

Федеральное агентство по образованию РФ
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ГОУВПО «АмГУ»)

УТВЕРЖДАЮ
Зав. кафедрой ИУС
А.В.Бушманов

« _____ » _____

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

по дисциплине «Мировые информационные ресурсы и сети»

для студентов специальности 230201 – Информационные системы и технологии

Составитель доцент кафедры ИУС Галаган Т.А.

Факультет математики и информатики

Кафедра информационных и управляющих систем

*Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного
университета*

Т.А. Галаган

Учебно-методический комплекс по дисциплине «Мировые информационные ресурсы и сети». Для студентов специальности 230201 «Информационные системы и технологии» очной формы обучения.- Благовещенск: Амурский гос. ун-т, 2007.

Пособие содержит рабочую программу, курс лекций, методические рекомендации по проведению и выполнению лабораторных работ.

I. РАБОЧАЯ ПРОГРАММА

Курс	3	Семестр	5
Лекции	36	Экзамен	-
Практические (семинарские) занятия	-	(час.)	Зачет 5 семестр
Лабораторные занятия	18	(час.)	
Самостоятельная работа	54	(час.)	
Всего часов	108	час.	

1. ЦЕЛЬ И ЗАДАЧИ ДИСЦИПЛИНЫ

Информационные процессы накопления знаний и превращение их в информационный ресурс общества становятся важнейшим фактором социально-экономического развития страны и национальной экономики. Многочисленные международные, региональные и национальные информационные системы обладают громадными потенциальными возможностями для поиска идей, поэтому важно понимание методов доступа к этим информационным массивам на основе современных информационно-вычислительных и коммуникационных технических средств.

Совместное использование информации из множества источников в среде открытых. Поскольку бизнес-деятельность все больше опирается на развивающуюся технологию, требования к повышению интерактивности обмена информацией и работе в реальном времени значительно возрастают.

Цель настоящего курса – дать студентам комплекс знаний по организации мировых информационных ресурсов; о состоянии рынка электронной информации Российской Федерации и условиям предоставления и технологии доступа к информационным ресурсам; по основным принципам построения компьютерных сетей, их стандартизации.

В результате изучения дисциплины студент должен иметь представление о национальных и мировых информационных ресурсах; знать принципы работы и построения компьютерных сетей, сетевые стандарты, требования предъявляемые к сетям на современном предприятии; владеть умениями и навыками эффективного использования мировых ресурсов, основами программирования на JavaScript.

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. ФЕДЕРАЛЬНЫЙ КОМПОНЕНТ

Программа курса «Мировые информационные ресурсы и сети» составлена в соответствии с требованиями государственного образовательного стандарта специализации – Интегрированные системы автоматизированного управления, специализации 230201, блок дисциплине специализаций ДС. 08.

2.2. ЛЕКЦИИ (36 часов)

2.2.1. Рынок информационных услуг. Биржевая, статистическая и коммерческая информация. Информационный потенциал. (2 часа)

2.2.2. Мировой рынок информационных услуг. Этапы развития мирового рынка информационных услуг. Профессиональные базы данных. Информационные ресурсы сети Интернет. Эффективность поиска информации. (10 часов)

2.2.3. Информационный рынок Российской Федерации. Правовые вопросы информационной работы РФ. Государственные информационные ресурсы. Библиотечная сеть РФ. Информационные ресурсы архивного фонда. Статистическая, научно-техническая информация. Государственные и негосударственные системы правовой информации. (18 часов)

2.2.4. Глобальные сети. Протокол TCP/IP. Протокол передачи гипертекста (HTTP). Поиск информации в Интернете. Электронная коммерция. (6 часов)

2.3 –

2.4. ЛАБОРАТОРНЫЕ РАБОТЫ (18 часов)

Лабораторные работы посвящены изучению основ программирования JavaScript.

2.4.1. Переменные, операции, функции ввода-вывода, операторы ветвления. (2 часа).

2.4.2. Операторы цикла, функции JavaScript. (2 часа)

2.4.3. Встроенные объекты JavaScript.(2 часа)

2.4.4. Основы создание сценариев. Работа с фреймами и окнами (4 часа)

2.4.5. Объектная модель браузера и документа. (4 часа)

2.4.6. Динамическое изменение элементов HTML-документа средствами JavaScript. (4 часа)

2.5. Практические занятия –

2.6. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

2.6. 1. Для самостоятельной работы студентам предлагается написать реферат, материал к которым они должны находить, используя знания и навыки, полученные при изучении курса "Мировые информационные ресурсы и сети".

2.6.2. Темы рекомендуемые для самостоятельного изучения

2.6.2.1. Средства факсимильной связи в масштабе предприятия

2.6.2.2. Управление электронной почтой в масштабе предприятия

2.6.2.3. Сети с коммутацией пакетов (X.25)

2.6.2.4. Поисковые системы Интернет

2.7. ПЕРЕЧЕНЬ И ТЕМЫ ПРОМЕЖУТОЧНЫХ ФОРМ КОНТРОЛЯ ЗНАНИЙ

2.7. 1. Темы рефератов:

1) Виды источников информации и их анализ.

- 2) Исследование и анализ рынка финансовой информации, его информационные службы
- 3) Исследование и анализ рынка экономической информации РФ.
- 4) Исследование и анализ рынка коммерческой информации РФ.
- 5) Исследование и анализ рынка деловых новостей РФ.
- 6) Анализ рынка биржевой информации РФ.
- 7) Информационные ресурсы ООО «Международное Бюро Информации и Телекоммуникаций» (МБИТ)
- 8) Исследование и анализ рынка информации «Консультант плюс»
- 9) Исследование и анализ рынка информации «ГАРАНТ – Сервис»
Исследование и анализ рынка информации центра компьютерных разработок «КОДЕКС»
- 10) Оценка стоимости информационных услуг (Стоимость доступа к информации в различных базах. Стоимость передачи информации пользователю. Стоимость эксплуатации пункта доступа).
- 11) Телеконференции – задачи, принцип работы, программные продукты, технологии, применяемые для проведения телеконференции
- 12) Информационные службы потребительской информации. Обслуживания личных потребностей человека (электронные покупки)
- 13) Анализ рынка туристической информации.
- 14) Беспроводные сети.
- 15) Использование шлюзов в сетях.
- 16) Сетевые операционные системы.

2.7.2. Для промежуточного контроля знаний разработан комплект тестов по темам лекций. Фрагмент теста по теме «мировой рынок информационных услуг»:

1. Товаром на рынке информационных услуг являются:

- а) компьютеры; б) программные средства; в) информация;
г) оргтехника.

2. Основной объем услуг на мировом информационном рынке оказывают в настоящее время:

- а) государственные информационные службы;
б) коммерческие информационные службы.

3. Расположите группы информации по убыванию требований к оперативности предоставления:

- а) научно-техническая; б) биржевая и финансовая; в) статистическая; г) коммерческая.

4. Информационные центры-генераторы выполняют функции:

- а) сбора информации;
б) формирование и ведения баз данных;
в) обслуживания потребителей.

5. На мировом рынке информационных ресурсов преобладают

- а) числовые базы; б) текстовые базы.

6. Лидерами на мировом рынке в области информации об интеллектуальной собственности являются:

а) Dun & Bradstreet; б) LEXIS-NEXIS; в) Questel-Orbit; г) Dialog.

7. Количество профессиональных баз данных на мировом информационном рынке:

а) около 8 500; б) свыше 13 500; в) свыше 20 000.

8. Мировые информационные ресурсы делятся на следующие секторы:

а) деловой, научно-технической и специальной; потребительской информации;

б) биржевой, потребительской; научно-технической и специальной информации;

в) деловой, статистической, финансовой; потребительской информации;

9. Количество крупных информационных агентств-поставщиков на мировом рынке:

а) около 1500; б) более 4400; в) более 6000;

10. Какие семантические показатели оценки качества поиска в массивах документов вы знаете:

а) полнота выдачи и информационный шум;

б) полнота выдачи и достоверность информации;

в) пертинентность и релевантность.

11. Пертинентность – это

а) объективно существующее смысловое соответствие между содержанием документа и запроса;

б) субъективно оцениваемое соответствие содержание документов или текстов информационным интересам потребителя.

12. В ресурсах пространства WWW тесно связанные между собой web-страницы (центральное ядро) составляют

а) 27%

б) 22%

в) 7%

2.8. Вопросы к зачету

2.8.1. Что такое информация?

2.8.2. Понятие информационных продуктов и услуг

2.8.3. Классификация информационных продуктов и услуг

2.8.4. Рынок информационных продуктов и услуг

2.8.5. Этапы развития мирового рынка информационных услуг

2.8.6. Профессиональные базы данных

2.8.7. Информационные ресурсы Internet

2.8.8. Рынок финансовой информации

2.8.9. Ресурсы коммерческой информации

2.8.10. Эффективность поиска информации в Internet и профессиональных базах

2.8.11. Правовые основы информационной работы в РФ

2.8.12. Государственные информационные ресурсы.

2.8.13. Библиотечная сеть РФ.

2.8.14. Информационные ресурсы архивного фонда.

2.8.15. Статистическая информация РФ

- 2.8.16. Научно-техническая информация РФ.
- 2.8.17. Государственные системы правовой информации.
- 2.8.18. Ресурсы и услуги «Консультат Плюс»
- 2.8.19. Ресурсы и услуги «Гарант-Сервис»
- 2.8.20. Биржевая и финансовая информация РФ.
- 2.8.21. Классификация сетей.
- 2.8.22. Протокол ТСР/IP
- 2.8.23. Управление электронной почтой в сети масштаба предприятия
- 2.8.24. Средства факсимильной связи в масштабе предприятия
- 2.8.25. Сети с коммутацией пакетов (X.25)
- 2.8.26. Протокол передачи гипертекста (НТТР)
- 2.8.27. Поиск информации в Интернете
- 2.8.28. Поисковые системы Интернета
- 2.8.29. Электронная коммерция

КРИТЕРИИ ОЦЕНОК ЗНАНИЙ СТУДЕНТОВ

Зачтено

Студент выполнил все лабораторные работы и защитил отчеты, получил «положительные» оценки при сдаче всех тестов, защитил реферат, дал ответы на заданные вопросы к зачету, демонстрируя свободное владение основными понятиями и терминологией.

3. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

3.1. ОСНОВНАЯ ЛИТЕРАТУРА

1. Селетков С.Н., Хорошилов А. В. Мировые информационные ресурсы. СПб.: Питер, 2004. – 176 с.

2. Хорошилов А.В., Селетков С.Н., Днепровская Н.В. Управление информационными ресурсами // под редакц. А.В. Хорошилова. М.: Финансы и статистика, 2006. – 272 с.

3. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 2001, 668 с

4. Дунаев В. Самоучитель JavaScript. - СПб.: Питер, 2005. – 395 с.

3.2. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Э.А. Якубайтис. Информационные сети и системы. Справочная книга. М: "Финансы и статистика" 1996, 368 с

2. Стэн Шат Мир компьютерных сетей. К.:ВНУ, 1996, 288с

3. Андерсон К., Минаси М. Локальные сети. Полное руководство. К.: ВЕК+, М.:ЭНТРОП, СПб.:КОРОНА 1999, 624 с

4. Экономическая информатика / под ред П.В. Конюховского и Д.Н. Колесова. – СПб: Питер, 2000. 560 с.

3.3. ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ ДИСЦИПЛИНЫ:

Компьютерный класс ПЭВМ

Операционная система WINDOWS 98/2000, Internet Explorer 6.0 или 5.0.

4. УЧЕБНО-МЕТОДИЧЕСКАЯ (ТЕХНОЛОГИЧЕСКАЯ) КАРТА ДИСЦИПЛИНЫ

Номер недели	Вопросы, изучаемые на лекции	Занятия (номера)		Используй- мые нагляд. и метод. пособия	Самостоятельная работа студентов		Форма контроля
		Практич. (се- мин.)	Лабо- рат.		Содержание	ча- сы	
1	2.2.1		2.4.1		Подбор материалов по теме реферата	5	отчет
2	2.2.2				Подбор материалов по теме реферата	5	
3	2.2.2		2.4.2		Подбор материалов по теме реферата	5	отчет
4	2.2.2				Подбор материалов по теме реферата	5	
5	2.2.2		2.4.3		Подбор материалов по теме реферата	5	отчет
6	2.2.2				Подготовка текста реферата	5	
7	2.2.2		2.4.4		Подготовка текста реферата	3	отчет
8	2.2.2				Подготовка текста реферата	3	
9	2.2.3		2.4.4		Подготовка текста реферата	2	отчет
10	2.2.3				Защита реферата	4	
11	2.2.3		2.4.5				отчет
12	2.2.3				Доработка реферата	2	
13	2.2.3		2.4.5		2.6.2	2	отчет
14	2.2.3				2.6.2	2	
15	2.2.4		2.4.6	3.2.1	2.6.2	2	отчет
16	2.2.4				2.6.2.	2	отчет,
17	2.2.4		2.4.6		2.6.2.	2	отчет
18	2.2.8						отчет, 2.7.2

II. ГРАФИК ВЫПОЛНЕНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Для самостоятельной работы студентам предлагается написать реферат по теме (список тем находится в рабочей программе). Возможно и предложение студентом своей темы при условии согласования с преподавателем.

Студенческий реферат – письменная работа по определенному вопросу, написанная по определенному вопросу на основе периодической и учебной литературы.

Работа начинается с отбора и изучения материалов (учебники, монографии, журнальные статьи, источники из Интернета). Рекомендуются использовать источники не далее пятилетней давности издания.

Затем составляется план, и материал группируется по пунктам плана. План обязательно должен содержать введение, 2-3 основные пункты, заключение.

Во введение делается обоснование темы, ее значение и актуальность.

В основной части последовательно раскрывается тематика, стройно и логично.

В заключение делаются обобщения и выводы.

Реферат должен также содержать библиографический список.

Реферативная работа оформляется на листах формата А4 в соответствии с требованиями стандарта АмГУ, предъявляемыми к письменным работам.

Количество страниц реферата от 15 до 25.

Библиографический список должен содержать не менее четырех современных публикаций, расположенных в алфавитном порядке.

Работа над рефератом начинается с первой недели семестра. На десятой неделе проводится защита и при необходимости доработка реферата в течении двух недель

Кроме реферата студентам предлагается перечень тем, рекомендуемых для самостоятельного изучения:

Средства факсимильной связи в масштабе предприятия;

Управление электронной почтой в масштабе предприятия;

Сети с коммутацией пакетов (Х.25);

Поисковые системы Интернет.

Работа по их изучению проводится в течение семестра. Вопросы данного перечня включены в вопросы к зачету.

III. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ И ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы проводятся по подгруппам в компьютерном классе. Каждый студент получает индивидуальное задание в соответствии с вариантом. Выполняя задание, студент пользуется материалом, изученным в тексте лабораторной работы.

Перед созданием любой программы требуется точно продумать алгоритм. Записать его блок-схемой или словесно. Надо четко определить, что в нее требуется ввести и что получить в результате, в какой последовательности выполнять действия. В случае необходимости выделить циклические структуры и подпрограммы. В циклах четко определить параметры, задать их начальные значения, определить условия повторения и завершения цикла. В функциях определить количество передаваемых и возвращаемых значений.

При кодировании программы нужно определить тип используемых данных в зависимости от возможного диапазона принимаемых значений. При вводе величины не забывать осведомить об этом пользователя, а иногда сообщить и о типе, диапазоне или порядке ввода значений. Такое сообщение должно быть информативно и коротко. Вывод данных лучше сопровождать текстом и форматированием. Формат вывода можно уточнить при помощи модификаторов.

В именах переменных необходимо отражать их назначение, что повышает читаемость и понимание программы.

При записи сложных выражений нужно обращать внимание на приоритет операций. Текст программы лучше сопровождать краткими и информативными комментариями, что облегчает как понимание программы, так и ее отладку.

Объявление локальных переменных предпочтительнее по сравнению с глобальными.

Для отладки программы нужно запустить ее на выполнение несколько раз, задавая различные значения вводимых величин. Перед запуском необходимо иметь заранее подготовленные тестовые примеры, содержащие исходные данные и ожидаемые результаты. Их количество зависит от алгоритма. проверьте реакцию программы на заведомо неверные исходные данные.

Для быстрого поиска ошибки в алгоритме рекомендуется выводить промежуточные данные.

При сдаче лабораторной работы студент должен продемонстрировать преподавателю созданную программу, правильно работающую, отлаженную.

Преподаватель, принимая лабораторную работу, тестирует программу студента и задает ему вопросы по конструкциям, используемым в программе и теоретическим основам программирования на JavaScript.

IV. ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

Лабораторная работа 1 (2 часа) Основы JavaScript

Для выполнения программ JavaScript в качестве интерпретатора (исполнительной системы) можно использовать веб-браузер Internet Explorer 6.0 (5.0). В качестве редактора программ текстовый редактор, например Блокнот. Создавать программы на JavaScript можно и с помощью специальных программ, предназначенных для разработки веб-сайтов, например Microsoft FrontPage.

Можно легко создать и собственный редактор программ, например кодом:

```
<HTML>
<H3>Редактор кодов</H3>
код:<br>
<TEXTAREA id="mycode" ROWS=10 COLS=60></TEXTAREA>
<p>Результат:<br>
<TEXTAREA id="myresult" ROWS=3 COLS=60></TEXTAREA>
<P>
<BUTTON onclick="document.all.myresult.value=eval(mycode.value)">
Выполнить</BUTTON>
<BUTTON onclick="document.all.mycode.value=' ';
document.all.myrezalt.value=' ' ">
Очистить </BUTTON>
<P>
<!Комментарий>
Введите выражения в верхнее поле.
Выражения разделяются точкой с запятой.
Можно также каждое выражение писать в отдельной строке
</HTML>
```

Ввод-вывод данных

Для обеспечения ввода-вывода JavaScript предоставляет несколько методов:

alert(), confirm(), prompt().

Первый из перечисленных выводит на экран диалоговое окно с заданным сообщением и кнопкой ОК.

Синтаксис данного метода:

alert (сообщение)

Сообщение может представлять собой данные любого типа: последовательность символов, заключенную в кавычки, число, переменную или выражение.

Текст необходимо заключить в кавычки. Например,

alert (“Всем привет! ”)

Для формирования строк используют служебные символы:

\n – новая строка,

\t – табуляция,

\f – новая страница,

\b – забой,

\r – возврат каретки.

Окно, создаваемое alert() является модальным (останавливающим все последующие действия программы и пользователя). Его можно убрать, щелкнув по кнопке ОК.

