложений, либо в интернет.

По окончании работы необходимо подготовить отчет.

2.3 Лабораторная работа №3. Основы криптографии

Цель работы

Изучить базовые методы криптографической защиты документов. Создать программы для шифрования документов несколькими способами.

Задание и порядок выполнения работы

Исходный текст - текстовый файл на английском или русском языках (только однобайтовая кодировка, например CP1251). Необходимо на языке Pascal создать следующие программы:

- шифрование/дешифрование простой заменой или подстановкой
- кодирование/декодирование шифром-перестановкой
- шифрование/дешифрование с ключевым словом.

Все программы должны получать имя файла с исходным текстом в командной строке. Для работы с командной строкой воспользуйтесь функцией ParamStr() языка Pascal.

Выберите исходный файл на русском или английском языках, может содержать цифры, знаки и буквы латинского алфавита. Кодировка файла CP1251. Размер файла - от 200 байт до 1 килобайта. Имя файла: <filename>.src

На языке Pascal написать программы которая производит шифрование/дешифрование файла. Выбор режима работы программы производится с помощью меню. Возможен также выбор режима работы с помощью ключей командной строки, например:

```
cyfer -encode <filename>
```

```
cyfer -decode <filename>
```

Результат работы программ записывается в файлы с именами: <filename>.dst.

По окончании работы необходимо подготовить отчет. В отчете привести тексты как исходного сообщения, так и закодированного шифром.

2.4 Лабораторная работа №4. Изучение языка гипертекстовой разметки HTML

Цель работы

Познакомиться с языком гипертекстовой разметки HTML. Создать и оформить html-документ.

Теоретическая часть

HTML был изобретён в 1990 году учёным Тимом Бёрнсом-Ли (Tim Berners-Lee) и предназначался для облегчения обмена документами между учёными различных университетов. Проект имел больший успех, чем Tim Berners-Lee мог ожидать. Этим изобретением HTML он заложил основы современной сети Internet.

HTML это язык, который позволяет представлять информацию (например, научные исследования) в Internet. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста. Чтобы увидеть HTML-коды страницы в Internet, щёлкните "View" в линейке меню вашего браузера и выберите "Source".

Для того, чтобы текстовый файл превратился в HTML-файл, поменять его расширение с ".txt" на ".html" недостаточно. Надо соблюсти "правило первой строки":

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки декларации версии HTML !DOCTYPE, которая обычно выглядит так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ. В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 3.2, которая хоть и не отличается новизной, но, в отличие от более поздних версий, является полноценным, широко распространенным стандартом без каких-либо неопределенностей.

После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью тега-контейнера <html>. Необходимо отметить, что любой HTML-документ открывается тегом <html> и им же закрывается.

Затем, между тегам <html> и </html> следует разместить заголовок и тело документа. Вот как должен выглядеть ваш базовый HTML-файл перед началом работы:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Заголовок документа</title>
</head>
<body>
Текст документа
</body>
</html>
```

Из схемы видно, что документ состоит из двух основных блоков – "заголовка" и "тела

документа". Заголовок определяется с помощью элемента HEAD, а тело – элементом BODY.

Заголовок содержит "техническую" информацию о документе, хотя чаще всего используется только для обозначения его названия (см. элемент TITLE).

Тело документа – святая святых. Именно в нем находится все то, что отображается на странице: текст, картинки, таблицы. Соответственно, делаем вывод: большинство ваших HTML-экспериментов будет проводиться в пространстве между тегами <BODY> и </BODY>.

Заголовок HTML-документа

Заголовок документа создается с помощью элемента HEAD, между тегами которого размещаются элементы, содержащие техническую информацию о документе. Заголовок обычно располагается до тела документа (см. структуру HTML-документа).

Элементы, относящиеся к заголовку документа:

- HEAD Определяет начало и конец заголовка документа
- TITLE Определяет имя всего документа, которое отображается в заголовке окна браузера
- BASE Определяет базовый адрес, от которого отсчитываются относительные ссылки внутри документа
- STYLE Используется для вставки в документ таблицы стилей CSS
- LINK Описывает взаимосвязь документа с другими объектами
- META Используется для вставки метаданных

HEAD

Определяет начало и конец заголовка документа. Является контейнером для элементов, содержащих техническую информацию о документе. (TITLE,BASE,STYLE,LINK,META).

Пример:

```
<HTML>
<!-- Начинаем заголовок... -->
<HEAD>
<title>Справочник по HTML</title>
</HEAD>
<!-- ...кончили. Дальше пошло тело документа -->
<BODY>
Текст документа
</BODY>
</HTML>
```

TITLE

Определяет имя всего документа. Имя, как правило, отображается в заголовке окна браузера. Данный элемент обязателен для любого HTML-документа и может быть указан не более одного раза.

Пример:

```
...
<HEAD>
```

```
<TITLE>Руководство по эксплуатации</TITLE>
</HEAD>
```

BODY

Указывает начало и конец тела HTML-документа. Между начальным и конечным тегами содержится текст документа, изображения и таблицы. Одним словом, все HTML-элементы, отвечающие за отображение документа, управление им и гипертекстовые ссылки. Элемент BODY должен встречаться в документе не более одного раза.

Атрибуты:

MARGINHEIGHT – определяет ширину (в пикселах) верхнего и нижнего полей документа. Работает только в браузерах Netscape.

TOPMARGIN – определяет ширину (в пикселах) верхнего и нижнего полей документа. Работает только в браузерах Internet Explorer.

MARGINWIDTH – определяет ширину (в пикселах) левого и правого полей документа. Работает только в браузерах Netscape.

LEFTMARGIN – определяет ширину (в пикселах) левого и правого полей документа. Работает только в браузерах Internet Explorer.

BACKGROUND – определяет изображение для "заливки" фона. Значение задается в виде полного URL или имени файла с картинкой в формате GIF или JPG.

BGCOLOR – определяет цвет фона документа.

TEXT – определяет цвет текста в документе.

LINK – определяет цвет гиперссылок в документе.

ALINK – определяет цвет подсветки гиперссылок в момент нажатия.

VLINK – определяет цвет гиперссылок на документы, которые вы уже просмотрели.

Значения атрибутов BGCOLOR, TEXT, LINK, ALINK и VLINK задаются либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

Замечания: несмотря на то что с помощью атрибутов тега BODY можно задать много полезных параметров HTML документа, в настоящий момент целесообразнее использовать CSS – каскадные таблицы стилей.

Пример:

```
<HTML>
<BODY BACKGROUND="images/bricks.jpg" BGCOLOR="#202020" TEXT="#FFFFFF"
LINK="#FF0000" VLINK="#505050" MARGINHEIGHT="30" TOPMARGIN="30"
LEFTMARGIN="40" MARGINWIDTH="40">
...
Текст документа.
...
</BODY>
</HTML>
```

Гиперссылки

Ссылки на другие документы в HTML создаются либо с помощью элемента A, либо с

помощью навигационных карт.

Элемент А применяется, если ссылкой планируется сделать часть текста или целое изображение. Навигационные карты имеет смысл применять, если ссылкой будет часть изображения.

A

Самый необходимый элемент, без которого Интернет просто немислим. Используется для создания и использования гипертекстовых ссылок.

Атрибуты:

HREF – определяет находящийся между начальным и конечным тегами текст или изображение как гипертекстовую ссылку (URL, или линк) на документ (и/или область документа), указанный в значении данного атрибута. Возможные значения:

http://... – создает ссылку на www-документ;

ftp://... – создает ссылку на ftp-сайт или расположенный на нем файл;

mailto:... – запускает почтовую программу-клиент с заполненным полем имени получателя. Если после адреса поставить знак вопроса, то можно указать дополнительные атрибуты, разделенные знаком "&;

news:... – создает ссылку на конференцию сервера новостей;

telnet://... – создает ссылку на telnet-сессию с удаленной машиной;

wais://... – создает ссылку на WAIS – сервер;

gopher://... – создает ссылку на Gopher – сервер;

Если тип соединения и адрес машины не указаны, в качестве отправной точки используется адрес текущего документа. Это позволяет использовать относительные ссылки.

Например, линк `Документация` будет ссылаться на файл title.html в подкаталоге docs (относительно текущего).

NAME – помечает находящуюся между начальным и конечным тегами область документа как возможный объект для ссылки. В качестве значения нужно латиницей написать любое слово-указатель, уникальное для данного документа.

Например: `Раздел1`. Теперь вы можете ссылаться на помеченную область простым указанием ее имени после имени документа. Например, линк `Раздел1` отправит вас в раздел "part" файла document.html, а линк `В конец документа` – в раздел "bottom" текущего документа. (см. Пример 1)

TARGET – определяет окно (фрейм), на которое указывает гипертекстовая ссылка. Этот атрибут используется только совместно с атрибутом HREF. В качестве значения необходимо задать либо имя одного из существующих фреймов, либо одно из следующих зарезервированных имен:

_self – указывает, что определенный в атрибуте HREF документ должен отображаться в текущем фрейме;

_parent – указывает, что документ должен отображаться во фрейме-родителе текущего фрейма. Иначе говоря, _parent ссылается на окно, содержащее FRAMESET, включающий текущий фрейм;

_top – указывает, что документ должен отображаться в окне-родителе всей текущей фреймовой структуры;

_blank – указывает, что документ должен отображаться в новом окне.

Пример 1:

```
<!-- Использование атрибута NAME: -->
<A NAME="history">История бодибилдинга</A>
...
<A NAME="now">Спорт глазами современника</A>
...
Вернуться к разделу<A HREF="#history">истории</A>
```

Пример 2:

```
<!-- Использование атрибута HREF: -->
<A HREF="ftp://ftp.cdrom.com" TARGET="_blank">FTP-site</A>
<A HREF="http://opengl.rdc.ru">Русский проект по OpenGL</A>
...
```

Пример 3:

```
<!-- Создадим ссылку для письма с указанием кучи атрибутов -->
<A HREF="mailto:green@igf.ru?subject=Приглашение
&cc=bg@microsoft.com&body=Приезжай на вечеринку.">
Отправить приглашение </A>.
<!-- или просто письмо : -->
<A HREF="mailto:green@igf.ru?subject=Привет">авторам</A>.
```

Текстовые блоки

В этом разделе описаны элементы, разбивающие текст документа на блоки тем или иным способом. Типичными примерами текстовых блоков являются параграфы, абзацы и главы. Для отделения одной части текста от другой также используются разделительные горизонтальные линии и символы возврата каретки.

Элементы:

- H1, H2, ... H6 Используются для создания заголовков текста
- P Используется для разметки параграфов.
- DIV Отделяет блок HTML-документа от остальной его части
- ADDRESS Оформляет текст как почтовый адрес
- BLOCKQUOTE Оформляет текст как цитату
- BR Осуществляет перевод строки
- HR Вставляет в текст горизонтальную разделительную линию.
- PRE Включает в документ (моноширинным шрифтом) блок предварительно отформатированного текста

H1,H2,...H6

Используются для создания заголовков текста. Существует шесть уровней заголовков, различающихся величиной шрифта. С их помощью можно разбивать текст на смысловые уровни – разделы и подразделы.

Атрибуты:

ALIGN – определяет способ выравнивания заголовка по горизонтали.

Возможные значения: left, right, center.

Пример:

```
<H1 ALIGN="center">Самый большой заголовок посередине</H1>
```

```
<H2>Заголовок поменьше</H2>
```

...

```
<H6>Малюююсенький такой заголовочек</H6>
```

P

Используется для разметки параграфов.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания параграфа.

Возможные значения: left, center, right. По умолчанию имеет значение left.

Пример:

```
<P ALIGN="center">Это центрированный параграф.<BR>
```

```
Текст располагается в центре окна браузера</P>
```

```
<P ALIGN="right">А это параграф, выровненный по правому краю.</P>
```

DIV

Используется для логического выделения блока HTML-документа. Элемент группировки, как и элемент SPAN. В современном сайтостроении используется как удобный контейнер для объектов страницы, которым легко динамически манипулировать – перемещать, включать/выключать, создавать слои, регулировать отступы и т.п.

В браузеронезависимой вёрстке обычно используется для выравнивания блока html-кода в окне браузера.

Находящиеся между начальным и конечным тегами текст или HTML-элементы по умолчанию оформляются как отдельный параграф.

Атрибуты:

ALIGN – определяет выравнивание содержимого элемента DIV. Атрибут может принимать значения: left, right, center.

Пример:

```
...Текст документа...
```

```
<DIV ALIGN="center">
```

```
...Текст, таблицы, изображения. Выравнивание по центру.
```

```
</DIV>  
...Текст документа...
```

ADDRESS

Находящийся между начальным и конечным тегами текст оформляется как почтовый адрес. Чаще всего оформление выражается в выделении строки адреса курсивом.

Пример:

```
Пишите по следующему адресу:  
<ADDRESS>  
Москва. ул. Академика Королева, 13 <BR>  
Мурзилке  
</ADDRESS>
```

BLOCKQUOTE

Оформляет находящийся между начальным и конечным тегами текст как цитату. Используется для длинных цитат (в отличие от элемента CITE). Цитируемый текст отображается отдельным абзацем с увеличенным отступом.

Пример:

```
Редакция журнала "Домосед" выражает благодарность  
Бухаресту Магарычу Шницелю за замечательное стихотворение:  
<BLOCKQUOTE>  
Синели красные ромашки, <BR>  
Желтели в небе облака, <BR>  
Синицы в теплый край летели, <BR>  
К истоку двигалась река. <BR>  
...  
</BLOCKQUOTE>
```

BR

Данный элемент осуществляет перевод строки, то есть практически аналогичен нажатию Enter в текстовом редакторе. После того, как в браузерах появилась возможность обтекания изображения текстом (см. атрибут ALIGN элемента IMG), понадобился дополнительный атрибут CLEAR. Элемент не имеет конечного тега.

Атрибуты:

CLEAR – указывает на необходимость завершения обтекания изображения текстом. Может принимать следующие значения :

- all – завершить обтекание изображения текстом.

- left – завершить обтекание текстом изображения, выровненного по левому краю.
- right – завершить обтекание текстом изображения, выровненного по правому краю.

Пример:

```
Первое предложение<BR>Второе предложение на следующей строке
```

HR

Вставляет в текст горизонтальную разделительную линию.

Атрибуты:

WIDTH – определяет длину линии в пикселах или процентах от ширины окна браузера.

SIZE – определяет толщину линии в пикселах.

ALIGN – определяет выравнивание горизонтальной линии. Атрибут может принимать следующие значения:

- left – выравнивание по левому краю документа.
- right – выравнивание по правому краю документа.
- center – выравнивание по центру документа (используется по умолчанию).

NOSHADE – определяет способ закрашки линии как сплошной. Атрибут является флагом и не требует указания значения. Без данного атрибута линия отображается объемной.

COLOR – определяет цвет линии. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов. Атрибут работает только в Internet Explorer.

Пример:

```
Текст, разделенный <HR NOSHADE WIDTH="50%"> сплошной горизонтальной линией.
```

Форматирование текста

В этом разделе описаны элементы для оформления и смыслового выделения текста – подчеркивания, изменения шрифта, выделения курсивом, цитирования и т.д.

Элементы форматирования текста:

- BASEFONT Определяет основной шрифт, которым должен отображаться текст документа
- FONT Позволяет изменять цвет, размер и тип шрифта текста
- I Выделяет текст курсивом
- EM Используется для смыслового выделения текста (курсивом)
- B Выделяет текст жирным шрифтом
- STRONG Усиленное выделение текста (жирным)
- U Выделяет текст подчеркнутым
- S, STRIKE Выделяет текст перечеркнутым
- BIG Отображает текст увеличенным шрифтом (относительно текущего)
- SMALL Отображает текст уменьшенным шрифтом (относительно текущего)
- SUP Отображает текст со сдвигом вверх (верхний индекс)

- SUB Отображает текст со сдвигом вниз (нижний индекс)
- CODE, SAMP Оформляют текст как формулу или программный код
- TT Отображает текст моноширинным шрифтом
- KBD Выделяет текст, который предлагается набрать на клавиатуре
- VAR Используется для обозначения в тексте переменных
- CITE Оформляет текст как цитату или ссылку на источник

FONT

Позволяет изменять цвет, размер и тип шрифта текста, находящегося между начальным и конечным тегами. Вне тегов и используется шрифт, указанный в элементе BASEFONT.

Атрибуты:

SIZE – определяет размер шрифта. Возможные значения:

- целое число от 1 до 7;
- относительный размер с указанием знака, вычисляется путем сложения с базовым размером, определенным с помощью атрибута SIZE элемента BASEFONT.

COLOR – определяет цвет текста. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

FACE – определяет используемый шрифт.

Пример:

```
<FONT SIZE="+2" COLOR="#AA0000">Увеличенный красный шрифт</FONT>
<FONT SIZE="3" FACE="Courier New" COLOR="Magenta">Моноширинный
фиолетовый текст 3 размера</FONT>
```

I

Текст, заключенный между начальным и конечным тегами, будет выделен курсивом.

Пример:

```
Текст с <I>курсивом</I>
```

EM

Логическое ударение. Используется для смыслового выделения текста, стоящего между начальным и конечным тегами. Результат обычно отображается курсивом. То есть элемент EM по действию практически аналогичен элементу I.

Пример:

```
Порой в России встречаются <EM>действительно талантливые</EM> веб-
мастера. Но только не друг с другом.
```


B

Текст, заключенный между начальным и конечным тегами, будет выделен жирным шрифтом.

Пример:

```
Текст с <B>выделенным</B> словом
```

Списки

Списки в HTML бывают двух видов: упорядоченные (пронумерованные) и неупорядоченные (непронумерованные). Отличаются они лишь способом оформления. Перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера.

Элементы:

- UL Создает неупорядоченный список
- OL Создает упорядоченный список
- LI Создает пункт меню внутри элементов OL или UL
- MENU, DIR Создает неупорядоченный список, подобный UL
- DL Открывает и закрывает список определений
- DT Создает термин в списке определений внутри элемента DL
- DD Создает определение термина внутри элемента DL

UL

Создает неупорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Пример:

```
<UL>  
<LI> Первый пункт списка </LI>  
<LI> Второй пункт списка </LI>  
<LI> Третий пункт списка </LI>  
</UL>
```

OL

Создает упорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов LI, обозначающих отдельные пункты списка.

Атрибуты:

START – определяет первое число, с которого начинается нумерация пунктов. (только целые числа)

TYPE – определяет стиль нумерации пунктов. Может иметь значения:

- "A" – заглавные буквы A, B, C ...
- "a" – строчные буквы a, b, c ...
- "I" – большие римские числа I, II, III ...
- "i" – маленькие римские числа i, ii, iii ...
- "1" – арабские числа 1, 2, 3 ...

По умолчанию <UL TYPE="1">.

Пример:

```
<OL TYPE="I" START="2">
<LI> Пункт два </LI>
<LI> Пункт три </LI>
<LI> Пункт четыре </LI>
</OL>
```

LI

Создает пункт в списке. Располагается внутри элементов OL или UL.

Атрибуты:

VALUE – изменяет порядок нумерации элементов списка. Используется только если элемент LI находится внутри элемента OL. В качестве значения указывается порядковый номер элемента.

Пример:

```
<OL TYPE="A" START="2">
<LI> Пункт, озаглавленный буквой B. </LI>
<LI VALUE="6"> Пункт, озаглавленный буквой F. </LI>
<LI> Пункт, озаглавленный буквой G. </LI>
</OL>
```

Объекты

Объекты – это графические и мультимедийные вставки в HTML-документ, такие как картинки, Flash-анимация, Java-апплеты, звуки, музыка, VRML.

Элементы:

- IMG Используется для вставки в HTML изображений
- EMBED Используется для вставки в HTML различных объектов
- NOEMBED Используется, если браузер не поддерживает элемент EMBED
- APPLET Используется для вставки в HTML Java-апплетов
- PARAM Используется для передачи параметров Java-программе (см. элемент APPLET)

IMG

Используется для вставки изображений в HTML-документ.

Это один из самых популярных элементов, незаменимый инструмент web-дизайнера. Элемент допускает вставку изображений в форматах JPEG (в том числе progressive jpeg) и CompuServe GIF (включая прозрачные и анимированные). Четвертые версии браузеров позволяют также использовать формат PNG, но до тех пор, пока они не устареют, от применения PNG лучше воздержаться.

Элемент IMG не имеет конечного тега.

Атрибуты:

SRC – обязательный атрибут. Указывает адрес (URL) файла с изображением.

HEIGHT и WIDTH – определяют ширину и высоту изображения соответственно. Если указанные значения не совпадают с реальным размером изображения, изображение масштабируется (порой с заметной потерей качества).

HSPACE и VSPACE – определяют отступ картинки (в пикселах) по горизонтали и вертикали от других объектов документа. Просто необходимо при обтекании изображения текстом.

ALIGN – обязательный атрибут. Указывает способ выравнивания изображения в документе. Может принимать следующие значения:

- left – выравнивает изображение по левому краю документа. Прилегающий текст обтекает изображение справа.
- right – выравнивает изображение по правому краю документа. Прилегающий текст обтекает изображение слева.
- top и texttop – выравнивают верхнюю кромку изображения с верхней линией текущей текстовой строки.
- middle – выравнивает базовую линию текущей текстовой строки с центром изображения.
- absmiddle – выравнивает центр текущей текстовой строки с центром изображения.
- bottom и baseline – выравнивает нижнюю кромку изображения с базовой линией текущей текстовой строки.
- absbottom – выравнивает нижнюю кромку изображения с нижней кромкой текущей текстовой строки.

NAME – определяет имя изображения, уникальное для данного документа. Вы можете указать любое имя без пробелов с использованием латинских символов и цифр. Имя необходимо, если вы планируете осуществлять доступ к изображению, например, из JavaScript-сценариев.

ALT – определяет текст, отображаемый браузером на месте изображения, если браузер не может найти файл с изображением или включен в текстовый режим. В качестве значения задается текст с описанием изображения.

BORDER – определяет ширину рамки вокруг изображения в пикселах. Рамка возникает, только если изображение является гипертекстовой ссылкой. В таких случаях значение BORDER обычно указывают равным нулю.

LOWSRC – указывает адрес (URL) файла с альтернативным изображением более низкого качества (и, соответственно, меньшего объема), чем изображение, указанное в атрибуте SRC. Браузеры Netscape, поддерживающие данный атрибут, сначала загрузят

картинку из LOWSRC, а затем заменяют ее картинкой из SRC.

USEMAP – применяет к изображению навигационную карту (image map), заданную элементом MAP. В качестве значения задается имя карты с предшествующей ему решеткой. Например, если имя карты – "map1", то ссылка на нее будет выглядеть как "#map1" (см. Пример 4). Обратите внимание: прописные и строчные буквы в данном атрибуте трактуются браузером как разные.

ISMAP – определяет изображение как навигационную карту (image map), обрабатываемую сервером. Имеет смысл использовать только тогда, когда изображение является гиперссылкой. После клика мышкой на изображении серверу отправляются x,y-координаты нажатия. В зависимости от полученных координат, сервер (при наличии на нем соответствующего программного обеспечения) может показать вам тот или иной документ. Данный атрибут является флагом и не требует присвоения значения.

Пример 1:

```
<IMG src="/img/picture.gif" WIDTH="45" HEIGHT="53" ALT="Рысь"
HSPACE="10" ALIGN="left"> Этот текст обтекает картинку справа и
находится от нее на расстоянии 10 пикселей.
```

Пример 2. Использование изображения в качестве гиперссылки:

```
<A HREF="carlson.html">
<IMG src="/img/button.jpg" WIDTH="70" HEIGHT="30" ALIGN="right"
BORDER="0" ALT="Досье Карлсона">
</A>
```

Для просмотра досье нажмите на кнопку справа.

Пример 3. Использование ISMAP:

```
<A HREF="http://www.igf.ru/bin/imagemaps/map1">
<IMG SRC="map.gif" ISMAP></A>');
```

Пример 4. Использование USEMAP:

```
<IMG src="/img/buttons.jpg" WIDTH="170" HEIGHT="120" ALIGN="middle"
BORDER="0" USEMAP="#ButtonsMap">');
```

Примечания (особо важно):

Золотое правило web-мастера – всегда явно задавать размеры картинки в атрибутах HEIGHT и WIDTH, резервируя тем самым место в окне браузера еще до загрузки изображения. В противном случае документ при загрузке каждой картинки будет заново перерисовываться. А на медленных машинах (или при подключении к сети через модем) это смотрится просто отвратительно.

Второе золотое правило web-мастера: если на картинке изображено что-то разборчивое, нужно описать это словами в атрибуте ALT.

Всегда сразу после ставьте
! А то проблем не миновать – после картинки появится пустое пространство в несколько пикселей. Причём ставьте вплотную, без пробелов:
.

Для завершения обтекания изображения текстом используйте атрибут CLEAR элемента BR.

Значения `top` и `texttop` атрибута `ALIGN` не совсем идентичны, и их использование порой дает разный результат. Попробуйте поэкспериментировать.

Указывайте значения атрибутов `HSPACE` и `VSPACE`, даже если вы хотите оставить поля нулевой ширины. Бывает, что некоторые браузеры по умолчанию присваивают им какое-то небольшое значение, не равное нулю.

Таблицы

Элементы для создания таблиц:

Таблицы в HTML формируются нетрадиционным способом – построчно. Сначала с помощью элемента `TR` необходимо создать ряд таблицы, в который затем элементом `TD` помещаются ячейки.

Важно:

В HTML таблицы используются не только для отображения таблиц как таковых, но и для дизайна. С помощью таблиц можно создать невидимый "каркас" страницы, помогающий расположить текст и изображения так, как вам нравится.

Элементы для создания таблиц:

- `TABLE` Создает таблицу
- `CAPTION` Задаёт заголовок таблицы
- `TR` Создает новый ряд (строку) ячеек таблицы
- `TD` и `TH` Создает ячейку с данными в текущей строке

TABLE

Элемент для создания таблицы. Обязательно должен иметь начальный и конечный теги. По умолчанию таблица печатается без рамки, а разметка осуществляется автоматически в зависимости от объема содержащейся в ней информации. Ячейки внутри таблицы создаются с помощью элементов `TR`, `TD`, `TH` и `CAPTION`.

Атрибуты:

`ALIGN` – определяет способ горизонтального выравнивания таблицы. Возможные значения: `left`, `center`, `right`. Значение по умолчанию – `left`.

`VALIGN` – должен определять способ вертикального выравнивания таблицы. Возможные значения: `top`, `bottom`, `middle`.

`BORDER` – определяет ширину внешней рамки таблицы (в пикселах). При `BORDER="0"` или при отсутствии этого атрибута рамка отображаться не будет.

`CELLPADDING` – определяет расстояние (в пикселах) между рамкой каждой ячейки таблицы и содержащимся в ней материалом.

`CELLSPACING` – определяет расстояние (в пикселах) между границами соседних ячеек.

`WIDTH` – определяет ширину таблицы. Ширина задается либо в пикселах, либо в процентном отношении к ширине окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

`HEIGHT` – определяет высоту таблицы. Высота задается либо в пикселах, либо в процентном отношении к высоте окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

`BGCOLOR` – определяет цвет фона ячеек таблицы. Задается либо RGB-значением в

шестнадцатиричной системе, либо одним из 16 базовых цветов.

BACKGROUND – позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

TR

Создает новый ряд (строку) ячеек таблицы. Ячейки в ряду создаются с помощью элементов TD и TH

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания содержимого всех ячеек данного ряда. Возможные значения: left, center, right.

VALIGN – определяет способ вертикального выравнивания содержимого всех ячеек данного ряда. Возможные значения: top, bottom, middle.

BGCOLOR – определяет цвет фона для всех ячеек данного ряда. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

TD и TH

Элемент TD создает ячейку с данными в текущей строке. Элемент TH также создает ячейку, но определяет ее как ячейку-заголовок.

Такое разграничение позволяет браузерам оформлять содержимое ячейки-заголовка и ячеек с данными разными шрифтами. Кроме того, должна улучшиться работа браузеров, использующих речевой интерфейс. В качестве содержимого ячейки можно использовать другие таблицы.

Атрибуты:

ALIGN – определяет способ горизонтального выравнивания содержимого ячейки. Возможные значения: left, center, right. По умолчанию способ выравнивания определяется значением атрибута ALIGN элемента TR. Если же и он не задан, то для TD выполняется выравнивание по левому краю, а для TH – центрирование.

VALIGN – определяет способ вертикального выравнивания содержимого ячейки. Возможные значения: top, bottom, middle. По умолчанию происходит выравнивание по центру (VALIGN="middle"), если значение этого атрибута не было задано ранее в элементе TR.

WIDTH – определяет ширину ячейки. Ширина задается в пикселах или в процентном отношении к ширине таблицы.

HEIGHT – определяет высоту ячейки. Высота задается в пикселах или в процентном отношении к высоте таблицы.

COLSPAN – определяет количество столбцов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

ROWSPAN – определяет количество рядов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

NOWRAP – блокирует автоматический перенос слов в пределах текущей ячейки. Обратите внимания на примечание, касающееся использования данного атрибута (далее, внизу страницы).

BGCOLOR – определяет цвет фона ячейки. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

BACKGROUND – заполняет ячейку фоновым рисунком. Необходимо указать URL рисунка. Данный атрибут не работает в старых версиях браузера Netscape (до 3.X включительно).

Задание и порядок выполнения работы

Создать мини-сайт по оригинальной тематике. Страницы сайта должны содержать как минимум следующие элементы языка HTML:

- заголовки
- абзацы
- списки
- изображения
- таблицы
- ссылки

Выбрать тему мини-сайта. Это может быть тематика, которая близка студенту.

Спроектировать структуру мини-сайта.

Произвести верстку страниц.

По итогам выполнения работы подготовить отчет. В отчете привести исходный код страниц сайта и их внешний вид.

2.5 Отчет по лабораторным работам

Отчет по каждой лабораторной работе должен включать следующие элементы:

1. Титульный лист. Образец оформления титульного листа приведен в Приложении А.
2. Оглавление. В оглавлении должны быть приведены заголовки всех разделов документа (вплоть до второго или третьего уровня) с указанием номеров страниц.
3. Задание на лабораторную работу. Задание должно быть полным, т. е. включать общую часть задания, индивидуальное задание согласно варианту, а также описание входных и выходных данных программы.
4. Краткую теорию. В краткой теории необходимо описать синтаксис и семантику анализируемого программой языка, а также все реализуемые в программе алгоритмы.
5. Результаты работы. В результаты работы включается формальное описание построенных синтаксических анализаторов, а также результаты тестовых прогонов программы.
6. Заключение (выводы). Здесь формулируются выводы по полученным результатам работы.
7. Список использованных источников. На приведенные источники в тексте документа должны быть расставлены ссылки.
8. Приложения. Например, это приложения с листингами программ и т.п.

3 Практические работы

3.1 Основные понятия теории информации. Количество информации. Энтропия Шеннона

3.1.1 Тесты

Сколько в одном Мегабайте килобайт?

1) 1024; 2) 1000; 3) 10000; 4) 2048

Сколько в одном килобайте байт?

1) 1000; 2) 1024; 3) 10000; 4) 2048

Какое количество чисел можно закодировать 10 битами?

1) 256; 2) 100; 3) 1024; 4) 64

Сколько бит используется для кодирования одного пиксела изображения в стандарте TrueColor?

1) 16; 2) 24; 3) 32; 4) 48

Количество символов кодируемых таблицей ASCII+?

1) 100; 2) 128; 3) 256; 4) 1000

3.1.2 Задачи

Задача 1

Каков код букв W и w в ASCII?

Задача 2

В цифровых магнитофонах DAT частота дискретизации — 48 КГц. Какова максимальная частота звуковых волн, которые можно точно воспроизводить на таких магнитофонах?

Задача 3

Сколько бит в одном килобайте?

Задача 4

Какое из соотношений несет в себе больше информации $x = 5$ или $x > 3$?

Задача 5

Найти энтропию д.с.в. X , заданной распределением

X	1	2	3	4	5	6	7	8
p	0.1	0.2	0.1	0.05	0.1	0.05	0.3	0.1.

Задача 6

Значения д. с. в. X_1 и X_2 определяются подбрасыванием двух идеальных монет, а д.с.в. Y равна сумме количества “гербов”, выпавших при подбрасывании этих монет. Сколько информации об X_1 содержится в Y ?

Задача 7

Сколько информации об X_1 содержится в д.с.в. $Z = (X_1 + 1)^2 - X_2$, где независимые д. с. в. X_1 и X_2 могут с равной вероятностью принимать значение либо 0, либо 1? Найти HX_1 и HZ . Каков характер зависимости между X_1 и Z ?

Задача 8

Д. с. в. X_1 и X_2 определяются подбрасыванием двух идеальных тетраэдров, грани которых помечены числами от 1 до 4. Д. с. в. Y равна сумме чисел, выпавших при подбрасывании этих тетраэдров, т. е. $Y = X_1 + X_2$. Вычислить $I(X_1, Y)$, HX_1 и HY .

Задача 9

Подсчитать сколько информации об X_1 содержится в д.с.в. $Z = X_1 * X_2$, а также HZ . Д.с.в. X_1 и X_2 берутся из предыдущего упражнения.

Задача 10