Метод confirm выводит на экран диалоговое окно с сообщением и двумя кнопками – ОК и Отмена. Этот метод возвращает логическую величину, значение которой зависит от того, по какой из кнопок щелкнет пользователь. Возвращаемое значение можно обработать в программе, создавая тем самым интерактивный эффект. Синтаксис применения данного метода аналогичен синтаксису метода alert.

Окно, создаваемое confirm также является модальным.

Метод prompt осуществляет вывод диалогового окна с сообщением и кнопками ОК и Отмена, а также с текстовым полем, в которое пользователь может ввести данные. В отличие от alert() и confirm() данный метод принимает два параметра: сообщение и значение, которое должно появиться в текстовом поле ввода данных по умолчанию. Если пользователь щелкнет по кнопке ОК, метод вернет содержимое поле ввода данных, если – по кнопке Отмена, то возвращается значение ложь. Возвращаемое значение можно также обработать в программе. Синтаксис метода:

prompt (сообщение, значение_поля_ввода данных)

Оба параметра не являются обязательными. Если они не указаны, на экране появится окно без сообщения, а в поле ввода данных подставлено значение по умолчанию – undefined (не определено). Чтобы значение по умолчанию не появилось, в качестве второго параметра указывается пустая строка (“ ”).

Типы данных

Типы данных языка JavaScript приведены в таблице 1. При создании программ за типом данных следит сам программист. Интерпретатор не выдаст ошибки при неверном их использовании. Он просто попытается привести данные к типу, требуемому в данной операции.

Например, при написании выражения 7+ “нет” результатом будет строка символов “7нет”. Интерпретатор сначала переводит число в строку, а затем выполняет сложение двух строк, результатом которого в JavaScript является слияние двух строк. Результатом вычисления выражения 5+6 будет 11, а выражения 5+“6” – “56”.

<i>Тип данных</i>	<i>Описание значений</i>
Строковый или	Последовательность символов, заключенная в ка-

символьный тип (string)	вычки, двойные или одинарные
Числовой (number)	Положительное или отрицательное число. Целая и дробная части разделяются точкой
Логический	Два значения: true или false
Null	Отсутствие какого-либо значения
Объект (object)	Программный объект с собственными свойствами. В частности, массив также является объектом.
Функция (function)	Программный код, выполнение которого может возвращать некоторое значение.

Табл.1. Типы данных JavaScript

Для преобразования строк в числа предусмотрены встроенные функции `parseInt()` и `parseFloat()`. Синтаксис:

`parseInt(строка, основание)`
`parseFloat(строка, основание)`

Если основание не указано, то предполагается 10 – десятиричная система счисления. В качестве основания можно также использовать 8, 10, 16.

При преобразовании строки в целое число округление не происходит – дробная часть просто отбрасывается.

Примеры:

```
parseInt( "8.94")           // результат 8
parseFloat("8.94")         // результат 8.94
parseInt( "весна" )        // результат NaN – значение не определено
parseFloat( "весна" )      // результат NaN – значение не определено
parseInt( "15", 8 )        // результат 255
parseFloat( "17.5")        // результат 435
```

Для определения того, является ли значение выражения числом, служит встроенная функция `isNaN(значение)`. Функция возвращает логический тип. Данная функция считывает числа, данные числового типа и строки, содержащие только числа.

Переменные

Имя переменной представляет собой конечную последовательность символов, содержащую буквы, цифры, символ подчеркивания. Имя переменной не должно начинаться с цифры или содержать пробелы. Для имен переменных нельзя использовать ключевые слова языка.

В отличие от многих других языков программирования переменной не нужно задавать тип при объявлении. Тип переменной определяется типом ее значения. Переменная может принимать значения разных типов и неоднократно его изменять.

Создавать переменную в программе можно несколькими способами. Можно ей просто присвоить значение с помощью оператора присваивания в формате: `имя_переменной = значение`

Например,

```
Month= "Январь"
```

Можно использовать ключевое слово `var` перед именем переменной. В этом случае переменная не будет иметь первоначальное значение, но в дальнейшем его можно передать с помощью оператора присваивания. Например,

```
var Month  
Month = "Январь"
```

При использовании `var` допускается и инициализация переменной, например:

```
var Month = "Январь"
```

Для каждой переменной инициализация при объявлении возможна только один раз.

Можно сразу объявить несколько переменных, используя `var` и разделяя их запятой, при этом возможно инициализировать их все или некоторые:

```
var Month= "Январь", day, pi=3.14, x
```

Комментарии

В JavaScript допустимы два вида операторов комментария:

- одна строка символов, расположенная справа от `//`
- произвольное количество строк, заключенных между `/*` и `*/`.

Операции

В таблице 2 заданы основные операции, определенные в языке JavaScript.

Операция	Краткое описание
Унарные операции	
<code>++</code>	инкремент (увеличение на 1)
<code>--</code>	декремент (уменьшение на 1)
Бинарные операции	
<code>*</code>	умножение
<code>/</code>	деление
<code>%</code>	остаток от деления
<code>+</code>	сложение
<code>-</code>	вычитание
<code><</code>	меньше
<code><=</code>	меньше или равно
<code>>=</code>	больше или равно
<code>==</code>	равно
<code>!</code>	отрицание (не)
<code>&&</code>	логическое И
<code> </code>	логическое ИЛИ
<code>=</code>	присваивание
<code>*=</code>	умножение с присваиванием
<code>/=</code>	деление с присваиванием
<code>%=</code>	остаток от деления с присваиванием

+ =	сложение с присваиванием
- =	вычитание с присваиванием

Табл.2 Операции JavaScript

Операции выполняются в соответствии с приоритетами. Приоритет операций аналогичен языку C++. Для изменения порядка выполнения операций используются круглые скобки.

Операторы ветвления

Операторы ветвления сохраняют преемственность языка C++.

Условный оператор if используется для разветвления процесса вычислений на два направления. Синтаксис оператора if:

if (условие) оператор1 else оператор 2

Пример

```
if ( fvalue >= 0.0 ) fvalue = fvalue else fvalue = -fvalue
// вычисляется модуль произвольного числа.
```

Ветвь с ключевым словом else может отсутствовать. Например,

```
if ( f != 0 ) l = 100 / f
```

Если в какой-либо ветви требуется выполнить несколько операторов, их необходимо заключить в блок (операторные скобки { }), иначе компилятор не сможет определить окончание ветвления.

```
if ( a ) { a++ v=60*a } else v = a
```

```
if ( e == 1000 ) { e /= 10 alert(e) } else { e = 10 + y * y y++ }
```

Условные операторы могут быть вложенными.

```
if ( a < b ) { if ( a < c ) m = a else m = c } else { if ( b < c ) m = b else m = c }
```

Необходимо помнить, что в этом случае else относится к ближайшему if. Операторные скобки после первого if необязательны.

Если требуется проверить несколько условий, их объединяют знаками логических операций.

Оператор switch (переключатель) предназначен для разветвления процесса вычислений на несколько направлений. Синтаксис оператора:

```
switch ( выражение ) {
case константное выражение 1 : операторы1 break
case константное выражение 2 : операторы2 break
...
case константное выражение n : операторыN break
default : операторы }
```

Выполнение оператора начинается с вычисления выражения, а затем управление передается первому оператору из списка, помеченному константным выражением, значение которого совпало с вычисленным.

После этого, если выход из переключателя явно не указан (отсутствует break), последовательно выполняются все нижележащие ветви. Выход из переключателя обычно выполняется с помощью оператора break.

Если совпадения не произошло, выполняются операторы, расположенные

после ключевого слова default. Ветвь default может отсутствовать.

Пример. Печать названия месяца по порядковому номеру

```
switch (x) {  
  case 1 : alert( "Январь" );      break  
  case 2 : alert( "Февраль" );     break  
  case 3 : alert( "Март" );        break  
  ...  
  case 12 : alert( "Декабрь" );    break  
  default : alert( "Неверный номер" )  
}
```

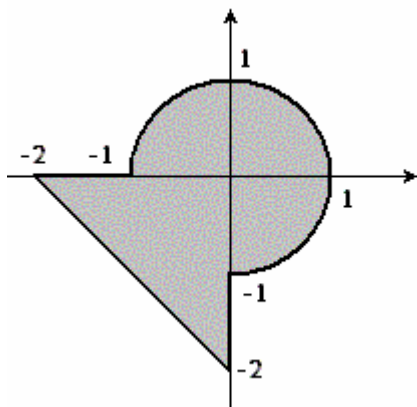
Параметр выражения оператора switch может принимать строковые, числовые и логические значения. В следующем примере переменная x содержит название языка, который выбрал пользователь. А выражение window.open () открывает новое окно браузера и загружает в него указанный в скобках HTML-документ.

```
switch (x) {  
  case " английский" : window.open ( "engl.htm" ); break  
  case " французский" : window.open ( "french.htm" ); break  
  case " русский" : window.open ( "russ.htm" );      break  
  default : alert( "Нет документа на таком языке" )  
}
```

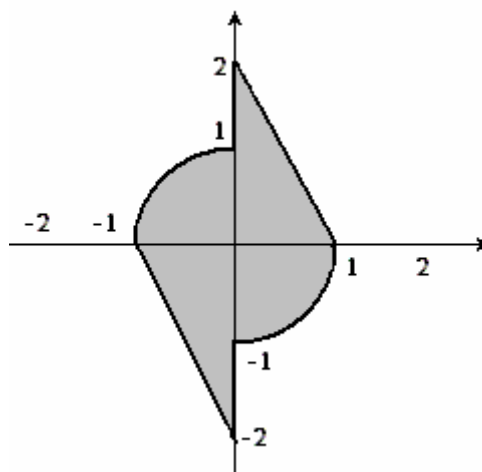
Задание

1. Создать собственный редактор программ, на основе HTML-кода, модифицировав код, приведенный выше. Файл открыть в браузере как веб-страницу.
2. Протестировать редактор, изучив в нем функции вывода и основные операции JavaScript.
3. Составить программу на основе разветвляющего алгоритма для задачи: Дана «мишень» в виде закрашенной области, изображенной на рисунке. Создать алгоритм для определения, попадает ли заданная точка с координатами (x, y) в мишень.
4. Для удобства пользователя рисунок мишени отобразить в редакторе.

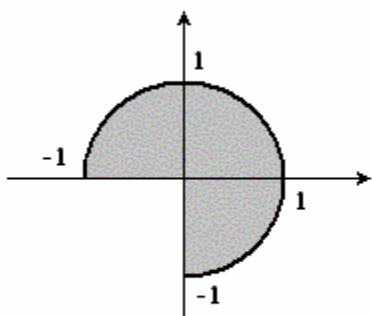
Вариант 1



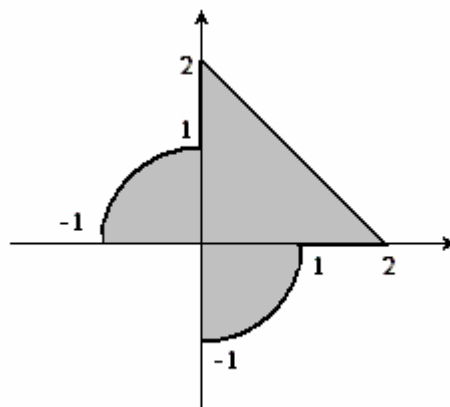
Вариант 2



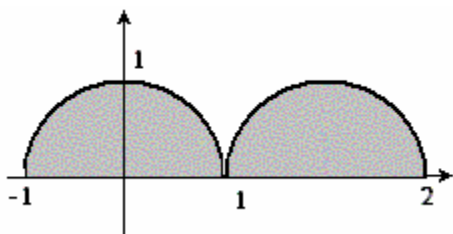
Вариант 3



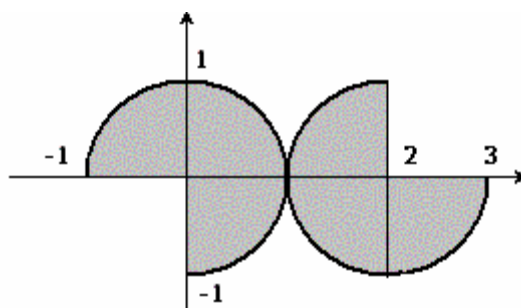
Вариант 4



Вариант 5



Вариант 6



Лабораторная работа №2 (4 часа)

Операторы цикла. Организация пользовательских функций

JavaScript содержит следующие встроенные функции (некоторые уже были рассмотрены):

`parseInt(строка, основание)` – преобразует указанную строку в целое число по указанному основанию (8, 10, 16); по умолчанию - десятичная система.

`parseFloat(строка, основание)` – преобразует строку в число с плавающей точкой.

`isNaN(строка, основание)` – возвращает `true`, если указанное в параметре значение не является числом.

`eval(строка)` – вычисляет выражение в указанной строке, выражение не должно содержать тегов HTML.

Например,

```
var a = 10; // значение a равно 10
var b = "if (a<=25) { a *= 2 }" // значение b равно строке символов
eval(b) // значение a равно 20
```

Другой пример применения данной функции в тексте программы редак-

тора. Там сценарии записаны в качестве значений атрибутов onclick, определяющих событие щелчок кнопкой мыши на HTML-кнопках, заданных тегами <button>.

escape(строка) – возвращает строку в виде %XX, где XX – ASCII-код указанного символа. Такую строку называют escape-последовательностью.

unescape(строка) – обратное преобразование.

typeof (объект) – возвращает тип указанного объекта в виде символьной строки, например “boolean”, “function”.

В программах на JavaScript пользователям также разрешено создавать собственные функции. Объявление функция состоит из заголовка и тела:

```
function имя_функции ( параметры )           //заголовок функции  
{ тело функции }
```

Тело функции ограничивается фигурными скобками. Параметры функции, стоящие в круглых скобках перечисляются через запятую.

Возвращение значений из функции происходит с помощью оператора return, за которым помещается само возвращаемое значение.

Для вызова функции можно воспользоваться выражениями вида:

```
имя_функции(параметры)   или  
имя_переменной = имя_функции(параметры)
```

Параметры в вызове функции должны быть представлены конкретными значениями.

Например, если описание функции дано в виде:

```
function cube (x)  
{ return x*x*x }
```

Данная функция вычисляет куб от параметра функции. Ее вызов может быть записан в виде:

```
y=cube(25)
```

В JavaScript функции могут вызываться как после их определения, так и до него. Можно не поддерживать соответствие количества параметров в определении функции и в ее вызове. Если в определении функции параметров больше, чем в вызове, то недостающим параметрам автоматически присваивается значение null. Лишние параметры в вызове функции игнорируются.

Внутри функции можно создавать переменные с помощью оператора присваивания или с помощью ключевого слова var.

Если в теле функции переменная в составе оператора присваивания встречается впервые в программе, или она была определена до этого – она действует как глобальная.

Если в теле функции используется переменная, объявленная только во внешней программе, она также является глобальной.

Если для определения переменной в теле функции используется ключе-

вое слово var, она будет локальной вне зависимости от того определена она во внешней программе или нет.

Операторы цикла

Как и другие языки программирования JavaScript имеет три вида оператора цикла: цикл с предусловием (while), цикл с постусловием (do while), цикл с параметром (for).

Синтаксис цикла с предусловием:

```
while ( условие )  
{  
операторы  
}
```

Выражение, стоящее в круглых скобках, определяет условие повторения тела цикла, представленного простым или составным оператором. Если оператор простой операторные скобки { } могут не ставиться.

Выполнение оператора цикла начинается с вычисления выражения. Если оно истинно, выполняется тело цикла. Если при первой проверке выражение ложно (false), цикл не выполнится ни разу.

Значение выражения вычисляется перед каждой итерацией цикла.

Пример. Программа вычисляет возведение 10 в степень 5.

```
var x =10  
y= 2  
while ( y <= 5 )  
{  
    x*=10; y++  
}
```

Цикл while обычно используется в тех случаях, когда число повторений заранее не известно. В этом случае используется и оператор цикла с постусловием. Его отличительной чертой является выполнение тела цикла хотя бы один раз. И только после первого его выполнения проверяется, надо ли его выполнять еще раз. Таким образом, даже если условие заведомо ложно цикл выполняется один раз. Если условие истинно, тело цикла выполнится еще раз. Цикл завершается, когда выражение станет равным false или в теле цикла будет выполнен оператор передачи управления.

Цикл for называют также циклом с заданным числом повторений. Он имеет следующий формат:

```
for (инициализация; выражение (условие); модификации )  
{  
операторы  
}
```

Инициализация используется для объявления и присвоения начальных значений величинам, используемым в цикле. Инициализация выполняется один

раз перед выполнением тела цикла.

Выражение определяет условие выполнения цикла: если его результат равен истине, то цикл выполняется. Цикл с параметром реализуется как цикл предусловием.

Модификации выполняются после каждой итерации цикла и служат обычно для изменения параметров цикла.

Тело цикла представляет собой простой или составной оператор.

```
for (i = 1, s=1; i<11; i++)
{
s*=i;           // вычисления факториала 10
}
```

Для принудительного выхода из тела любого цикла используются операторы break и continue. break позволяет переход в точку программы, находящуюся непосредственно за оператором, внутри которого находится, т.е. управление передается первой строке, следующей за телом цикла.

Например,

```
time=1
sum=0
while (time<10)
{ sum+=time
if ( sum>20 ) break;
time++
}
```

Инструкция continue заставляет программу пропустить все оставшиеся строки цикла, но сам цикл при этом не завершается. Для решения некоторых задач удобно комбинировать инструкции break и continue.

Задание

1. Составьте программу на основе циклического алгоритма для вычисления суммы ряда с заданной точностью ε . Определите и выведите на экран значение суммы и число элементов ряда, вошедших в сумму.

Номер варианта	Задание	Точность
1	$\frac{\pi}{3} + \sum_{n=1}^{\infty} (-1)^n \frac{(\pi/3)^{2n+1}}{(2n+1)!}$	$0.5 * 10^{-4}$
2	$\sum_{n=1}^{\infty} (-1)^n \frac{(\pi/6)^{2n}}{(2n)!}$	$0.5 * 10^{-4}$
3	$\sum_{n=1}^{\infty} \frac{(-1)^n (2n+1)}{(3n)!}$	$0.1 * 10^{-3}$
4	$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{3n^2}$	$0.1 * 10^{-2}$

5	$\sum_{n=1}^{\infty} \frac{(-1)^n (2n+1)}{n^3 (n+1)!}$	$0.1 * 10^{-2}$
6	$\sum_{n=1}^{\infty} \frac{(-1)^n n}{(2n-1)^2 (2n+1)^2}$	$0.1 * 10^{-2}$

2. Создать пользовательскую функцию, продемонстрировать ее работу в программе.

Вариант 1. Составить функцию вычисления факториала.

Вариант 2. Составить функцию, для вычисления $(a - b)^3$.

Вариант 3. Составить функцию нахождения максимума из трех чисел.

Вариант 4. Составить функцию нахождения минимума из трех чисел.

Вариант 5. Составить функцию проверки на равенство трех чисел

Вариант 6. Составить функцию для вычисления многочлена третьей степени.

Лабораторная работа №3 (2 часа)

Встроенные объекты JavaScript. Строки и массивы

Объекты представляют собой программные единицы, обладающие заданными свойствами. Как любой объект, объект JavaScript, обладает свойствами (данными) и методами (функциями) для их обработки. Программный код встроенных объектов JavaScript недоступен.

Управление web-страницами с помощью сценариев, созданных на JavaScript, заключается в использовании и изменении свойств объектов HTML-документа и самого браузера.

Встроенные объекты имеют фиксированные названия. Наиболее важными объектами в разработке web-сайтов являются String (символьные строки), Array (массивы), Math (математические формулы и константы), Date (работа с датами).

Объекты, названия которых совпадают с их фиксированными названиями, называются статическими. Можно создавать и экземпляры (копии) статических объектов, имеющие собственные имена и наследующие все свойства и методы статических объектов.

Встроенные объекты имеют прототипы (prototype), позволяющие добавлять новые свойства и методы к уже существующим в экземплярах объектов.

Объект String

С помощью объекта String можно создавать строку или строковый объект. Синтаксис:

имя_переменной = new String ("значение")

Создать строковый объект можно и с помощью обычного оператора присваивания:

имя_переменной = "значение"

или

```
var имя_переменной = “значение”
```

Например,

```
mystring1 = new String ( “строка”)  
mystring2 = “строка”
```

Свойство String

length – длина (количество символов), включая пробелы.

Доступ к свойствам и методам объекта осуществляется через операцию точка. Например,

```
str = “весна”  
str.length           // значение равно 5  
“лето”.length       // значение равно 4
```

Методы String

Как и любые функции, методы могут иметь параметры. Параметры перечисляются через запятую в круглых скобках, стоящих после имени метода.

charAt(индекс) – возвращает символ, занимающий в строке указанную позицию. Индекс является числом. Необходимо помнить, что нумерация элементов строки начинается с нуля.

Примеры

```
y = “Осень”.charAt(3)           // значение “н”  
str = “Зима”  
str.charAt (str . length – 2)  // значение “м”
```

charCodeAt(индекс) – преобразует символ в указанной позиции в его код. Поддерживает систему кодов Unicode, NN4 – ISO-Latin1.

fromCharCode(номер [1, номер2[, номер3, ..., номерn]]) – возвращает строку символов, числовые коды символов которой указаны в строке параметров.

concat(строка) – конкатенация (слияние) строк.

Синтаксис: строка1.concat(строка2)

Возвращает строку, полученную дописыванием символов строки2 к строке1.

Пример

```
x = “Петр”  
y = x .concat (“ Семенович”)   // результат “Петр Семенович”
```

indexOf(строка_поиска [, индекс]) - поиск строки, указанной параметром. Метод возвращает индекс первого вхождения строки. Поиск в пустой строке возвращает -1. Второй параметр, не является обязательным, о чем говоря квадратные скобки. Индекс указывает позицию, с которой начинается поиск.

lastIndexOf (строка_поиска [, индекс]) – поиск первого вхождения строки, указанной параметром. Причем поиск начинается с конца исходной строки, но возвращаемый индекс отсчитывается сначала.

localeCompare (строка) – сравнение строк в кодировке Unicode, то есть с

учетом используемого браузером языка общения с пользователем. Синтаксис:
строка1.localeCompare(строка2)

Если строки одинаковы, метод возвращает 0. Если строка1 меньше, чем строка2, метод возвращается отрицательное число, в противном случае - положительное.

slice(индекс1[, индекс2]) – возвращает подстроку исходной строки, начальный и конечный индексы которой указываются параметрами, за исключением последнего символа. Второй параметр не обязателен. Если он не указан – подразумевается до конца строки.

split(разделитель [, ограничитель]) – возвращает массив элементов, полученных из исходной строки. Первый параметр – строка символов, используемая в качестве разделителя строки на элементы. Второй параметр – число, указывающее количество элементов возвращаемого массива из строки, полученной при разделении. Второй параметр необязателен. Если разделитель – пустая строка, возвращается массив символов строки.

Например,

```
x = "Привет всем!"
```

```
x.split(" ") // значение – массив из элементов "Привет", "всем!"
```

```
x.split("e") // значение – массив из элементов "Прив", "т вс", "м!"
```

substr(индекс[, длина]) – возвращает подстроку исходной строки, начальный индекс и длина, которой указываются параметрами. Если второй параметр не указан, возвращается подстрока с начальной позиции до конца.

substring(индекс1, индекс2) – возвращает подстроку исходной строки, с позиции1 до позиции2.

toLocaleLowerCase(), toLowerCase() – переводят строку в нижний регистр.

toLocaleUpperCase(), toUpperCase() – переводят строку в верхний регистр.

Тексты web-страниц, как правило, создаются и форматируются с помощью тегов HTML. Это же можно сделать средствами JavaScript.

Например, для вывода строки полужирным шрифтом используют метод bold(). Данный метод не выводит строку в окно браузера, а лишь форматирует ее. Для вывода строки в HTML-документе используется метод write() для объекта document:

```
<HTML>  
<SCRIPT>  
st = "Доброе утро!".bold( )  
document.write(st)  
</SCRIPT>  
</HTML>
```

Методы форматирования строк носят названия, соответствующие тегам HTML:

anchor(“anchor_имя”)
blinc()
bold()
fixed()
fontcolor(значение цвета)
fontsize(число от 1 до 7)
italics()
link(расположение или URL)
big()
small()
strike()
sub()
sup()

Объект Array

Массив представляет собой упорядоченный набор данных. Нумерация элементов массива начинается с нуля. К элементам массива можно обращаться по их порядковому номеру, заключив его в квадратные скобки, расположенные после имени массива. Элементы массива в JavaScript могут быть разного типа.

Существует несколько способов создания массива:

1. имя_массива = new Array ([длина_массива])

Если длина массива не указана, создается пустой массив, не содержащий ни одного элемента. Иначе создается массив указанной длины, все элементы которого имеют значение null. Создав пустой массив, можно присвоить значения его элементам операцией присваивания.

2. инициализация массива при объявлении:

имя_массива = new Array (значение1[, значение2[, ... значенияn]])

3. инициализация каждого отдельного элемента массива, подобно свойствам объекта

```
имя_массива = new Array()  
имя_массива.имя_элемента1 = значение1  
[имя_массива.имя_элемента2 = значение 2  
[... имя_массива.имя_элемента n = значение n] ]
```

Например,

```
child = new Array (4)  
child[0]= “Женя”  
child[1]= 22  
child[2]= “ июнь”  
child[3]= 1996  
child = new Array ( “Женя”, 22, “ июнь”, 1996)  
y=child.length // y=4
```

```
child = new Array ()  
child.name = “Женя”  
child.day= 22
```



```
child.month= “ июнь”  
child.year= 1996
```

Свойства объекта Array

Свойство `length`, возвращает количество элементов объекта `Array`.

Свойство `prototype` позволяет добавлять новые свойства и методы для всех созданных массивов.

Например,

```
function SumNegative (massiv)
```

```
{var s = 0
```

```
  for (i=0; i<=massiv.length-1; i++)
```

```
    if (massiv[i]<0 s +=massiv[i]
```

```
  return s
```

```
}
```

```
mass = new Array(1, -9, 7, -23, -3)
```

```
Array.prototype.SumN = SumNegative //добавляем метод к объекту
```

```
Massiv.SumN(mass) // применяем метод SumN к массиву mass
```

Многомерные массивы

Для создания многомерного массива требуется указать все его размерности, заключив каждую из них в квадратные скобки.

Например,

```
matrix = new Array ( )
```

```
matrix[0] = new Array ( 2, 3, 8)
```

```
matrix[1] = new Array ( 1, -3, 7)
```

```
matrix[2] = new Array ( 0, 2, -6) // массив размерности 3 на 3
```

Методы объекта Array

`concat()` – объединяет два массива в третий и возвращает полученный массив.

Синтаксис: `имя_массива3 = имя_массива1.concat(имя_массива2)`

`join()` – создает строку из элементов массива с указанным разделителем между ними, возвращает строку символов.

Синтаксис: `имя_массива2 = имя_массива1.join(строка)`

`pop()` – удаляет последний элемент массива и возвращает его значение.

Синтаксис: `имя_массива1.pop()`

`push()` – добавляет к массиву последний элемент, значение которого указано в качестве параметра и возвращает новую длину массива.

`shift()` – удаляет первый элемент массива и возвращает его значение.

`unshift()` – добавляет к массиву первый элемент, значение которого указано в качестве аргумента.

`reverse()` – переписывает массив в обратном порядке, возвращает массив.

`slice(индекс1[, индекс2])` – создает массив из элементов исходного массива с индексами указанного диапазона. Возвращает массив. Если второй индекс

не указан, то новый массив создается из элементов с индекса 1 до конца исходного массива.

`sort()` – упорядочивает элементы массива. Если параметр не указан, сортировка производится на основе ASCII-кодов символов значений, что удобно для строк, но не подходит для чисел. Параметром может служить имя функции сравнивающей два элемента массива.

Пример,

```
massiv = new Array (7, 1, 34, 5, 63)
```

```
function cmp (x, y)
```

```
{ return x - y }
```

```
massiv.sort(cmp) //массив будет сортироваться по возрастанию
```

Эта функция дает критерий сортировки.

`splice(индекс, количество [, элем1[, элем2[, ... элемn]]])` – удаляет (заменяет) из массива элементы. Возвращает массив из удаленных элементов. Первый параметр является индексом первого удаляемого элемента, второй - количеством удаляемых элементов. Если указаны необязательные параметры, то происходит замена элементов указанного диапазона на указанные значения параметров. Но это справедливо, если второй параметр не равен нулю.

Пример

```
b = new Array( "один", 2, 3, 4, "пять")
```

```
c = b.splice(1, 3, "два", "три", "четыре")
```

```
// массив b из элементов «один», «два», «три», «четыре», «пять»
```

```
// массив c из элементов 2, 3, 4
```

`toLocaleString()`, `toString()` – преобразуют содержимое массива в символьную строку.

Задание

1. Создайте пользовательскую функцию для работы со строками с использованием методов объекта `String`.

Вариант 1.

Функция вставки строки в исходную строку. Функция должна иметь три параметра: исходную строку, вставляемую строку и позицию вставки.

Вариант 2.

Функция замены в исходной строке все вхождения заданной подстроки на подстроку замены. Функция должна иметь три параметра: исходную строку, заменяемую подстроку и подстроку, которой следует заменить все вхождения заменяемой подстроки.

Вариант 3.

Функция удаления лишних пробелов в начале исходной строки.

Вариант 4.

Функция удаления лишних пробелов в конце строки.

Вариант 5.

Функция удаления слов, длина которых меньше заданного размера. Функция

должна иметь два параметра, исходную строку и длину удаляемых слов.

Вариант 6.

Функция удаления в строке одинаковых слов.

2. Изучите функции форматирования строк. Продемонстрируйте работу функции из задания 1, отформатировав вновь полученную строку тремя различными способами в HTML-документе.

3. Создайте функцию для работы с объектом Array.

Вариант 1.

Замена минимального элемента значением, заданным как параметр функции.

Вариант 2.

Сортировка по возрастанию элементов массива, расположенных между максимальным и минимальным его элементами.

Вариант 3.

Сортировка по убыванию элементов массива, расположенных до максимального значения.

Вариант 4.

Удаление из массива минимального элемента, и добавление утроенного найденного значения в качестве первого элемента.

Вариант 5.

Удаление максимального элемента массива и добавление найденного значения, помноженного на пять, в качестве последнего элемента массива.

Вариант 6.

Создание массива из элементов исходного, расположенных между максимальным и минимальным его элементами.

4. Создайте 2 массива. Добавьте созданную функцию из задания 3 в качестве нового свойства ко всем созданным массивам.

Лабораторная работа № 4 (2 часа)

Работа с окнами

Главное окно браузера создается автоматически при запуске браузера. С помощью сценария можно создать любое количество окон, а также разбить окно на несколько прямоугольных областей, называемых фреймами. Окну браузера соответствует объект `window`, а HTML-документу, загруженному в окно, соответствует объект `document`. Эти объекты могут содержать в себе другие объекты. В частности, объект `document` входит в состав `window`.

Доступ к свойствам и методам данного объекта происходит, как и в других объектах, через точку. Поскольку объект `document` является подобъектом объекта `window`, ссылка на HTML-документ, загруженный в текущее окно: `window.document`. Объект `document` имеет метод `write` (запись строки в текущий HTML-документ).

Для его применения используют `window.document.write(строка)`. Объект окна `window` - корневой объект, имеющий свои подобъекты. Например, `location`

хранит информацию об URL-адресе загруженного документа, screen – данные о возможностях экрана монитора пользователя.

В объектной модели документа объекты сгруппированы в *коллекции*. Коллекция – промежуточный объект, содержащий объекты собственно документа. Коллекция является упорядоченным массивом объектов, отсортированных в порядке упоминания соответствующих им элементов в HTML-документе. Индексация объектов в коллекции начинается с нуля. Синтаксис обращения к элементам коллекции аналогичен синтаксису обращению к элементам массива. Коллекция имеет длину – свойство length.

Коллекция всех графических изображений документа называется images, коллекция всех форм – forms, ссылок – links. Коллекция всех объектов документа называется all.

Один и тот же объект может входить в частную коллекцию (например, images), но он обязательно входит в коллекцию all. При этом его индексы могут быть разными в разных коллекциях.

При использовании документа, загруженного в текущее окно, объект window можно не упоминать, а сразу начинать с объекта document.

Например,
document.images(0)

Вместо индекса можно использовать значение атрибута ID в теге, который определяет соответствующий элемент HTML-документа.

Однако универсальный способ обращения к объектам документа – обращение посредством коллекции all.

С помощью сценария можно создавать любое количество окон. Для этого применяется метод open():

window.open(параметры)

Данному методу передаются следующие необязательные параметры:

адрес документа, который нужно загрузить в создаваемое окно;

имя окна (как имя переменной);

строка описания свойств окна (features).

В строке свойств записываются пары свойство = значение, которые отделяются друг от друга запятыми.

Свойства, передаваемые в строке features

Свойство	Значения	Описание
channel mode	yes, no, 1, 0	Показывает элементы управления channel
directories	yes, no, 1, 0	Включают кнопки каталога
full screen	yes, no, 1, 0	Полностью разворачивает окно
height	число	Высота окна в пикселях
left	число	Положение по горизонтали относительно левого края экран в пикселях
location	yes, no, 1, 0	Текстовое поле Address
menubar	yes, no, 1, 0	Стандартное меню браузера
resizeable	yes, no, 1, 0	Возможность пользователя изменять размер окна
scrollbars	yes, no, 1, 0	Горизонтальные и вертикальные полосы прокрут-

status	yes, no, 1, 0	ки Стандартная строка состояния
toolbar	yes, no, 1, 0	Включает панели инструментов браузера
top	число	Положение по вертикали относительно верхнего края экрана в пикселях
width	число	Ширина окна в пикселях

Примеры

```

window. open ( "mypage.htm", "NewWin", "height=150, width=300")
window. open ( "mypage.htm" )
strfeatures = "top=100, left=15, height=250, width=300, location=no"
window. open ( "www.amsu.ru", strfeatures)

```

Вместо строки `strfeatures` можно использовать значение `true`, тогда указанный документ загружается в существующее окно, вытесняя предыдущий документ.

Метод `window.open()` возвращает ссылку на объект окна, сохранив которую, можно использовать позднее, например, при закрытии окна.

Для закрытия используют метод `close()`. Однако, выражение `window.close()` закрывает главное окно. Для закрытия других окон используют ссылки.

Пример

```

var str = window. open ( "mypage.htm", "моя страница")
str.close()

```

Объект `document` является центральным в иерархической объектной модели. Он предоставляет всю информацию о HTML-документе с помощью коллекций и свойств и множество методов для работы с документами.

Коллекция document

<code>all</code>	все теги и элементы основной части документа
<code>anchor</code>	якоря (закладки) документа
<code>applets</code>	все объекты документа, включая встроенные элементы управления, графические элементы, апплеты, внедренные объекты
<code>embeds</code>	все внедренные объекты документа
<code>forms</code>	все формы на странице
<code>frames</code>	фреймы, определенные в теге <code><FRAMESET></code>
<code>images</code>	графические элементы
<code>links</code>	ссылки и блоки <code><AREA></code>
<code>plugins</code>	другое название внедренных документов
<code>scripts</code>	все разделы <code><SCRIPT></code> на странице
<code>styleSheets</code>	контейнерные свойства стиля, определенные в документе

Методы document

<code>clear</code>	очищает выделенный участок
--------------------	----------------------------

close	закрывает текущее окно браузера
createElement	создает экземпляр элемента для выделенного тега
elementFromPoint	возвращает элемент с заданными координатами
execCommand	выполняет команду над выделенной областью
open	открывает документ
queryCommandEnabled	сообщает, доступна ли данная команда
queryCommandIndeterm	сообщает, если данная команда имеет неопределенный статус
queryCommandState	возвращает текущее состояние команды
queryCommandSupported	сообщает, поддерживается ли данная команда
queryCommandText	возвращает строку, с которой работает команда
queryCommandValue	возвращает значение команды, определенное для документа или объекта TextRange
write (writeln)	записывает текст и код HTML в документ, находящийся в указанном окне

Свойства document

Свойство	Атрибут	Назначение
activeElement		Активизирует активный элемент
alinkColor	ALINK	Цвет ссылок на странице
bgColor	BGCOLOR	Определяет цвет фона элемента
body		Ссылка только для чтения на неявный основной объект документа, определенный в теге <BODY>
cookie		Строка cookie-записи. Значение этого свойства приводит к записи на диск.
domain		Устанавливает или возвращает домен документа для его защиты или идентификации.
fgColor	TEXT	Устанавливает цвет текста переднего плана
lastModified		Дата последней модификации страницы, доступна как строка
linkColor	LINK	Цвет еще непосещенных гиперссылок на странице
location		Полный URL документа
parentWindow		Возвращает родительское окно для документа
readyState		Определяет текущее состояние загружаемого объекта
referrer		URL страницы, которая вызвала текущую
selection		Ссылка только для чтения на дочерний для document объект selection
title	TITLE	Определяет справочную информацию элемента, используемую при загрузке и всплывающей подсказке
url	URL	URL-адрес документа клиента или в теге <META>
vlinkColor	VLINK	Цвет посещенных ссылок на странице

Объект window кроме дочерних объектов имеет свои методы, свойства, события.