Д.с.в. X_1 может принимать три значения $-1, 0$ и 1 с равными вероятностями. Д.с.в. X_2 с равными вероятностями может принимать значения $0, 1$ и 2 . X_1 и X_2 — независимы. $Y = X_1^2 + X_2$. Найти $I(X_1, Y)$, $I(X_2, Y)$, HX_1 , HX_2 , HY .

Задача 11

Д.с.в. X равна количеству “гербов”, выпавших на двух идеальных монетках. Найти энтропию X . Придумать минимальный код для X , вычислить его среднюю длину и обосновать его минимальность.

3.2 Передача информации по каналам связи

3.2.1 Тесты

Частота дискретизации для компакт диска стандарта CD-Digital Audio?

1) 10кГц; 2) 20кГц; 3) 44100Гц; 4) 22050Гц

Разрядность оцифрованного сигнала в стандарте CD-Digital Audio?

1) 10 бит; 2) 16 бит; 3) 8 бит; 4) 32 бит

Какова максимальная скорость передачи двоичной информации по каналу связи при условии что несущая частота 100кГц?

1) 1000 бит/с; 2) 100кБит/с; 3) 1024 бит/с; 4) 256 бод

3.2.2 Задачи

Задача 1

Вычислить НХ и М L(X) для кодов Хаффмена и Шеннона-Фэно для X. Д.с.в. X задается следующим распределением вероятностей:

X	1	2	3	4	5
p	7/18	1/6	1/6	1/6	1/9

Задача 2

Вычислить длины кодов Хаффмена и арифметического для сообщения ААВ, полученного от д. с. в. X со следующим распределением вероятностей $P(X = A) = 1/3$, $P(X = B) = 2/3$.

Задача 3

Декодировать арифметический код 011 для последовательности значений д.с.в. из предыдущего упражнения. Остановиться, после декодирования третьего символа.

Задача 4

Составить арифметический код для сообщения ВААВС, полученного от д.с.в. X со следующим распределением вероятностей $P(X = A) = 1/4$, $P(X = B) = 1/2$, $P(X = C) = 1/4$. Каков будет арифметический код для этого же сообщения, если X распределена по закону $P(X = A) = 1/3$, $P(X = B) = 7/15$, $P(X = C) = 1/5$?

Задача 5

Закодировать сообщение ВВСВВС, используя адаптивный алгоритм Хаффмена с

упорядоченным деревом.

Задача 6

Закодировать сообщения “AABCDAAACCCDBB”, “КИБЕРНЕТИКИ” и “СИНЯЯ СИНЕВА СИНИ”, используя адаптивный алгоритм Хаффмена с упорядоченным деревом. Вычислить длины в битах исходного сообщения в коде ASCII+ и его полученного кода.

Задача 7

Распаковать сообщение 'A'0'F'00'X'0111110101011011110100101, полученное по адаптивному алгоритму Хаффмена с упорядоченным деревом, рассчитать длину кода сжатого и несжатого сообщения в битах.

Задача 8

Составить адаптивный арифметический код с маркером конца для сообщения BAABC.

Задача 9

Закодировать сообщения “AABCDAAACCCDBB”, “КИБЕРНЕТИКИ” и “СИНЯЯ СИНЕВА СИНИ”, вычислить длины в битах полученных кодов, используя алгоритмы, LZ77 (словарь — 12 байт, буфер — 4 байта), LZ78 (словарь — 16 фраз), LZSS (словарь — 12 байт, буфер — 4 байта), LZW (словарь — ASCII+ и 16 фраз).

Задача 10

Может ли для первого символа сообщения код LZ78 быть короче кода LZW при одинаковых размерах словарей? Обосновать. Для LZW в размер словаря не включать позиции для ASCII+.

Задача 11

Распаковать каждое приведенное сообщение и рассчитать длину кода каждого сжатого сообщения в битах. Сообщение, сжатое LZ77 (словарь — 12 байт, буфер — 4 байта), — h0,0,'A'i h0,0,'F'i h0,0,'X'i h9,2,'F'i h8,1,'F'i h6,2,'X'i h4,3,'A'i. Сообщение, сжатое LZSS (словарь — 12 байт, буфер — 4 байта), — 0'A' 0'F' 0'X' 1h9,2i 1h8,2i 1h6,3i 1h4,4i 1h9,1i. Сообщение, сжатое LZ78 (словарь — 16 фраз), — h0,'A'i h0,'F'i h0,'X'i h1,'F'i h2,'X'i h5,'A'i h3,'A'i h2,'F'i h0,'A'i. Сообщение, сжатое LZW (словарь — ASCII+ и 16 фраз), — 0'A' 0'F' 0'X' h256i h257i h257i 0'A' h258i 0'F' 0'F' 0'A'.

Задача 12

По каналу связи без шума могут передаваться четыре сигнала длительностью 1 мс каждый. Вычислить емкость такого канала.

Задача 13

Три передатчика задаются случайными величинами со следующими законами распределения вероятностей:

1) $P(X_1 = -1) = 1/4$, $P(X_1 = 0) = 1/2$, $P(X_1 = 1) = 1/4$;

2) $P(X_2 = -1) = 1/3$, $P(X_2 = 0) = 1/3$, $P(X_2 = 1) = 1/3$;

3) $P(X_3 = n) = 2^{-n}$, $n = 1, 2, \dots$

Емкость канала связи с шумом равна 4000 бод. Вычислить максимальную скорость передачи данных по этому каналу каждым из передатчиков, обеспечивающую сколь угодно высокую надежность передачи.

3.3 Кодирование информации

3.3.1 Тесты

Тег языка HTML для оформления полужирного начертания текста

1) ``; 2) `<i>`; 3) `<u>`; 4) `
`

Тег языка HTML для оформления наклонного (курсива) начертания текста

1) ``; 2) `<i>`; 3) `<u>`; 4) `
`

3.3.2 Задачи

Задача 1

Пусть двоичный симметричный канал используется для передачи строк из двух бит. Построить таблицу вероятностей приема.

Задача 2

По двоичному симметричному каналу передаются строки длины 14. Какова вероятность того, что ровно пять символов будут приняты неправильно? Какова вероятность того, что менее пяти символов будут приняты неправильно? Сколько имеется строк, отличающихся от данной не больше, чем в четырех позициях?

Задача 3