Свойства window:

parent	возвращает родительское окно для текущего
self	возвращает ссылку на текущее окно
top	возвращает ссылку на главное окно
name	название окна
opener	окно, создаваемое текущим
closed	сообщает, если окно закрыто
status	текст, показываемый в строке состояния браузера
defaultStatus	текст по умолчанию строки состояния браузера
returnValue	позволяет определить возвращаемое значение для события или диалогового окна
client	ссылка, возвращаемая объект навигатора браузеру
document	ссылка только для чтения на объект окна document
event	ссылка только для чтения на глобальный объект event
history	ссылка только для чтения на объект окна history
location	ссылка только для чтения на объект окна location
navigator	ссылка только для чтения на объект окна navigator
screen	ссылка только для чтения на объект окна screen

Например,

window.status = «работает сценарий»

Свойство parent позволяет обратиться к объекту, расположенному в иерархии на одну ступень выше. Для перемещения на две ступени выше используют parent.parent. Для обращения к самому главному окну – окну браузера, используют свойство top.

Свойство status используют для вывода сообщений во время работы сценария. Например, window.status = “сценарий работает”

Методы window

open()	открывает новое окно браузера
close()	закрывает текущее окно браузера
showHelp()	показывает окно подсказки как диалоговое
showModalDialog()	показывает новое модальное(диалоговое) окно
alert()	окно предупреждения с сообщением и кнопкой ОК
prompt()	окно приглашения с сообщением, текстовым полем и кнопками ОК и Cancel (Отмена)
confirm()	окно подтверждения с сообщением и кнопками ОК и Cancel
navigate()	загружает другую страницу с указанным адресом
blur()	убирает фокус с текущей страницы
focus()	устанавливает страницу в фокус
scroll()	разворачивает окно на заданную ширину и высоту

setInterval()	указывает процедуре выполняться автоматически через заданное число миллисекунд
setTimeout()	запускает программу через заданное количество миллисекунд после загрузки страницы
clearInterval()	обнуляет таймер, заданный методом setInterval()
clearTimeout()	обнуляет таймер, заданный методом setTimeout()
execScript()	выполняет код сценария, по умолчанию Jscript

Рассмотренные выше методы позволяют работать с независимыми (немодальными) окнами. Для создания модального окна используется метод showModalDialog(). В качестве параметра данный метод принимает адрес документа (файла), имя окна, и строку свойств.

Свойства модального окна

Свойство	Значения	Описание
order	thick, thin	Размер рамки вокруг окна (толстая/тонкая)
center	yes, no, 1, 0	Выравнивание окна по центру главного
dialogHeight	число + единицы измерения	Высота окна
dialogLeft	число + единицы измерения	Горизонтальная координата
dialogTop	число + единицы измерения	Вертикальная координата
dialogWidth	число + единицы измерения	Ширина окна
font	строка таблицы стилей	Стиль окна, определенный по умолчанию
font-family	строка таблицы стилей	Вид шрифта, определенный по умолчанию для окна
font-size	строка таблицы стилей	Размер шрифта, определенный по умолчанию для окна
font-style	строка таблицы стилей	Тип шрифта, определенный по умолчанию для окна
font-variant	строка таблицы стилей	Вариант шрифта, определенный по умолчанию для окна
font-weight	строка таблицы стилей	Толщина шрифта, определенная по умолчанию для окна
help	yes, no, 1, 0	Включение кнопки Help в верхнюю панель
maximize	yes, no, 1, 0	Включение кнопки Maximize в верхнюю панель
minimize	yes, no, 1, 0	Включение кнопки Minimize в верхнюю панель

При работе с модальными окнами пользователь не может обратиться к другим окнам, в том числе и к главному. Окна, создаваемые методами alert(), prompt(), confirm() являются модальными.

Одним из главных назначений сценариев в HTML-документе является обработка событий, таких как щелчок кнопки мыши по элементу документа, помещение указателя мыши на элемент, нажатие клавиши и др. Для одного и того же элемента можно определить несколько событий на которые он будет реагировать.

Сообщение о событии формируется в виде объекта, т.е. контейнера для хранения информации. Объект события в одном из свойств содержит ссылку на элемент, с которым связано данное событие (на кнопку, изображение и т.п.)

Обычно обработчики событий оформляются в виде функций, определения которых помещаются в контейнерный тег <SCRIPT>.

События window

onblur	выход окна из фокуса
onfocus	окно становится активным
onhelp	нажатие пользователем клавиши F1
onresize	изменение пользователем размеров окна
onscroll	прокрутка окна пользователем
onerror	ошибка при передаче
onbeforeunload	для сохранения данных перед выгрузкой страницы
onload	страница полностью загружена
onunload	непосредственно перед выгрузкой страницы

В случае открытия нескольких окон браузера, пользователь может переключаться между ними, переводя фокус с одного окна на другое. Эти действия инициируются программными событиями `onblur` и `onfocus`. Эти же действия можно вызвать, используя методы `blur` и `focus`.

Событие `onerror` происходит при ошибке загрузки страницы или ее элемента. Его можно использовать в программе при попытке вновь загрузить страницу. Например,

```
<SCRIPT>
function window.onerror( ) {
  alert (“ Ошибка! Повтори попытку!”)
}
</SCRIPT>
```

События document

onafterupdate	окончание передачи данных
onbeforeupdate	перед выгрузкой страницы
onclick	при щелчке левой кнопкой мыши
ondblclick	при двойном щелчке левой кнопкой мыши
ondragstart	при возникновении перетаскивания
onerror	ошибка при передаче
onhelp	нажатие клавиши F1
onkeydown	нажатие клавиши

onkeypress	возникает при нажатии клавиши и продолжается при удержании клавиши в нажатом состоянии
onkeyup	пользователь отпускает клавишу
onload	при полной загрузке документа
onmousedown	при нажатии кнопки мыши
onmousemove	при перемещении указателя мыши
onmouseout	когда указатель мыши выходит за границы элемента
onmouseover	когда указатель мыши входит на документ
onmouseup	пользователь отпускает кнопку мыши
onreadystatechange	возникает при изменении свойства readystate
onselectstart	когда пользователем впервые запускается выделенная часть документа

Динамическое изменение элементов документа

Элементы HTML-документа задаются тегами, большинство из которых имеют параметры (атрибуты). В объектной модели документа тегам соответствуют объекты, а атрибутам – свойства этих объектов. Названия свойств объектов, как правило, совпадают с названиями атрибутов, но записываются в нижнем регистре.

Наиболее удобный способ динамического изменения HTML-документа основан на использовании свойств `innerText`, `outerText`, `innerHTML` и `outerHTML`. С их помощью можно получить доступ к содержимому элемента. Изменяя значения перечисленных свойств можно частично или полностью изменить сам элемент. Например, можно изменить только надпись на кнопке, а можно превратить кнопку в изображение или Flash-анимацию.

Значением свойства `innerText` является все текстовое содержимое между открывающим и закрывающим тегами элемента. Внутренние теги игнорируются. Данные открывающего и закрывающего тегов соответствующего элемента также не входят.

В отличие от предыдущего свойство `outerText` включает в себя данные открывающего и закрывающего тегов. Таким образом, `outerText` есть весь текст, содержащийся в контейнере, включая его внешние теги. Например, задан HTML-код:

```
<DIV ID = "my" >
<A HREF = 'raznoe.htm'>
<IMG SRC = 'picture.jpg' > Ссылка на раздел <B> Разное </B>
</A>
</DIV>
```

Здесь свойства `innerText` и `outerText` для элемента, заданного контейнерным тегом `<DIV>`, совпадают:

```
document.all.my.innerText //значение равно – «Ссылка на раздел Разное»
```

При присвоении свойствам `innerText` и `outerText` новых значений нужно помнить, что если значения содержат теги, то они не интерпретируются, а воспринимаются как обычный текст.

Свойство `innerHTML` содержит внутренний HTML-код контейнера элемента. Присвоение этому свойству нового значения, содержащего HTML-код, приводит к интерпретации кода. Свойство `outerText` дополнительно включает внешние открывающие и закрывающие теги элемента.

Для приведенного HTML-кода значение `document.all.my.innerHTML` равно “ Ссылка на раздел Разное ”/

Значение `document.all.my.outerHTML` – “<DIV ID = “my” > Ссылка на раздел Разное </DIV>”.

Если в сценарии выполнить выражение `document.all.my.innerHTML = “<BUTTON>Щелкни здесь</BUTTON>”` ссылка, изображение и текст будут заменены кнопкой с надписью «Щелкни здесь». При этом контейнерный тег “<DIV ID = “my” > сохранится. Если аналогичным образом использовать `outerHTML`, кнопка также появится, но уже без контейнера “<DIV ID = “my” >”.

Свойства `innerHTML` и `outerHTML` могут применяться к элементам, заданным неконтейнерными тегами. Тогда `innerHTML` и `outerHTML` совпадают.

Для ускорения загрузки графики можно использовать следующие возможности JavaScript. Можно организовать предварительную загрузку изображений в кэш-память браузера, не отображая их на экране. Это особенно эффективно при начальной загрузке страницы. Пока изображения загружаются в память, оставаясь невидимыми, пользователь может рассматривать текстовую информацию.

Для предварительной загрузки изображения требуется создать его объект в памяти браузера. Это можно сделать следующим выражением:

```
myimg = new Image (ширина, высота)
```

Параметры должны соответствовать значениям атрибутов `WIDTH` и `HEIGHT` тега ``, который используется для отображения предварительно загруженного изображения.

Для созданного в памяти объекта изображения можно создать имя или URL-адрес графического файла:

```
myimg.src = “URL-адрес изображения”
```

что предписывает браузеру загрузить изображения без его отображения.

После загрузки в кэш-память всех изображений и загрузки всего документа можно сделать их видимыми. Для этого свойству `src` элемента `` нужно присвоить значение этого же свойства объекта изображения в кэш-памяти. Например,

```
document.images[0].src = myimg.src
```

Здесь слева указано свойство `src` первого в документе элемента, соответствующего тега ``, справа – свойство `src` объекта изображения в кэш-памяти.

С помощью JavaScript можно через заданный интервал времени запускать код или функцию. При этом создается эффект одновременного (параллельного) выполнения вычислительных процессов.

Для организации повторения через заданный интервал выполнения некоторого выражения служит метод `setInterval()` объекта `window`:

```
setInterval( выражение, период, [, язык])
```

Первым параметром является строка, например вызов функции. Период указывается в миллисекундах. Третий параметр – необязательный, в котором указывается язык с помощью которого написано заданное выражение. По умолчанию – JavaScript.

Метод `setInterval()` возвращает некоторое целое число – идентификатор временного интервала, который может быть использован в дальнейшем, например для прекращения выполнения процесса методом `clearInterval()`. Например,

```
var pr = setInterval( "myfunc(), 100" )  
if (confirm ( "Прервать процесс?" ) )  
clearInterval(pr)
```

Если требуется выполнить действие с некоторой временной задержкой, используется метод `setTimeout()`, имеющий синтаксис аналогичный `setInterval()`. Для отмены задержки процесса, запущенного `setTimeout()`, используют `clearTimeout()`.

Задание

Создать HTML-документ, расположив в нем список названий графических объектов, одно исходное отображение, две кнопки.

Щелчок на элементе списка должен приводить к изменению цвета элемента списка и отображению соответствующего графического элемента, и соответствующего ему тестового сопровождения.

При этом изображение кнопки должно быть также изменено.

Щелчок по первой кнопке через 5 секунд должен инициализировать функцию открытия документа в окне, заданного размера, определенного размера текстового поля и название. Окно должно содержать горизонтальные и вертикальные полосы прокрутки, размер окна не должен изменяться по желанию пользователя. Выведенный в окне текст должен быть синим на сером фоне, иметь выделенный заголовок, ссылки на другие объекты. По выбору продемонстрируйте по пять событий и свойств объектов `window` и `document`.

Это действие может быть отменено с помощью второй кнопки.

Лабораторная работа №5 (2 часа)

Работа с фреймами

Фрейм – прямоугольная область окна браузера, в которую можно загрузить HTML-документ. Разбиение окна браузера на отдельные окна производится с помощью тега `<FRAMESET>`, внутрь которого вставляются теги `<FRAME>` с атрибутами, указывающими имя фрейма и адрес HTML-документа.

Пример
<HTML>

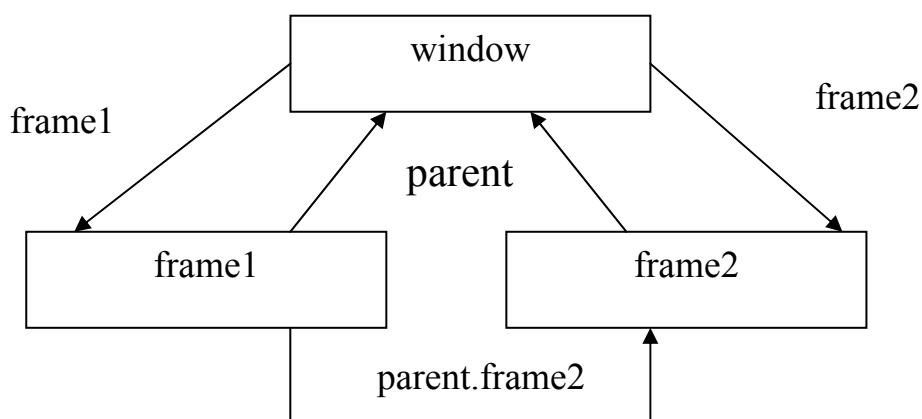
```

<FRAMESET ROWS= "30%, 70%">
<FRAMESET SRC= "документ1.htm" NAME = "frame1" >
<FRAMESET SRC= "документ2.htm" NAME = "frame2" >
</FRAMESET>
</HTML>

```

Здесь применяется вертикальное расположение фреймов. Для горизонтального размещения фреймов вместо атрибута ROWS в теге <FRAMESET> использовать COLS.

Используя вложение тега <FRAMESET>, можно разбить уже имеющийся фрейм на два других.



При разбиении окна на фреймы и, в свою очередь, фрейма на другие фреймы возникают отношения родитель-потомок. Каждому из фреймов соответствует свой объект document. Обеспечение доступа к в иерархии объектов представлено на рисунке.

Так при обращении из одного фрейма-потомка к другому, необходимо помнить, что прямой связи между фреймами-потомками не существует. Поэтому сначала нужно обратиться к родительскому окну, а затем к его второму потомку:

```
parent.frame2.document.write(" Привет от первого фрейма.")
```

Можно изменить элемент одного фрейма из другого. Например, при щелчке на тексте в правом фрейме в левом изменится один из текстовых документов. Тогда документ в левом фрейме с именем LEFT:

```

<HTML>
Делай раз<BR>
Делай два
<H1 ID = "XXX"> Делай три </H1>
</HTML>

```

Документ в правом фрейме:

```

<HTML>
<SCRIPT>
function change( ) {
parent.LEFT.document.all.XXX.innerHTML = "Делай пять!!!!"

```

```

}
</SCRIPT>
<H1 onclick = "change( )"> Щелкни здесь</H1>
</HTML>

```

В теле функции `change()` происходит обращение к левому фрейму с именем `LEFT` (задается в установочном HTML-файле) через `parent`. Изменение элемента происходит за счет присвоения значения свойству `innerText`. Кроме данного свойства можно использовать `outerText`, `innerHTML` или `outerHTML`.

Важно, что изменения в одном фрейме по событию в другом происходят без перезагрузки HTML-документа.

Фреймы удобно использовать при создании навигационных панелей. В одном фрейме располагаются ссылки, а второй предназначен для отображения документов, вызываемых при активизации соответствующих ссылок.

Пример

```

// установочный файл frame.htm
<HTML>
<FRAMESET COLS = "25%, 75%">
<FRAME SRC = "menu.htm" NAME = "menu" >
<FRAME SRC = "start.htm" NAME = "main" >
</FRAMESET>
</HTML>

```

Здесь `start.htm` – документ, который первоначально показан во фрейме `main`.

```

//menu.htm – навигационная панель
<HTML>
<SCRIPT>
function load (url) {
parent. main. location. href = url;
}
</SCRIPT>
<BODY>
<A HREF = "javascript:load('первый.htm')">Первый </A>
<A HREF = "второй.htm" TARGET = "main"> Второй </A>
<A HREF = "третий.htm" TARGET = "top"> Третий </A>
</BODY>
</HTML>

```

В примере окно браузера разделено на два фрейма. Первый из них играет роль навигационной панели, а второй – окна для отображения документов. Продемонстрированы два способа загрузки новой страницы во фрейм `main`. В первом случае используется функция `load()`, параметр которой указывает, какой файл следует загрузить. При этом место, в которое он загружается, определяется самой функцией `load()`. Во второй ссылке используется атрибут

TARGET. В третьей ссылке демонстрируется, как можно избавиться от фреймов.

Для удаления фрейма с помощью load() достаточно записать:

```
parent.location.href = url
```

Атрибут TARGET в теге ссылки <A HREF> обычно применяется в случаях, когда требуется загрузить одну страницу в один фрейм. Язык сценариев используют при необходимости выполнения нескольких действий.

Для ссылок из родительского окна к объектам его дочерних фреймов можно использовать коллекцию frames. Обращение к определенному фрейму из этой коллекции возможно по индексу или по имени фрейма:

```
window.frames [индекс]
```

```
window.имя_фрейма
```

При обращении к объекту документа, загруженного во фрейм, следует сначала упомянуть объект document:

```
window.frames(0).document.all.Myinput.Value
```

```
window.LEFT.document.all.Myinput.Value
```

Ссылка из дочернего фрейма на родительский - осуществляется с использованием parent.

При использовании top следует учитывать, что создаваемый сайт может быть загружен в другой. Тогда объект top окажется объектом другого сайта. Поэтому лучше использовать parent для ссылок на вышестоящее окно или фрейм.

Ссылки top или self используют для предотвращения отображения сайта внутри фреймов другого сайта. Сценарий, выполняющий это, следует разместить в начале документа, например:

```
<SCRIPT>
```

```
if (top != self )
```

```
    top.Location = location
```

```
</SCRIPT>
```

т.е., ссылка на свойство top на верхнее окно, должна совпадать со ссылкой self на текущее окно.

Для вставки одного HTML-документа в тело другого средствами браузера служит контейнерный тег <IFRAME>:

```
<IFRAME SRC = "адрес документа" > </IFRAME>
```

Данный элемент представляет собой прямоугольную область с прокруткой или без. Такое окно называют плавающим фреймом. Данный документ можно позиционировать с помощью параметров таблицы стилей (тег <STYLE> или атрибут STYLE).

Плавающий фрейм аналогичен обычному фрейму. При создании он помещается в коллекцию frames. Среди его свойств широко используется align – выравнивание плавающего фрейма относительно окружающего содержимого документа. Его возможные значения:

absbottom – выравнивает нижнюю границу фрейма по подстрочной линии символов окружающего текста,

absmiddle – выравнивает середину границу фрейма по центральной линии между top и absbottom окружающего текста,

baseline – выравнивает нижнюю границу фрейма по базовой линии окружающего текста,

bottom – совпадает с baseline (только IE)

left – выравнивает фрейм по левому краю элемента-контейнера,

middle – выравнивает воображаемую центральную линию окружающего текста по воображаемой центральной линии фрейма,

right – выравнивает фрейм по правому краю элемента-контейнера,

texttop – выравнивает верхнюю границу фрейма по надстрочной линии символов окружающего текста,

top – выравнивает верхнюю границу фрейма по верхней границе окружающего текста.

Задание

Вариант 1. Окно браузера поделить на два фрейма. В левом расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в левом окне по щелчку мыши на миниатюре изображения в правом фрейме. В левом фрейме дополнительно организовать навигационную модель. Создать новый HTML-документ и вставить его в ранее созданный, как плавающий вертикальный фрейм, выравнивая нижнюю границу фрейма по базовой линии окружающего текста.

Вариант 2. Окно браузера поделить на три фрейма. В левом расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в среднем окне по щелчку мыши на миниатюре изображения в правом фрейме. В левом фрейме дополнительно организовать навигационную модель. Создать новый HTML-документ и вставить его в ранее созданный, как плавающий вертикальный фрейм, выравнивая воображаемую центральную линию окружающего текста по воображаемой центральной линии фрейма.

Вариант 3. Окно браузера поделить на три фрейма. В верхнем фрейме расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в среднем окне по щелчку мыши на миниатюре изображения в верхнем фрейме. В верхнем фрейме дополнительно организовать навигационную модель для среднего фрейма. Создать новый HTML-документ и вставить его в ранее созданный, как плавающий фрейм, выравнивая нижнюю границу фрейма по верхней границе текста.

Вариант 4. Окно браузера поделить на два фрейма. В нижнем расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в верхнем окне по щелчку мыши на миниатюре изображения в нижнем фрейме. Создать вертикальный, относительно двух ранее созданных, фрейм. В нем организовать навигационную модель для первого фрейма. Создать новый HTML-документ и вставить его в ранее созданный, как плавающий фрейм, выравнивая нижнюю границу фрейма по верхней границе текста.

Вариант 5. Окно браузера поделить на два фрейма. В нижнем расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в верхнем окне по щелчку мыши на миниатюре изображения в нижнем фрейме. Создать вертикальный, относительно верхнего, из ранее созданных, фрейм. В нем организовать навигационную модель для второго фрейма. Создать 2 новых HTML-документа и вставить их в ранее созданный, как плавающие, выравнивая верхнюю границу фрейма по надстрочной линии символов окружающего текста.

Вариант 6. Окно браузера поделить на четыре горизонтальных фрейма. В правом расположить небольшое изображение. Организовать возможность вывода полномасштабного изображения в самом левом окне по щелчку мыши на миниатюре изображения правого фрейма. Создать вертикальный, относительно ранее созданных, фрейм. В нем организовать навигационную модель для второго фрейма. Создать новый HTML-документ и вставить его в ранее созданный документ, как плавающий фрейм, выравнивая верхнюю границу фрейма по верхней границе окружающего текста.

Лабораторная работа №6 (2 часа)

Простые визуальные эффекты

Смена изображений

Для смены одного изображения на другое достаточно с помощью сценария заменить значение атрибута SRC тега . Например:

```
<HTML>
<IMG ID = "myimg" SRC= 'pict1.gif '
onclick = "document.all.myimg.src = 'pict2.gif' ">
</HTML>
```

Здесь смена изображения из файла pict1.gif на изображение из файла pict2 происходит при первом щелчке на нем. Последующие щелчки не приведут к видимым изменениям, поскольку второе изображение будет заменяться им же. Чтобы при повторном щелчке происходила замена изображения на предыдущее необходимо создать переменную-триггер (флаг), принимающий одно из двух возможных значений, по которому можно определить, какое из двух значений надо отобразить.

```
<HTML>
<IMG ID = "myimg" SRC= 'pict1.gif '
onclick = " imgchange( )">
<SCRIPT>
var flag=false
function imgchange( ) {
if (flag) document.all.myimg.src = "pict1.gif"
else document.all.myimg.src = "pict2.gif"
flag=!flag
}
</SCRIPT>
```

</HTML>

Цветовые эффекты

Задача изменения цвета кнопки при наведении на нее указателя мыши и возвращения в первоначальное состояние при удалении указателя с кнопки может быть решена следующим образом:

```
<HTML>
<STYLE>
mystile {font-weight:bold; background-color: a0a0a0} // серый цвет кнопок
</STYLE>
```

```
<FORM onmouseover = "colorchange ( 'yellow' )" onmouseout
= "colorchange ( 'a0a0a0' )">
<INPUT TYPE = "BUTTON" VALUE = "Кнопка" CLASS = "mystile"
onclick = "alert( 'Вы нажали кнопку' )" >
</FORM>
<SCRIPT>
function colorchange (color){
if (event. srcElement.type == "button")
    event. srcElement. style. backgroundColor = color;
}
</SCRIPT>
</HTML>
```

Функция colorchange() проверяет, является ли инициатор события объектом типа button. Если это так, то цвет кнопки меняется. Без этой проверки менялся бы не только цвет кнопок, но и текста.

Аналогичным способом можно изменять цвет фрагментов текста. Но в этом случае текст должен быть заключен в контейнер, например в теги <P>, , <I>, <DIV>.

Можно создать прямоугольную рамку, окаймляющую текст, которая периодически изменяет цвет. Рамка создается тегами одноячеечной таблицы с заданием нужных атрибутов и параметров стиля:

```
<TABLE ID= "tab" BORDER=1 WIDH=200 style= "border:10 solid : red">
<TR><TD> Доброе утро! </TR></TD>
</TABLE>
```

Функция изменения цвета:

```
<SCRIPT>
function flash( ) {
if ( !document.all) return null;
if (tab.style.borderColor == 'red')    tab.style.borderColor = 'yellow'
else tab.style.borderColor = 'red';
}
setInterval ("flash", 500);           //мигание рамки с интервалом 500 мс
</SCRIPT>
```

Объемные заголовки

Объемные заголовки часто используются на веб-страницах. Идея создания объемного заголовка состоит в наложении нескольких надписей с одинаковым содержанием с некоторым сдвигом по координатам. Наилучший эффект достигается путем подбора цветов надписей (игрой света и тени) с учетом цвета фона. Для этого используют библиотеку стилей. Функция, создающая заголовок с заданными параметрами:

```
function d3 (text, x, y, tcolor, fsize, fweight, family, zind) {
/*   text – текст заголовка
   x – горизонтальная координата (left)
   y – вертикальная координата (top)
   tcolor – цвет переднего плана
   fsize – размер шрифта (пт)
   fweight – вес (толщина шрифта)
   family – название семейства шрифтов
   zind z-Index      */
if (!text) return null // если текст не указан ничего не выполняется
//значение параметров по умолчанию
if (!x) x=0
if (!y) y=0
if (!tcolor) tcolor='00aaff'
if (!fsize) fsize=36
if (!fweight) fweight =800
if (!family) ffamily='arial'
// внутренние настройки
var sd=5, hd=2
var xzind= “ ”
if (zind) xzind= “; - Index:”+zind
var xstyle =’font-family:’ + family + ‘;font-size:’ + fsize + ‘;font-weight:’ + fweight
+ ‘;’
var xstr = ‘<DIV STYLE = “position: absolute; top:’ + (y +sd ) + ‘; left :’ +
( x + sd ) + xzind + ‘ “>’
xstr+=‘<P styl e = “ ’ + xstyle + ‘color: darked”>’ + text + ‘</P></DIV>’
xstr+= ‘<DIV STYLE = “ position: absolute; top:’ + y + ‘; left :’ +
x + xzind + ‘ “>’
xstr+=‘<P styl e = “ ’ + xstyle + ‘color: silver”>’ + text + ‘</P></DIV>’
xstr+= ‘<DIV STYLE = “ position: absolute; top:’ + ( y + hd ) + ‘; left :’ +
(x +hd) + xzind + ‘ “>’
xstr+=‘<P styl e = “ ’ + xstyle + ‘color:’ + tcolor + “”>’ + text + ‘</P></DIV>’
document.write(xstr)      //запись в документ
}
```

Параметр z-Index позволяет установить слой, в котором находится заголовок, и тем самым указать, будет ли заголовок располагаться над или под другим видимым элементом документа. Элементы с более высоким значением z-Index находятся над элементами, у которых z-Index меньше. Перекрытие эле-

ментов с одинаковыми значениями z-Index определяется порядком их следования в HTML-документе.

Вызов приведенной выше функции может выглядеть так:

d3 (“это не графика, это просто стиль текста”, 50, 50, ‘blue’, 72, 800, ‘times’)

Задание

Выполнить следующие действия на веб-странице:

1. Создать программу для работы с галереей миниатюр. При щелчке кнопкой мыши по миниатюре изображение должно увеличиваться, а затем при щелчке на увеличенном изображении оно должно уменьшаться. Доработайте приведенную в тексте функцию функцию `imgchange()`. Для решения этой задачи потребуется массив флагов и функция обработчик, определяющая на каком именно изображении произошел щелчок:

```
var p1=new Array (“pict1.gif” , ... ) //массив имен исходных файлов
var p2=new Array (“pict2.gif” , ... ) //массив имен замещающих файлов
//формирование тегов, описывающих изображения
var xstr = “ “
for (i=0; i<p1.length; i++)
xstr+= ‘<IMG ID = “i’ + i +’ ” SRC = “ ’+ p1[i]+’ ” onclick = “imgchange( )” >’
}
document.write(xstr) // запись в документ
```

2. Выполнить замену фрагмента текста с черного на красный при наведении на него указателя мыши.

3. Выделите фрагмент текста мигающей трехцветной рамкой.

4. Создать эффект динамического изменения цвета ссылок. Различать цвета мерцания использованных и неиспользованных ссылок. (Множество цветов задать массивом. Использовать свойства `linkColor` и `linkColor` объекта `document`). Для изменения цвета случайным образом можно использовать метод `random()` (счетчик случайных чисел) встроенного объекта `Math`. Если требуется получить случайное число x , лежащее в интервале от A до B , то $x=A+(B - A)*\text{Math.random}()$

5. Создать три объемных заголовка, поэкспериментировав со значениями внутренних параметров `sd`, `hd`, а также с заданием параметров по умолчанию.

Лабораторная работа №7 (2 часа)

Применение фильтров

С помощью фильтров каскадных таблиц стилей можно получить разнообразные эффекты: постепенное появление или исчезновение рисунка, плавное преобразование одного изображения в другое, задание степени прозрачности и др.

Фильтр следует понимать как некий инструмент преобразования изображения, взятого из графического файла и вставленного в HTML – документ с

помощью тега . Следует иметь в виду, что фильтры работают только в IE4+.

Фильтры можно применять не только к графическим объектам, но и к текстам, текстовым областям, кнопкам.

Прозрачность

С помощью фильтра alpha можно установить прозрачность графического объекта. Сквозь прозрачные графические объекты видны нижележащие изображения. Прозрачность имеет несколько вариантов градиентной формы. Например, она может увеличиваться от центра к краям изображения

Фильтр alpha задается с помощью каскадной таблицы стилей и имеет ряд параметров. В примере для графического изображения стиль определяется с помощью атрибута STYLE:

```
<IMG ID = "myimg" SRC = " pict. gif "
STYLE = "position: absolute; top:10; left: 50;
filter: alpha (opacity = 70, style = 3)">
```

Здесь целочисленный параметр opacity определяет степень непрозрачности. Значение 0 соответствует полной прозрачности изображения, а 100 – полной непрозрачности. Параметр style задает градиентную форму распределения прозрачности по изображению как целое число от 0 до 3. По умолчанию значение параметра равно 0, и градиент не применяется. Фильтр имеет и другие параметры, определяющие прямоугольную область изображения, к которому применяется фильтр. По умолчанию фильтр применяется ко всему изображению.

Фильтр можно определить в каскадной таблице стилей внутри контейнерного тега <STYLE>:

```
<HTML>
<STYLE>
#myimg {position: absolute; top:10; left: 50; filter: alpha (opacity = 70, style = 3)}
</STYLE>
< IMG ID = "myimg" SRC = " pict. gif "
</HTML>
```

Доступ к свойствам фильтра в сценарии:

```
document.all.id_изображения.filters ["имя_фильтра"]. параметр = значение
```

Для рассмотренного примера это выражение имеет вид:

```
document.all.myimg.filters ["alpha"]. opacity = 30
```

Для остальных параметров alpha аналогично.

Для IE5.5+ можно использовать другой синтаксис, в котором в каскадной таблице стилей задается ссылка на специальный компонент и имя фильтра:

```
#myimg { filter: progid: DXImageTransform . Microsoft . alpha
(opacity = 70, style = 3)}
```

Тогда доступ к свойствам фильтра:

```
document.all.myimg.filters ["DXImageTransform. Microsoft alpha"]. opacity = 30
```

Трансформация

Фильтр alpha статический. Существуют и динамические фильтры: apply() – фиксирует изображение, play() – трансформирует, revealtrans() – преобразовывает изображение, stop() останавливает процесс преобразования при необходимости.

Фильтр revealtrans() имеет параметры: duration – длительность преобразования в секундах (число с плавающей точкой) и transition – тип преобразования (целое от 0 до 23).

Для эффекта появления изображения можно воспользоваться фрагментом, который происходит после загрузки документа, т.е. по событию onload:

```
<HTML>
<BODY onload = "transform()" >
<IMG ID = "myimg" SRC = " pict. gif" STYLE = "position: absolute; top:10;
left: 50; visibility = "hidden" filter: revealtrans (duration = 3, transition =12)" >
//transition =12 соответствует плавной трансформации
</BODY>
<SCRIPT>
function transform ( ) {                               //появление изображения
document.all.myimg.style. visibility = "hidden"       // изображение невидимо
myimg. filters ( "revealtrans"). apply( )
myimg.style. visibility = "visible"
myimg. Filters ( "revealtrans"). play( )              // выполняем преобразования
}
</SCRIPT>
</HTML>
```

Для замены одного изображения на другое необходимо установить начальное и конечное изображение путем присвоения нужных значений свойству src объекта, соответствующего изображению, например фрагментом:

```
document.all.myimg.src = "pict2. gif"
```

Рассмотренный синтаксис воспринимается браузерами IE4+. Для IE5.5+ в каскадной таблице стилей задается ссылка на специальный компонент и имя фильтра. Так для трансформации изображения по щелчку мыши на графическом объекте в другое, и обратно, можно воспользоваться программой:

```
<HTML>
<STYLE>
#myimg{position: absolute; top:10; left: 50; filter: progid: DXImageTransform . Microsoft revealtrans (duration = 3, transition = 12)}
</STYLE>
< IMG ID = "myimg" onclick = "transform( )" SCR = " ear. gif">
<SCRIPT>
function transform( ) {
//фиксация исходного изображения
myimg. filters ("DXImageTransform . Microsoft revealtrans"). apply ( )
//определение конечного изображения
if (document. all. myimg. src. indexOf ("ear")!= -1)
```

```

document.all.myimg.src = "s.gif"
    else document.all.myimg.src = "ear.gif"
//выполняем преобразование
myimg.filters ("DXImageTransform . Microsoft revealtrans"). play ( )
}
</SCRIPT>
</HTML>

```

В браузере IE5.5+ возможно применение фильтра basicimage, с помощью которого изображение можно повернуть на угол, кратный 90 градусам, задать прозрачность, зеркально отразить и др.

Задание

В HTML-документе из предыдущей лабораторной работы применить к рисункам методы трансформации и изменения прозрачности изображения и фона.

Лабораторная работа №8 (2 часа)

Динамическое изменение элементов документа

Элементы HTML-документа задаются тегами, большинство из которых имеют параметры (атрибуты). В объектной модели документа тегам соответствуют объекты, а атрибутам – свойства этих объектов. Названия свойств объектов, как правило, совпадают с названиями атрибутов, но записываются в нижнем регистре.

Наиболее удобный способ динамического изменения HTML-документа основан на использовании свойств innerText, outerText, innerHTML и outerHTML. С их помощью можно получить доступ к содержимому элемента. Изменяя значения перечисленных свойств можно частично или полностью изменить сам элемент. Например, можно изменить только надпись на кнопке, а можно превратить кнопку в изображение или Flash-анимацию.

Значением свойства innerText является все текстовое содержимое между открывающим и закрывающим тегами элемента. Внутренние теги игнорируются. Данные открывающего и закрывающего тегов соответствующего элемента также не входят.

В отличие от предыдущего свойство outerText включает в себя данные открывающего и закрывающего тегов. Таким образом, outerText есть весь текст, содержащийся в контейнере, включая его внешние теги. Например, задан HTML-код:

```

<DIV ID = "my" >
<A HREF = 'raznoe.htm'>
<IMG SRC = 'picture.jpg'> Ссылка на раздел <B> Разное </B>
</A>
</DIV>

```

Здесь свойства `innerText` и `outerText` для элемента, заданного контейнерным тегом `<DIV>`, совпадают:

```
document.all.my.innerText //значение равно – «Ссылка на раздел Разное»
```

При присвоении свойствам `innerText` и `outerText` новых значений нужно помнить, что если значения содержат теги, то они не интерпретируются, а воспринимаются как обычный текст.

Свойство `innerHTML` содержит внутренний HTML-код контейнера элемента. Присвоение этому свойству нового значения, содержащего HTML-код, приводит к интерпретации кода. Свойство `outerText` дополнительно включает внешние открывающие и закрывающие теги элемента.

Для приведенного HTML-кода значение `document.all.my.innerHTML` равно ` Ссылка на раздел Разное "/`

Значение `document.all.my.outerHTML` – `<DIV ID = "my" > Ссылка на раздел Разное </DIV>`.

Если в сценарии выполнить выражение `document.all.my.innerHTML = "<BUTTON>Щелкни здесь</BUTTON>"` ссылка, изображение и текст будут заменены кнопкой с надписью Щелкни здесь. При этом контейнерный тег `<DIV ID = "my" >` сохранится. Если аналогичным образом использовать `outerHTML`, кнопка также появится, но уже без контейнера `<DIV ID = "my" >`.

Свойства `innerHTML` и `outerHTML` могут применяться к элементам, заданным неконтейнерными тегами. Тогда `innerHTML` и `outerHTML` совпадают.