Имеется (8, 9)-код с проверкой четности. Вычислить вероятность того, что в случае ошибки этот код ее не обнаружит, если вероятность ошибки при передаче каждого бита равна 1%. Вычислить также вероятность ошибочной передачи без использования кода. Сделать аналогичные расчеты для случая, когда вероятность ошибки в десять раз меньше.

Задача 4

Вычислить минимальную и максимальную оценки количества дополнительных разрядов r для кодовых слов длины n , если требуется, чтобы минимальное расстояние между ними было d . Рассмотреть случаи $n = 32$, $d = 3$ и $n = 23$, $d = 7$.

Задача 5

Может ли (6, 14)-код, минимальное расстояние между кодовыми словами которого 5, быть совершенным?

Задача 6

Построить кодовые слова квазисовершенного (9, n)-кода, исправляющего однократные ошибки, для тех сообщений, которые соответствуют числам 55, 200 и декодировать слова 1000001000001, 1100010111100, полученные по каналу связи, использующему этот код.

Задача 7

По кодирующему многочлену $x^7 + x^5 + x + 1$ построить полиномиальные коды для двоичных сообщений 0100, 10001101, 11110.

Задача 8

Принадлежат ли коду Голея кодовые слова 10000101011111010011111 и 11000111011110010011111?

Задача 9

Построить CRC-4 код для сообщений 10000000 и 101111001, используя полином-генератор $x^4 + 1$.

Задача 10

Оценить объем памяти необходимый для хранения изображений:

- размер картинки 320*200 максимальное количество цветов 256
- размер картинки 640*480 максимальное количество цветов 256
- размер картинки 800*600 максимальное количество цветов 256
- размер картинки 1024*768 максимальное количество цветов 256
- размер картинки 640*480 максимальное количество цветов 65536
- размер картинки 800*600 максимальное количество цветов 65536
- размер картинки 1024*768 максимальное количество цветов 65536
- размер картинки 640*480 максимальное количество цветов 16М
- размер картинки 800*600 максимальное количество цветов 16М
- размер картинки 1024*768 максимальное количество цветов 16М
- размер картинки 1280*1024 максимальное количество цветов 16М

Задача 11

Оценить объем памяти необходимый для хранения видеоролика:

1. размер кадра 320*240, частота кадров 12fps, продолжительность 4 мин, максимальное количество цветов 16М
2. размер кадра 320*240, частота кадров 24fps, продолжительность 12 мин, максимальное количество цветов 16М

3. размер кадра 320*240, частота кадров 30fps, продолжительность 1 мин, максимальное количество цветов 16М
4. размер кадра 640*320, частота кадров 12fps, продолжительность 7 мин, максимальное количество цветов 16М
5. размер кадра 640*300, частота кадров 20fps, продолжительность 8 мин, максимальное количество цветов 16М
6. размер кадра 640*320, частота кадров 24fps, продолжительность 10 мин, максимальное количество цветов 16М
7. размер кадра 640*480, частота кадров 30fps, продолжительность 1 мин, максимальное количество цветов 16М

Задача 12

Рассчитать объем памяти необходимый для хранения в цифровой форме:

1. непрерывное изменение аналоговой величины в течение 10 секунд, характеристики АЦП: разрядность 8 бит, частота преобразования 10 отсчетов в секунду
2. непрерывное изменение аналоговой величины в течение 1 минуты, характеристики АЦП: разрядность 8 бит, частота преобразования 1000 отсчетов в секунду
3. непрерывное изменение аналоговой величины в течение 1 час, характеристики АЦП: разрядность 10 бит, частота преобразования 200 отсчетов в секунду
4. аудиофрагмент один канал длительность 23 секунды, характеристики АЦП: разрядность 8 бит, частота дискретизации 8кГц
5. аудиофрагмент один канал длительность 3 минуты, характеристики АЦП: разрядность 8 бит, частота дискретизации 11025 Гц
6. аудиофрагмент один канал длительность 1 час, характеристики АЦП: разрядность 10 бит, частота дискретизации 22050 Гц
7. аудиофрагмент стерео длительность 70 минут, характеристики АЦП: разрядность 16 бит, частота дискретизации 44100 Гц

Задача 13

Рассчитать объем памяти необходимый для хранения изображения с цифровой фотокамеры:

1. разрешение снимка 0.3 МПикс
2. разрешение снимка 2 МПикс
3. разрешение снимка 3.2 МПикс
4. разрешение снимка 4 МПикс
5. разрешение снимка 5 МПикс
6. разрешение снимка 6 МПикс
7. разрешение снимка 8 МПикс
8. разрешение снимка 10 МПикс
9. разрешение снимка 15.1 МПикс

Список рекомендованной литературы

1. Акулиничев, Юрий Павлович. Теория и техника передачи информации : учебное пособие. - Томск : Эль Контент , 2012. - 209 с.
2. Теория информационных систем / Г. П. Тартаковский. - М. : Физматкнига, 2005. - 303[1] с.
3. Лидовский, В. В. Основы теории информации и криптографии : учебное пособие / В. В. Лидовский. — 2-е изд. — Москва : ИНТУИТ, 2016. — 141 с.

Приложение А

(справочное)

Образец титульного листа отчета

Министерство науки и высшего образования РФ

Томский государственный университет
систем управления и радиоэлектроники

Кафедра Электронных приборов (ЭП)

Дисциплина «Теория информации и информационных систем»

ОТЧЕТ

по лабораторной работе

« _____ »

Выполнил студент
гр. 348
XXXXXXXXXXXXXX

Проверил преподаватель

Томск
2024