Для ускорения загрузки графики можно использовать следующие возможности JavaScript. Можно организовать предварительную загрузку изображений в кэш-память браузера, не отображая их на экране. Это особенно эффективно при начальной загрузке страницы. Пока изображения загружаются в память, оставаясь невидимыми, пользователь может рассматривать текстовую информацию.

Для предварительной загрузки изображения требуется создать его объект в памяти браузера. Это можно сделать следующим выражением:

```
myimg = new Image (ширина, высота)
```

Параметры должны соответствовать значениям атрибутов `WIDTH` и `HEIGHT` тега ``, который используется для отображения предварительно загруженного изображения.

Для созданного в памяти объекта изображения можно создать имя или URL-адрес графического файла:

```
myimg.src = "URL-адрес изображения"
```

что предписывает браузеру загрузить изображения без его отображения.

После загрузки в кэш-память всех изображений и загрузки всего документа можно сделать их видимыми. Для этого свойству `src` элемента `` нужно присвоить значение этого же свойства объекта изображения в кэш-памяти. Например,

```
document.images[0].src = myimg.src
```


Здесь слева указано свойство src первого в документе элемента, соответствующего тега , справа – свойство src объекта изображения в кэш-памяти.

С помощью JavaScript можно через заданный интервал времени запускать код или функцию. При этом создается эффект одновременного (параллельного) выполнения вычислительных процессов.

Для организации повторения через заданный интервал выполнения некоторого выражения служит метод setInterval() объекта window:

```
setInterval( выражение, период, [, язык])
```

Первым параметром является строка, например вызов функции. Период указывается в миллисекундах. Третий параметр – необязательный, в котором указывается язык с помощью которого написано заданное выражение. По умолчанию – JavaScript.

Метод setInterval() возвращает некоторое целое число – идентификатор временного интервала, который может быть использован в дальнейшем, например для прекращения выполнения процесса методом clearInterval(). Например,

```
var pr = setInterval( "myfunc(), 100" )  
if (confirm ( "Прервать процесс?" ) )  
clearInterval(pr)
```

Если требуется выполнить действие с некоторой временной задержкой, используется метод setTimeout(), имеющий синтаксис аналогичный setInterval(). Для отмены задержки процесса, запущенного setTimeout(), используют clearTimeout().

Задание

1. Создать HTML-документ, в котором отображается список названий графических объектов и одно исходное изображение. Щелчок на элементе списка должен приводить к отображению соответствующего элемента.
2. Создать в HTML-документе две кнопки. Щелчок по кнопке ПУСК открывает через 5 секунд новое окно и загружает в него некоторый документ. Это действие может быть отменено с помощью кнопки ОТМЕНА.

V. КОНСПЕКТ ЛЕКЦИЙ

Лекция 1

Глава 1

Мировой рынок информационных услуг

1.1. Основные понятия

Рынок информационных услуг - совокупность экономических, правовых и информационных отношений по торговле (продаже и покупке услуг) между поставщиками и потребителями. Он характеризуется определенной номенклатурой услуг, условиями и механизмами их предоставления и ценами.

Товаром на рынке информационных услуг является *информация*. Информация, зафиксированная на материальных носителях и хранящаяся в *информационных системах* (библиотеках, архивах, фондах, банках данных, других информационных системах), образует *информационные ресурсы*.

Под информационной системой понимается организационно упорядоченная совокупность документов и информационных технологий

Мировые информационные ресурсы обычно делят на три сектора:

- ◆ сектор деловой информации;
- ◆ сектор научно-технической и специальной информации;
- ◆ сектор массовой потребительской информации.

Виды деятельности, связанные с формированием информационных ресурсов, поддержанием их в актуальном состоянии, созданием средств связи, обработки и копирования информации, объединяют в понятие *информационной индустрии*. Под средствами обработки наряду с вычислительной техникой понимается и программное обеспечение.

Способность решать задачи информационного обслуживания на уровне максимальных возможностей, определяемых достигнутым на данный момент состоянием развития вычислительной техники и связи, называют *информационным потенциалом*.

Структуры, которые работают на информационном рынке, предлагают потребителям следующие виды услуг:

- ◆ непосредственный доступ к базам данных — режим on-line;
- ◆ пакетный доступ к базам данных — режим off-line;
- ◆ базы данных на дискетах и компакт-дисках;
- ◆ консультации, оказываемые специалистами в области информационных ресурсов;
- ◆ обучение доступу к мировым информационным ресурсам.

Поставщики информации на рынке информационных услуг - коммерческие структуры, государственные и общественные организации, частные лица. Обычно они именуется информационными корпорациями, информационными агентствами, информационными службами, информационными центрами.

Классификация *пользователей* может проводиться по различным основаниям. Например,

- ◆ специалисты промышленных предприятий и предприятий торговли;

- ◆ специалисты консалтинговых и маркетинговых информационных агентств;
- ◆ работники научно исследовательских учреждений и учебных заведений;
- ◆ работники государственных учреждений;
- ◆ работники общественных организаций;
- ◆ индивидуальные пользователи.

1.2. Этапы развития мирового рынка информационных услуг

Рынок информационных услуг имеет многолетнюю историю. Качественные изменения он претерпел в середине 1960-х гг. с появлением вычислительной техники и ее использованием в области обработки и передачи информации. В этот период основными источниками информации являлись государственные информационные службы, учебные заведения, различные общественные организации, занимающиеся сбором информации в определенной области, библиотеки. Пользователям предоставлялась библиография, реферативная и аналитическая информация.

Первые автоматизированные информационные системы получили название «Информационно-поисковые системы» (ИПС). При внедрении ИПС была автоматизирована лишь часть функций информационного обслуживания.

Следующим шагом, который стал возможным в связи с развитием вычислительной техники, ростом объемов памяти и быстродействия, явилось возникновение фактографических информационных систем. Эти системы содержали уже формализованную информацию в виде значений свойств различных объектов (лиц, организаций, событий и т. д.) в виде баз данных. С расширением рынка электронной информации наблюдается уменьшение доли государственных служб на информационном рынке, что отражено на рис. 1.

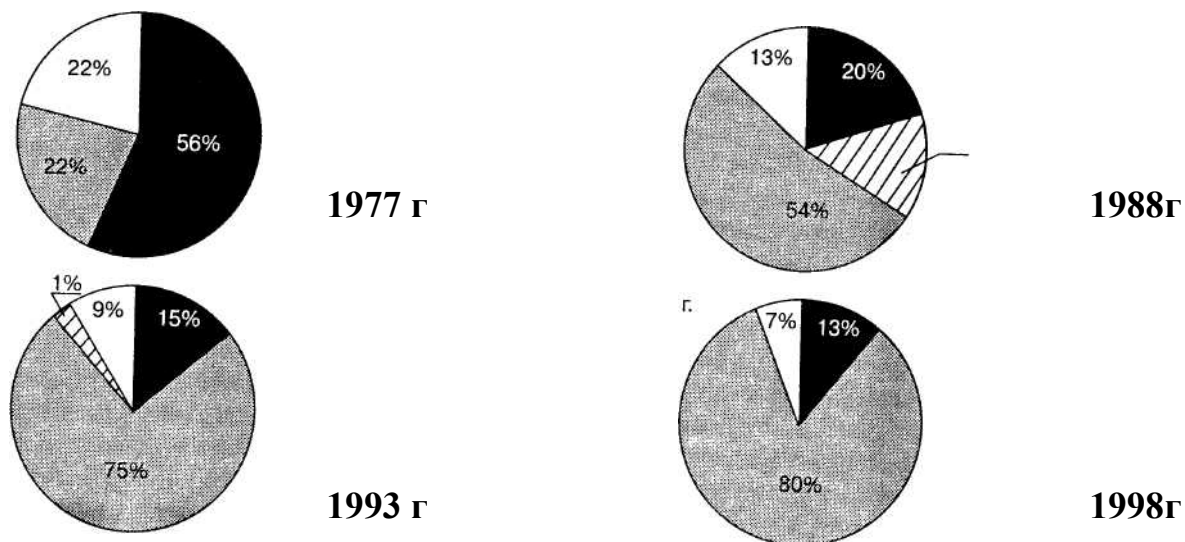


Рис. 1. Доли государственных и негосударственных служб на информационном рынке

Становление рынка электронной информации сопровождалось также специализацией (разделением труда) организаций, занимающихся информационным обслуживанием. Сформировалось три группы информационных служб:

- ◆ **центры-генераторы** (производители информации) — специализируются на добыче информации, формировании баз данных и поддержании их в актуальном состоянии;
- ◆ **центры распределения** (поставщики информации, так называемые вендоры) — занимаются информационным обслуживанием пользователей на основе баз данных, поставляемых им на коммерческой основе центрами-генераторами;
- ◆ **информационные агентства** — помимо функции сбора информации, формирования и ведения баз данных осуществляют и функции обслуживания пользователей.

К концу 2000 г. в мире функции центров-генераторов выполняли 3520 компаний, а функции центров-поставщиков — 4220 компаний.

Лекция 2

1.3. Профессиональные базы данных

Одной из форм предоставления информационных услуг потребителю является обеспечение его доступа к **профессиональным базам данных** научно-технической, деловой, правовой и другой информации. На рис. 2 показаны тенденции роста количества баз данных и их объема за последнее двадцатилетие.

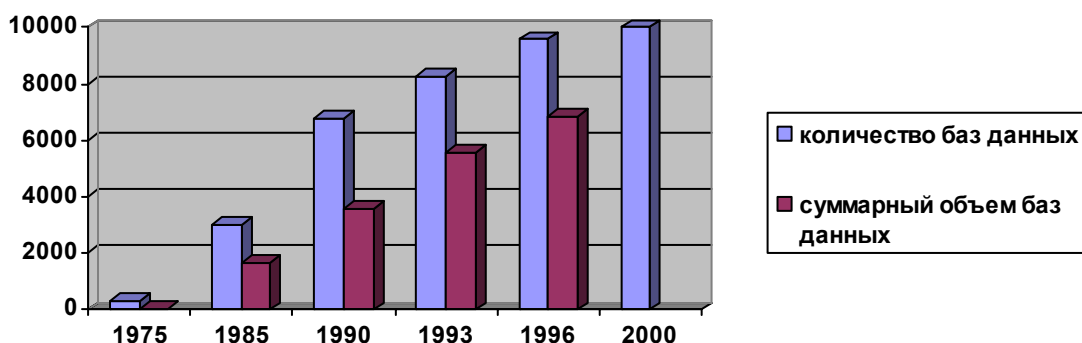


Рис. 2. Динамика изменения количества информационных агентств и баз данных на мировом информационном рынке

Содержание баз данных отражает потребности пользователей. Если первоначально потребители запрашивали главным образом научно-техническую информацию, то в настоящее время их основной интерес направлен на информацию, касающуюся бизнеса, торговли и промышленности. Большой интерес также проявляется к патентной и правовой информации. В табл. 1 показано распределение баз данных по содержащейся в них информации. Классификация баз данных по тематике, приведенная в таблице, представляется условной.

Таблица 1 Распределение баз данных по содержанию, %

Тематика баз данных	1988 г.	1991 г.	1993 г.
Бизнес	34	33	33
Потребительская информация	3	7	10

Медицина и биология	8	11	9
Гуманитарные дисциплины	2	4	4
Правовая информация	8	9	12
Политематическая информация	1	6	3
Новости	8	4	4
Общественные науки	9	7	6
Научно-техническая информация	22	19	19

По своей форме информационные ресурсы, содержащиеся в базах данных, могут быть разделены на числовые (фактографическая информация) и текстовые. Объемы текстовых баз данных значительно превышают объемы числовых баз, и этот разрыв все более увеличивается с течением времени (рис. 3).

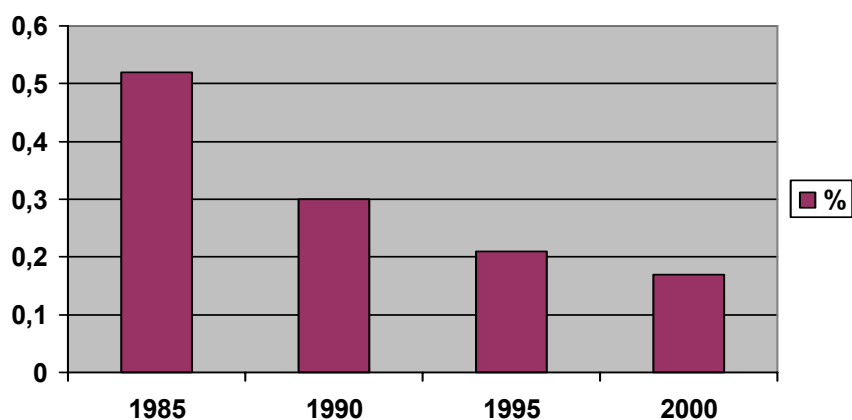


Рис. 3. Динамика отношения объема числовых баз к общему объему баз данных на мировом информационном рынке

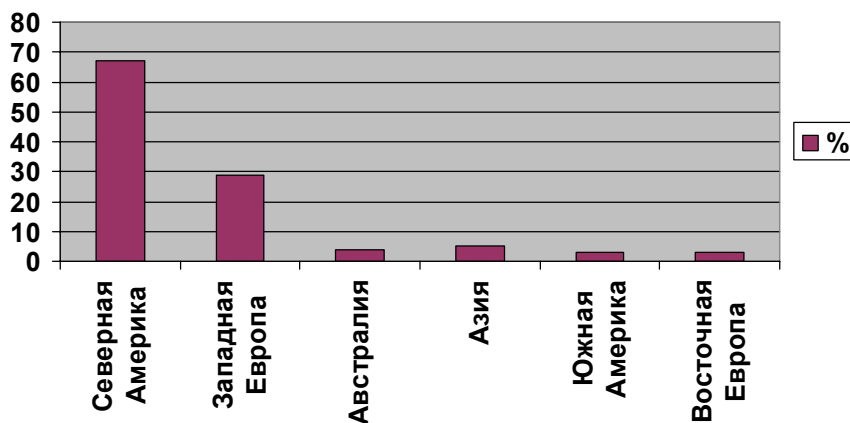


Рис. 4. Доля региона в общем объеме мировых баз данных (2000 г.), %

Распределение информационных мировых ресурсов по регионам мира показано на рис. 4.

Ориентироваться в гигантском объеме мировых информационных ресурсов позволяет популярное во всем мире справочное издание Gale Directory of Databases, выпускаемое дважды в год в виде двух томов фирмой Gale Research, Inc.

Первый том содержит список всех онлайн-баз данных, а также ос-

новые показатели развития информационной индустрии

Во втором томе приводится информация о базах данных, распространяемых на CD-ROM.

Отдельно выделены подробные сведения о производителях (генераторах) и поставщиках баз данных и списки предлагаемых ими баз.

Лекция 4

1.4. Ведущие мировые информационные корпорации

Информационная корпорация «Дан энд Брэдстрит» (Dun & Bradstreet)

Эта крупнейшая в мире американская многопрофильная информационная корпорация, возникшая более 150 лет назад. В корпорации занято свыше 60 тыс. сотрудников, работающих по всему миру.

Информационные продукты и услуги:

- ◆ информация, освещающая финансовое состояние и кредитоспособность около 70 млн компаний во всех странах мира;
- ◆ маркетинговые исследования;
- ◆ справочники «страна/фирма/продукт», включая Exporter's Encyclopedia;
- ◆ регулярные обзоры таможенного, налогового и инвестиционного климата в 180 странах мира;
- ◆ справочники-рейтинги долговых обязательств Mood's — «настольная книга» инвесторов по всему миру;
- ◆ международная программа подготовки специалистов в области финансов, маркетинга и информационного обеспечения.

Основными видами продукции корпорации являются аналитические справки и деловая справочная литература.

Корпорация выпускает ежемесячный бюллетень «*Международный обзор коммерческих рисков и платежей*».

Информационные обзоры по отдельным странам содержат общую и детальную информацию по отдельным странам, а также сведения о факторах риска в этих странах

Сопоставительные обзоры по отраслям, выпускаемые корпорацией, включают:

- ◆ отчет о нормативных показателях по отраслям (средние, верхние и нижние);
- ◆ выборочные нормативные показатели — отчет о показателях по выбранной заказчиком отрасли;
- ◆ основные относительные показатели (ежегодно изучаются 370 различных показателей для промышленности).

Информационная корпорация «Диалог» и компания Data-Star

Основанная в 1972 является одной из крупнейших универсальных ин-

формационных систем мира с числом клиентов около 20 тыс. и объемом продаж, превышающим 280 млн долл. Система «Диалог» включает в себя свыше 250 баз данных объемом 12 терабайт.

В группе «Бизнес» содержится информация о компаниях, организациях, патентах, торговых марках, данные финансового и промышленного анализа, деловые новости.

В состав корпорации входит в настоящее время компания Data-Star — европейский лидер среди служб on-line доступа к базам данных. Фирма основана в 1981 г. в Швейцарии.

Наиболее важные разделы: деловые новости, финансовая информация, маркетинговые исследования, статистика торговли, экономический анализ; забота о здоровье, фармацевтика; химия; экологическая индустрия; биомедицина; биотехнология и др.

Информационная корпорация Questel-Orbit

Включает две подсистемы – Questel (Франция) и Orbit (США). Подсистемы определенным образом специализированы как по содержанию баз данных, так и по поисковым возможностям и языкам общения. Корпорация является одним из лидеров на рынке информации, характеризующей интеллектуальную собственность (патенты, товарные знаки), существует более 20 лет и обеспечивает 35 тыс. потребителей по всему миру.

В табл. 2 приведено количество баз данных по предметным группам и подсистемам.

Таблица 2 Распределение баз данных в системе Questel-Orbit

№ п/п	Предметные группы	Количество баз данных	
		Orbit	Questel
1	Бизнес	16	25
2	Химия	23	4
3	Энергетика и науки о Земле	12	4
4	Инжиниринг	14	9
5	Здоровье, здравоохранение и окружающая среда	9	4
6	Гуманитарные и социальные науки	1	3
7	Материаловедение	14	5
8	Медицина	3	10
9	Новости	3	16
10	Патенты	21	15
11	Наука и технология	30	20
12	Торговые марки	—	14

В базах данных по компаниям содержатся сведения об европейских компаниях и 36 тыс. компаний США, занимающихся разработками новейших технологий или производством высокотехнологичных продуктов в области ком-

пьютерного производства и программного обеспечения, взаимодействия с окружающей средой, робототехники, биотехнологии и др.

В группе баз данных «Патенты» размещена международная информация по патентам, патентная информация по отдельным странам, патентная информация по отраслям и юридическая информация по патентам.

Информационное агентство LEXIS-NEXIS

On-line система LEXIS создана компанией Mead Data Central для американских юристов. NEXIS – постоянно обновляющаяся электронная библиотека газет, журналов и других информативных изданий функционирует с 1979.

LEXIS-NEXIS — крупнейшая в мире полнотекстовая онлайн-информационная система. В ней предоставляется юридическая, политическая, коммерческая и другая информация, содержится более 33 тыс. информационных источников, среди источников – 80 крупнейших информационных агентств. LEXIS-NEXIS обслуживает 2 млн клиентов, в базах системы содержится 3 млрд. документов объемом 27 терабайт. Ежедневный поток поступающей информации составляет 2 млн. документов.

Лекция 5

1.5. Информационные ресурсы сети Интернет

Обычно под словом «Интернет» понимается глобальная компьютерная сеть. С точки зрения пользователя Интернет можно рассматривать как мощное глобальное средство обмена информацией. Одним из распространенных и перспективных сервисов Интернет является сервис прямого доступа Word Wide Web — WWW, представляющий собой систему документов, включающих текстовую и графическую информацию, размещенных на узлах Интернет и связанных между собой гиперссылками.

Классификация источников информации в Интернете может проводиться по разным основаниям.

По способам представления информации выделяют:

- ◆ web-страницы — наиболее распространенный и используемый из информационных ресурсов. Этот ресурс представляет собой страницы гипертекста, содержащие наряду с текстовой - графическую, звуковую, видеоинформацию;
- ◆ файловые серверы, представляющие собой реализацию в Интернете традиционного способа представления информации;
- ◆ телеконференции. Они разбиваются на группы (рубрики) по тематике. Участвующие в телеконференциях могут написать свое сообщение или послать комментарии на чужое сообщение;
- ◆ базы данных, содержащие, кроме текстовой, и другие виды информации.

Информационные ресурсы также могут быть разделены *по языковому признаку* (бесспорным лидером являются англоязычные сайты).

В сети имеет место классификация и *по территориальному признаку*. Ряд

сайтов предоставляет свою информацию для потребителей определенного региона, хотя доступ к сайту возможен и из любой точки сети.

Наиболее важным аспектом классификации информационных ресурсов сети Интернет является *содержание информации*. Деловая информация, необходимая в предпринимательской деятельности, по этому критерию может быть разделена на следующие группы:

1. сведения о фирмах, организациях,
2. сведения о состоянии мировой экономики и экономики отдельных стран,
3. сведения о состоянии отраслевых рынков. Анализ отраслевых рынков,
4. деловые новости,
5. справочная информация.

1.6. Виды информации, хранимой в Интернете и профессиональных базах

В соответствии с Законом от 20 февраля 1995 г. № 224-ФЗ «Об информации, информатизации и защите информации» под термином «информация» понимаются сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления.

Информация о предметной области, то есть об объектах и их связях, может быть представлена в двух видах: в формализованном виде и в виде текста на естественном языке.

Формализованное описание конкретного объекта включает *имя свойства* (характеристики) и *значение этого свойства* для данного объекта. Имя свойства отражает грамматическую роль, которую играет значение этого свойства по отношению к данному классу объектов. Значение свойства (характеристики) может задаваться нормированным словарем или произвольно. Например, значение свойства «организационно-правовая форма фирмы, организации» должно выбираться из нормированного словаря, а значение свойства «название организации» является произвольным. Формализованное описание объекта во многом схоже с анкетной формой описания объекта. Некоторые авторы называют формализованное описание информации числовым. Это не совсем точное определение, так как фамилия лица или название организации задается не числовым значением, но может быть представлено в формализованном виде.

Общими недостатками естественного языка с позиций оценки эффективности поиска являются *избыточность* и *недостаточность*.

Избыточность проявляется в следующем:

- 1) активную роль в процессе передачи содержания текста играют только некоторые слова или словосочетания, так называемые ключевые слова. Другие можно отбросить практически без ущерба для понимания смысла;
- 2) в естественном языке используются синонимы, то есть слова, различающиеся по звучанию и написанию, но тождественные или близкие по смыслу;
- 3) имеет место неоднозначность выражения, то есть возможность выразить одну и ту же мысль словами, не являющимися синонимами (например, «*уменьше-*

ние сопротивления» — «увеличение проводимости»).

Недостаточность выражения на естественном языке проявляется в следующем:

1) многозначность отдельных слов — омонимия (совпадение по звучанию и написанию разных по значению слов). Например, «ключ» (для двери), «ключ» (источник);

2) эллипсность естественного языка. Под эллипсностью понимается пропуск в тексте подразумеваемых слов.

Для облегчения поиска в больших массивах текстовой информации, стали создавать информационно-поисковые системы. С помощью элементов этих же языков описывались и запросы. Для отбора документов в ответ на запрос осуществлялось сравнение поисковых образов запросов и поисковых образов документов, которое проводилось на одном и том же искусственном языке. Такой подход является вынужденным.

Лекция 6

1.7. Вопросы эффективности поиска информации в Интернете и профессиональных базах

Одними из наиболее важных показателей эффективности информационных систем, содержащих текстовую информацию, являются семантические показатели. Семантические показатели основаны на оценке релевантности между документами и запросами.

Релевантность — объективно существующее смысловое соответствие между содержанием документа и запроса. Объективность оценок релевантности обеспечивается тем, что они устанавливаются экспертным путем, а не автором запроса.

Семантическими показателями являются *полнота выдачи* (потери информации) и *точность выдачи* (информационный шум).

Введем следующие обозначения:

a - множество релевантных и выданных системой документов;

б - множество нерелевантных, но выданных системой документов;

в - множество релевантных, но не выданных системой документов.

$$\text{Полнота выдачи (ПВ)} = \frac{a}{a + в} \times 100\%$$

$$\text{Точность выдачи (ТВ)} = \frac{a}{a + б} \times 100\%$$

$$\text{Потери информации (ПИ)} = \frac{в}{a + в} \times 100\%$$

$$\text{Информационный шум (ИШ)} = \frac{б}{a + б} \times 100\%$$

Другой группой показателей оценки эффективности документальных ин-

формационных систем являются прагматические показатели. Эти показатели можно определить только в процессе эксплуатации информационной системы.

Прагматические показатели определяют абоненты системы на базе оценок пертинентности выданных документов.

Пертинентность — это субъективно оцениваемое соответствие содержания документов или текстов информационным интересам потребителя. Пертинентность может оценить только автор запроса, работающий с информационной системой. Оценки пертинентности, как правило, отличаются от результатов, полученных на основе оценок релевантности.

Анализ содержимого профессиональных баз за последние 15 лет показывает неуклонный рост доли текстовой информации в общем объеме информации в профессиональных базах. Если в 1985 г. доля текстовой информации составляла 47 %, то в 2000 г. эта доля составляла уже 84 %. Представляется, что основная информация в Интернете также является текстовой. Эти обстоятельства позволяют сделать вывод о том, что подходы к оценке эффективности поиска в документальных системах в полной мере распространяются и на профессиональные базы, и на информационные ресурсы Интернета.

Информационные ресурсы Интернета и имеющиеся в среде Интернет поисковые средства обладают определенной спецификой, которая оказывает существенное влияние на эффективность поиска в этой среде.

Основными поисковыми средствами в Интернете являются поисковые системы и каталоги. *Поисковые системы* состоят из трех частей:

- ◆ робот — программа, которая посещает web-серверы, считывает и индексирует полностью или частично их содержимое и далее следует по ссылкам, найденным на сервере. Просмотры серверов осуществляются периодически, например раз в месяц, раз в две недели;
- ◆ индексные массивы и копии текстов просмотренных страниц, хранящиеся в поисковой системе;
- ◆ программа, которая, просматривая в соответствии с запросом пользователя индексные массивы, отбирает и выдает потребителю найденные документы.

В *каталогах* имеются иерархические тематические рубрики. Отнесение серверов к тем или иным рубрикам каталога человеком. Пользователь ищет информацию в каталоге вручную, используя рубрики.

В связи с тем, что в средствах поиска в Интернете не используются информационно-поисковые языки, на которых могли бы быть описаны исходные документы и запросы, полнота поиска в Интернете с учетом указанных выше поисковых средств будет значительно ниже, чем в документальных системах, построенных на базе информационно-поисковых языков.

В 2000 г. специалисты компаний AltaVista, IBM и Compaq исследовали ресурсы и гиперсвязи существующего информационного пространства WWW. Просмотрев с помощью поисковых средств AltaVista свыше 600 млн web-страниц и 1,5 млрд ссылок, размещенных на этих страницах, они пришли к выводу, что исследуемое пространство состоит из следующих компонентов:

- ◆ центральное ядро — тесно связанные между собой web-страницы, с каждой

из которых можно попасть на любую другую (27 %);

- ◆ отправные страницы. В них могут быть ссылки, ведущие к ядру, но из ядра к отправным страницам попасть нельзя (22 %);
- ◆ окончательные web-страницы, к которым можно прийти по ссылкам из ядра, но к ядру от них попасть нельзя (22 %);
- ◆ полностью изолированные от центрального ядра страницы (22 %);
- ◆ web-страницы, не пересекающиеся с остальными ресурсами Интернета (7 %).

Исследования показали, что при увеличении общего объема информационных ресурсов Интернета установленные отношения компонентов остаются прежними. Проведенный анализ позволяет сделать вывод о том, что информационное пространство Интернета является достаточно сложным и неоднородным. К отдельным ресурсам Интернета поисковые машины не имеют доступа.

Информационные ресурсы Интернета делятся на «видимую» и «невидимую» части сайтов.

Перечисленные особенности информационных ресурсов позволяют сделать вывод о том, что эффективность поиска информации в Интернете существенно уступает эффективности поиска в документальных информационно-поисковых системах, использующих специальные информационно-поисковые языки, и эффективности поиска в профессиональных базах. Организация информации в профессиональных базах отличается от организации информации в Интернете в первую очередь тем, что информация накапливается и постоянно обновляется в базах данных, которых в настоящее время свыше 13 тыс. В каждой базе собрана специфичная информация. Отбор достоверных источников и накопление информации ведут информационные агентства-генераторы. Они же поддерживают эти базы в актуальном состоянии, то есть обновляют. Получение информации из баз потребителями обеспечивают агентства-поставщики. Потребителю предоставляется язык запроса и документация, характеризующая базы данных, которая включает следующие сведения по каждой базе: название, отражающее вид информации, хранимой в базе, с какого времени ведется база, объем накопленной информации, период обновления, источники информации.

Указанные особенности обеспечивают высокие показатели по достоверности, полноте и точности предоставляемой информации.

Выводы по главе

1. Мировой рынок деловой информации представлен более чем 7 тыс. крупных информационных агентств, в полном объеме обеспечивающих разнообразные потребности в деловой информации различных видов бизнеса. Эти агентства предоставляют услуги доступа к 13,5 тыс. информационных баз, содержащих биржевую и финансовую информацию, сведения о фирмах, товарах, услугах, рынках сбыта, статистическую, научно-техническую, правовую информацию и деловые новости. К большинству баз имеется режим непосредственного доступа из любой точки мира.

2. Основная информация из профессиональных баз (свыше 85 %) предос-

тавляется на коммерческой основе. Несмотря на относительно высокие цены, спрос на деловую информацию постоянно растет. За пять лет, с 1992 по 1997 г., доходы от продаж информации через Интернет возросли в 28 раз и, по мнению экспертов, в ближайшее время должны приблизиться к 20 млрд долл. США.

3. Информационные ресурсы стали одним из важнейших ресурсов государств. США — лидер мировой экономики — формирует более 70 % мировых ресурсов деловой информации.

4. Бизнесмены в развитых странах стараются предоставить информацию о своей фирме в максимально возможном объеме. Отсутствие или недостаток информации о фирме будет, как правило, преградой на пути делового сотрудничества с такой фирмой.

Лекция 7

Глава 2. Информационный рынок Российской Федерации

2.1. Правовые основы информационной работы в РФ

Основным законом, определяющим правовые основы информационной работы в России, является Федеральный закон от 20 февраля 1995 г. № 24-ФЗ «Об информации, информатизации и защите информации».

В соответствии с этим законом информационные ресурсы делятся на государственные и негосударственные. Собственниками государственных информационных ресурсов являются Российская Федерация и субъекты РФ. Эти ресурсы создаются, приобретаются, накапливаются за счет федерального бюджета, бюджетов субъектов РФ, а также иными установленными законом способами.

Физические и юридические лица являются собственниками тех документов (массивов документов), которые созданы за счет их средств, приобретены ими на законных основаниях, получены в порядке дарения или наследования. Эти ресурсы являются негосударственными.

Граждане, органы государственной власти, органы местного самоуправления, организации и общественные объединения обязаны представлять документированную информацию органам и организациям, ответственным за формирование и использование государственных информационных ресурсов.

Перечни представляемой в обязательном порядке документированной информации и перечни органов и организаций, ответственных за сбор и обработку федеральных информационных ресурсов, утверждает правительство РФ. Порядок и условия обязательного представления информации доводятся до сведения граждан и организаций.

По категориям доступа информация делится на открытую информацию и информацию с ограниченным доступом. Информация с ограниченным доступом делится, в свою очередь, на *информацию, отнесенную к государственной тайне, и конфиденциальную.*

Отнесение информации к государственной тайне осуществляется в соот-

ветствии с Законом РФ от 21 июля 1993 г. № 5485-1 «О государственной тайне» (с изменениями, внесенными Законом от 6 октября 1997 г. № 131-ФЗ).

Государство имеет право выкупа документированной информации у физических и юридических лиц в случае отнесения этой информации к государственной тайне. Собственник информационных ресурсов, содержащих сведения, отнесенные к государственной тайне, вправе распоряжаться этой собственностью только с разрешения соответствующих органов государственной власти.

Запрещено относить к информации с ограниченным доступом:

- ◆ законодательные и другие нормативные акты, устанавливающие правовой статус органов государственной власти, органов местного самоуправления, организаций, общественных объединений, а также права, свободы и обязанности граждан, порядок их реализации;
- ◆ документы, содержащие информацию о чрезвычайных ситуациях, экологическую, метеорологическую, демографическую, санитарно-эпидемиологическую и другую информацию, необходимую для обеспечения функционирования населенных пунктов, производственных объектов, безопасности граждан и населения в целом;
- ◆ документы, содержащие информацию о деятельности органов государственной власти и местного самоуправления, об использовании бюджетных средств и других государственных и местных ресурсов, о состоянии экономики и потребностях населения, за исключением сведений, относящихся к государственной тайне;
- ◆ документы, накапливаемые в открытых фондах библиотек и архивов, информационных системах органов государственной власти, органов местного самоуправления, общественных объединений, организаций, представляющие общественный интерес или необходимые для реализации прав, свобод и обязанностей граждан.

Персональные данные (информация о гражданах) относятся к конфиденциальной информации. Подлежит обязательному лицензированию деятельность негосударственных организаций и частных лиц, связанная с обработкой и предоставлением пользователям персональных данных. Порядок лицензирования определяется законодательством РФ.

Другим важным законом, определяющим порядок формирования фондов государственной библиотечной системы и органов научно-технической информации, является Федеральный закон от 29 декабря 1994 г. № 77-ФЗ «Об обязательном экземпляре документов» и Постановление Правительства РФ от 24 июля 1995 г. № 739 «Об обязательных экземплярах изданий».

Обязательные экземпляры – экземпляры различных видов тиражированных документов, подлежащие передаче производителями в соответствующие учреждения и организации в порядке и в количестве, установленных Федеральным законом.

Обязательные бесплатные экземпляры — экземпляры различных видов документов, подлежащие безвозмездной передаче их производителями в соответствующие учреждения и организации в порядке и в количестве, установленных Федеральным законом.

Лекция 8

2.2. Российские организации информационных ресурсов

Научно-технический центр «Информрегистр» Государственного комитета РФ по связи и информатизации

Работы по учету и регистрации баз и банков данных и по ведению Государственного регистра баз данных осуществляются «Информрегистром» в соответствии с Постановлением Правительства РФ от 28 февраля 1996 г. № 226 «О государственном учете и регистрации баз и банков данных». Порядок ведения государственного учета и регистрации баз данных определен Временным положением о государственном учете и регистрации баз и банков данных (государственный учет и регистрацию баз и банков данных правовой информации ведет научно-технический центр правовой информации «Система» Федерального агентства правительственной связи и информации при Президенте РФ (ФАПСИ) Указом Президента РФ от 12 марта 2003 г. ФАПСИ было расформировано.

Центр «Информрегистр», кроме ведения Государственного регистра баз данных, оказывает методическую помощь в подготовке сведений о создаваемых и эксплуатируемых базах и банках данных для их государственного учета и регистрации, публикует сведения о зарегистрированных базах и учтенных банках данных, оказывает информационные услуги на основе базы метаданных и Государственного регистра баз данных.

Сведения о российском информационном рынке предоставляет также ряд коммерческих структур.

ООО «Международное Бюро Информации и Телекоммуникаций» (МБИТ)

(www.mbt.ru) предлагает на информационном рынке следующие продукты и услуги:

- ◆ «Российская энциклопедия информации и телекоммуникаций». Электронная версия находится на сайте «Инфопартнер»;
- ◆ web-каталог деловой информации;
- ◆ web-каталог бизнес-ресурсов по рынку товаров и услуг;
- ◆ информационно-поисковые, аналитические и консультационные услуги.

Центр информационной поддержки предпринимательства

выпустил компакт-диск «Кто есть кто на рынке деловой информации России». Диск содержит базу данных по организациям, предоставляющим информационные услуги. Помимо описаний более чем 4100 организаций на диске размещены:

- ◆ база данных 250 деловых информационных изданий;
- ◆ информация по органам управления РФ;
- ◆ база данных по коммерческим организациям Подмосковья.

Каталог предназначен в основном для руководителей и среднего управленческого звена коммерческих и государственных организаций, снабженческих, маркетинговых, информационных и аналитических подразделений.

Другие источники

Правительство Москвы, Комитет по телекоммуникациям и СМИ, Межрегиональный центр коммерческой информации Larics выпускают справочник «Москва. Администрация, Предприятия и организации, Информационные агентства, Средства Массовой информации, Рекламные службы».

Обзорную информацию о состоянии информационных ресурсов России публикует ежемесячный журнал «Информационные ресурсы России», выпускаемый Российским объединением информационных ресурсов научно-технического развития.

Агентством деловой информации «Бизнес-карта» в конце 1998 г. выпущен первый номер нового журнала «Информация и бизнес», в котором публикуются обзоры информационного рынка, рейтинги участников, предлагаемые услуги. В настоящее время этот журнал выходит ежеквартально.

Лекция 9, 10

2.3. Государственные информационные ресурсы

Государственные информационные ресурсы — это ресурсы, которые как элемент имущества находятся в собственности государства. Государственные информационные ресурсы делятся на:

- ◆ федеральные ресурсы;
- ◆ ресурсы, находящиеся в совместном ведении Российской Федерации и субъектов РФ;
- ◆ ресурсы субъектов РФ.

Государственные информационные ресурсы, являясь важнейшим фактором, влияющим на выполнение всех функций государства, обеспечивают выполнение следующих основных задач:

- ◆ государственного управления;
- ◆ обеспечения прав и безопасности граждан;
- ◆ поддержки социально-экономического развития страны, развития культуры, науки, образования и т. д.

Ряд федеральных органов управления можно назвать чисто «информационными», поскольку их главной задачей является формирование государственных информационных ресурсов. В них относятся: Министерство культуры РФ, Государственный комитет РФ по статистике (Госкомстат), Государственный комитет РФ по стандартизации, метрологии и сертификации (Госстандарт), Министерство природных ресурсов РФ, Федеральная служба по гидрометеорологии и мониторингу окружающей среды (Росгидромет) и др.

Государственные информационные ресурсы делят на две группы:

- 1) предназначенные для решения задач конкретного органа управления опреде-

ленного звена;
2) ориентированные внешнему пользователю.

Библиотечная сеть РФ

Библиотечная сеть РФ насчитывает свыше 150 тыс. библиотек и включает:

- ◆ публичные библиотеки всех уровней (федерального, регионального и муниципального);
- ◆ систему научно-технических библиотек и справочно-информационных фондов, входящую в российскую государственную систему научно-технической информации (ГСНТИ);
- ◆ информационно-библиотечную систему Российской академии наук (РАН);
- ◆ библиотечную систему высших учебных заведений;
- ◆ сеть муниципальных библиотек;
- ◆ сеть сельскохозяйственных библиотек;
- ◆ другие системы и сети.

При построении систем сочетаются отраслевой и региональный принципы.

На федеральном уровне крупнейшими *публичными библиотеками* являются:

- ◆ Российская государственная библиотека (РГБ) — 38 млн единиц хранения;
- ◆ Российская национальная библиотека — 30 млн единиц хранения;
- ◆ Всероссийская государственная библиотека иностранной литературы.

На региональном уровне имеется 217 центральных библиотек всех 89 субъектов РФ и 49,7 тыс. публичных городских и муниципальных библиотек.

Наряду с обслуживанием индивидуальных пользователей библиотеки также ведут обслуживание организаций.

Сеть *научно-технических библиотек* трехуровневая, включает 7 тыс. библиотек с фондом 10 млн единиц хранения (без координационного центра — ГПНТБ).

Библиотеки РАН объединены в четыре системы:

- ◆ система библиотек Сибирского региона, включающая 67 библиотек с фондом 13,5 млн единиц хранения;
- ◆ система библиотек Санкт-Петербургского региона, включающая 41 библиотеку с фондом 16,5 млн единиц хранения;
- ◆ отраслевая система библиотек по естественным наукам — 13 млн единиц хранения;
- ◆ отраслевая система библиотек институтов РАН гуманитарного и общественного направлений — 13 млн единиц хранения.

Библиотечная сеть вузов включает 500 библиотек — 300 млн единиц хранения.

Состояние библиотечной сети в период реформ резко ухудшилось. Практически прекратилось финансирование пополнения фондов со стороны государства.

Информационные ресурсы архивного фонда

Архивный фонд РФ (АФ РФ) находится в ведении Федеральной архивной службы (Росархив). Объем фондов — 460 млн единиц хранения. Постоянное хранение архивных документов осуществляют государственные и муниципальные архивы, государственные музеи и библиотеки, учреждения системы РАН. Временное хранение архивного фонда документов осуществляют министерства, ведомства, учреждения, организации и предприятия, отнесенные установленным порядком к источникам комплектования государственных и муниципальных архивов.

В архивных фондах хранится управленческая документация (78 %), документы по личному составу (19 %), научно-техническая информация (2%).

Статистическая информация

Государственная система статистики РФ находится в настоящее время в стадии реформирования, разработки новых показателей и перехода на методологическую базу, используемую в мировой практике.

В процессе реформ удалось сохранить единую систему государственной статистики, в которую входят Госкомстат России (федеральный орган исполнительной власти, осуществляющий руководство системой государственной статистики РФ); его органы в республиках, краях, областях, автономной области и автономных округах, городах Москве и Санкт-Петербурге, районах и городах; подведомственные предприятия, организации и учебные заведения.

Единая система государственной статистики включает 2250 организаций: 7 – на федеральном уровне, 87 комитетов на региональном уровне и 2156 региональных (городских) отделов статистики.

Основными задачами Госкомстата России являются:

- ◆ разработка научно обоснованной статистической методологии, соответствующей потребностям общества на современном этапе, а также международным стандартам;
- ◆ предоставление официальной статистической информации Президенту РФ, Правительству РФ, Федеральному собранию ГД РФ, федеральным органам исполнительной власти, общественности, а также международным организациям;
- ◆ предоставление всем пользователям равного доступа к открытой статистической информации путем распространения официальных докладов о социально-экономическом положении Российской Федерации, субъектов РФ, отраслей и секторов экономики; публикации статистических сборников и других материалов.

Научно-техническая информация

Среди ведомств, формирующих информационные ресурсы в области науки и техники, бесспорным лидером является Государственный комитет РФ по науке и технологиям (ГКНТ России). Созданная под его руководством в 1960-1970-е гг. государственная система научно-технической информации (ГСНТИ) была одной из крупнейших информационных систем в стране и не знала равной

себе в мире. Кризис 1990-х гг. затронул ГСНТИ особенно остро. Система, включавшая около двухсот специализированных организаций, а также десятки тысяч служб на предприятиях, фактически развалилась. Из почти 80 отраслевых институтов информации функционируют не более десятка. Федеральные информационные центры сохранились почти все, однако сократили масштабы своей деятельности в десятки раз.

В настоящее время государственная система научно-технической информации представляет собой совокупность научно-технических библиотек и организаций — юридических лиц независимо от формы собственности и ведомственной принадлежности, специализирующихся на сборе и обработке научно-технической информации и взаимодействующих между собой с учетом принятых на себя обязательств.

В состав государственной системы научно-технической информации входят федеральные органы научно-технической информации и научно-технические библиотеки, отраслевые органы научно-технической информации и научно-технические библиотеки, региональные центры научно-технической информации.

К федеральным органам научно-технической информации и научно-техническим библиотекам относятся:

Всероссийский научно-технический информационный центр Министерства науки и технологий РФ (ВНТИЦ);

Всероссийский научно-исследовательский институт межотраслевой информации (ВИМИ);

Российское объединение информационных ресурсов научно-технического развития (Росинформресурс) Министерства науки и технологий РФ;

Всероссийский институт научной и технической информации РАН и Министерства науки и технологий РФ;

Государственная публичная научно-техническая библиотека Министерства науки и технологий РФ, Библиотека РАН, Библиотека по естественным наукам РАН и Государственная публичная научно-техническая библиотека Сибирского отделения РАН — по естественным и техническим наукам;

Государственная центральная научная медицинская библиотека Министерства здравоохранения РФ;

Всероссийский научно-исследовательский институт информации и технико-экономических исследований агропромышленного комплекса Министерства сельского хозяйства и продовольствия РФ и Центральная научная сельскохозяйственная библиотека Российской академии сельскохозяйственных наук;

Институт научной информации по общественным наукам РАН;

Российский федеральный геологический фонд, Всероссийский научно-исследовательский институт экономики минерального сырья и недропользования и Всероссийская геологическая библиотека Министерства природных ресурсов РАН;

Федеральный фонд государственных стандартов, общероссийских классификаторов технико-экономической информации, международных (региональных) стандартов, правил, норм и рекомендаций по стандартизации, нацио-

нальных стандартов зарубежных стран Государственного комитета РФ по стандартизации, метрологии и сертификации — по нормативным документам по стандартизации, метрологии и сертификации;

Федеральный институт промышленной собственности и Информационно-издательский центр Российского агентства по патентам и товарным знакам — по патентной документации, относящейся к изобретениям, полезным моделям, промышленным образцам, товарным знакам, знакам обслуживания и наименованиям мест происхождения товаров, а также по документации, относящейся к зарегистрированным программам для ЭВМ, базам данных и топологиям интегральных микросхем;

Российская книжная палата Государственного комитета РФ по печати;

Российский государственный архив научно-технической документации Федеральной архивной службы РФ;

Научно-технический центр «Информрегистр» — по электронным изданиям;

Институт промышленного развития («Информэлектро»);

ряд других всероссийских государственных структур по отраслям промышленности, сельского хозяйства и видам деятельности.

Порядок формирования федеральных информационных фондов указанных организаций устанавливается законодательством РФ.

Лекция 12, 13

2.4. Правовая информация

Информационные ресурсы в области права стали бурно развиваться на начальном этапе перехода к рынку. Особенности развития нашей страны в настоящее время характеризует высокий объем ежегодных публикаций — около 11 тыс. правовых документов.

Источником информационных ресурсов в области права являются в основном государственные учреждения. Широкий круг пользователей предъявляет к средствам информационного обеспечения в области права следующие основные требования:

- ◆ максимальная полнота представления материалов по интересующему пользователя вопросу;
- ◆ постоянное обновление информационных ресурсов по мере выхода новых правовых актов;
- ◆ возможность получения комментариев по отдельным положениям правовых актов;
- ◆ удобство доступа и работы с правовой информацией.

Реализацию этих требований обеспечивают компьютерные технологии. Однако автоматизированные информационные системы еще не могут быть использованы для процедуры официального опубликования. Одной из таких причин является то, что в России доступ к электронным источникам в настоящее время имеет ограниченный круг пользователей. Кроме того, еще не создан надежный механизм подтверждения подлинности официального нормативного

акта. Несмотря на эти трудности, в стране широкое распространение получили справочные правовые системы.

Компьютерная справочная правовая система (СПС) — это программный комплекс, включающий в себя массив правовой информации и программные инструменты, позволяющие специалисту работать с этим массивом (производить поиск конкретных документов или их фрагментов, выводить информацию на печать и т. д.).

В соответствии с Указом Президента РФ от 4 августа 1995 г. № 808 задачи сбора, обработки, хранения, анализа правовой информации и организации ее использования возложены на Министерство юстиции РФ.

Министерство юстиции РФ располагает следующими основными информационными ресурсами:

- ◆ программно-технологическим комплексом «Фонд», содержащим 340 тыс. правовых актов СССР, РСФСР, Российской Федерации, правовых органов государственной власти субъектов РФ, правовых актов государств — участников СНГ;
- ◆ базой данных действующего российского законодательства «Эталон», содержащей редакции 30 тыс. наиболее актуальных нормативных актов;
- ◆ фондом правовых актов на бумажных носителях Министерства юстиции РФ, содержащим более 1 млн документов;
- ◆ Государственным реестром общественных объединений и религиозных организаций, имеющих общероссийский и международный статус;
- ◆ базой данных судебной статистики.

Полнота баз данных Министерства юстиции РФ составляет 0,93 % от общего количества изданных нормативных правовых актов.

Значительные ресурсы правовой информации сосредоточены в Федеральном агентстве правительственной связи и информации (ФАПСИ).

Справочные правовые системы, распространяемые негосударственными структурами

Наиболее крупными компаниями, обеспечивающими поставку потребителям СПС, являются АО «Консультант Плюс», НПП «ГАРАНТ-Сервис», ЦКР «Кодекс». Также на рынке СПС активно работают такие фирмы, как INTRALEX, ЗАО «Референт-Сервис», издательство «Дело и Право», «Информационные системы и технологии», компания «1С» и ряд других.

«Консультант Плюс»

Компания «Консультант Плюс» - крупнейшее предприятие на информационном рынке страны. Сеть насчитывает свыше 300 информационных сервисных центров в различных регионах РФ. В справочные системы «Консультант Плюс» входит более 500 тыс. документов. Каждый месяц в системы передается до 8 тыс. новых документов.

Основные справочные системы «Консультант Плюс» разделяются на *системы по законодательству* и *системы поддержки принятия решений*.

«ГАРАНТ-Сервис»

НПП «ГАРАНТ-Сервис» предлагает комплекс «Справочно-правовая система ГАРАНТ» — базу данных, подготавливаемую на русском и английском языках для работы как на компьютерах IBM, так и на Macintosh, с использованием гипертекстовой среды. Одной из отличительных особенностей компании является общероссийский масштаб деятельности: более 207 представительств в 120 регионах РФ и стран СНГ. Постоянными партнерами компании являются 186 органов власти и управлений федерального и регионального уровней. Двусторонние договоры предусматривают обмен информационными ресурсами и ведение совместных проектов. Программа сотрудничества с известными производителями компьютерных бухгалтерских систем предусматривает как создание совместных продуктов, так и проведение совместных проектов.

Система «ГАРАНТ» предлагает своим клиентам шесть вариантов комплектов, различающихся между собой размером и количеством информационных правовых блоков.

Центр компьютерных разработок «КОДЕКС»

Центр предлагает профессиональные юридические системы «КОДЕКС»:

- ◆ «Кодекс-Профессионал» — полное собрание российского законодательства, материалов судебной и арбитражной практики;
- ◆ «Кодекс-Правовед» — действующие документы РФ;
- ◆ «Кодекс-Универсал» — нормативно-правовые акты РФ и разъяснения к ним. Все документы гражданского и экономического законодательства;
- ◆ «Кодекс-Юрисконсульт» — документы экономического и гражданского законодательства;
- ◆ «Международное право» — международные договоры, документы международных организаций, регламенты международных арбитражных институтов;
- ◆ «Помощник бухгалтера» — бухгалтерский учет и налогообложение предприятий, консультации аудитора, налоговый календарь;
- ◆ «Таможенный консультант» — таможенное законодательство РФ;
- ◆ «Москва» — законодательство г. Москвы, включая архивные документы;
- ◆ «Москва. Местные налоги» — документы по местным налогам г. Москвы;
- ◆ «Санкт-Петербург» — законодательство г. Санкт-Петербурга, включая архивные документы;
- ◆ «Санкт-Петербург. Местные налоги» — документы по местным налогам г. Санкт-Петербурга.

Агентство INTRALEX

Юридическое информационное агентство INTRALEX работает над созданием правовых информационных систем для персональных компьютеров. С 1989 агентство сотрудничает с высшими органами законодательной, исполнительной и судебной власти РФ. Агентство INTRALEX зарегистрировано в Министерстве печати РФ в качестве средства массовой информации по распространению нормативных актов органов государственной власти и управления

РФ.

Разработанный агентством программный комплекс «ЮСИС» («Юридическая справочно-информационная система») — это полнотекстовая база данных действующего законодательства РФ с развитым системным аппаратом поиска информации, которая ежедневно обновляется по телекоммуникационным каналам связи. «ЮСИС» предоставляет информацию по всем областям права Российской Федерации.

Три варианта комплектации системы обеспечивают оптимальное соотношение назначения, объема информации и цены.

Лекция 14, 15

2.5. Биржевая и финансовая информация

Информационные ресурсы в области биржевой информации включают справочники и базы данных по организациям, оказывающим соответствующие услуги, справочники и базы данных по выпускам ценных бумаг, финансовым услугам, печатные справочники и базы данных, характеризующие состояние рынка ценных бумаг и соответствующие коммерческие предложения, а также специальные базы текстовой информации.

Основными потребителями биржевой информации в России являются профессиональные участники биржевого, фондового и финансового рынка — оптовики, профессиональные инвесторы, а также аналитические и консультационные службы, ориентированные на их обслуживание.

Рынок финансовой информации в России часто рассматривается отдельно от рынка биржевой информации, так как эти виды информации имеют определенные отличия.

Финансовая информация включает информацию по обменным курсам валют и по состоянию рынка денег и соответствующих инструментов.

Особенностью биржевой и финансовой информации — ее ежеминутное изменение, отражающее события, происходящие на финансовом рынке. Главное условие успеха деятельности на финансовом рынке — наличие оперативной и достоверной информации.

Всех производителей информационной продукции на российском рынке биржевой и финансовой информации можно разделить на несколько групп.

1. поставщики «сырой» финансовой информации — финансовые институты и службы, дающие первичную и оперативную биржевую и другую информацию - государственные органы управления, такие как Центральный банк РФ и Министерство финансов РФ; банки, биржи.

2. информационные, консультационные агентства, информационные центры и отделы финансовых институтов и объединений, например Агентство экономических новостей (ЭН), Международное агентство финансовой информации (МАФИ), агентство «Прайм» и др.

3. издательские дома, газетные объединения и информационные службы, такие как «Финансовая газета», издательский дом «Коммерсантъ», газета

«Экономика и жизнь», агентство «ИТАР-ТАСС» и ряд других.

Основные зарубежные производители информационных продуктов и услуг на финансовом рынке РФ

На российском рынке биржевой и финансовой информации действуют ряд крупных зарубежных информационных агентств, в том числе: «Рейтер» (Reuters), «Доу-Джонс Телерейт» (DowJones Telerate), «Тен-фор» (Tenfore) и «Блумберг» (Bloomberg).

Агентство «Рейтер»

Является мировым лидером в области предоставления предпринимательским кругам биржевой, финансовой информации и деловых новостей. Информация в агентство поступает в режиме реального времени со 180 бирж, рынков ценных бумаг и от 4 тыс. организаций из 80 стран мира. Новости агентство «Рейтер» получает из своих 128 информационных бюро в 86 странах мира, в которых работает 1800 репортеров.

Компания «Доу-Джонс Телерейт»

Является одной из крупнейших мировых компаний, специализирующихся в области сбора, обработки и анализа биржевой и информации. Компания включает в себя следующие информационные подразделения:

- ◆ Dow Jones Telerate (электронная система);
- ◆ газета «Уолл-стрит Джорнел» (Wall Street Journal) (распространяется также и по электронным сетям);
- ◆ 21 региональная газета.

В 1994 г. открылось московское представительство DJT.

Основой информационного сервиса компании является унифицированное программное обеспечение для рабочей станции трейдера — **Dow Jones Workstation**. Вне зависимости от того, какой из трех информационных пакетов (финансовый, сырьевой или энергетический) используется, а также независимо от принятой конфигурации системы пользовательский интерфейс и структура информационных потоков остаются практически неизменными.

Агентство «Тенфор»

Международное агентство финансовой информации «Тенфор» (Tenfore) является датско-швейцарской компанией, которая была создана в 1987 г. «Тенфор» имеет представительства в 36 странах, в том числе и в России. Московское представительство компании работает с июня 1994 г. Информация из различных источников — международных бирж, банков, информационных агентств — поступает в вычислительный центр в Копенгагене, а затем передается по спутниковой сети потребителю.

Агентство «Блумберг» (Bloomberg)

Информационное агентство «Блумберг» основано в Нью-Йорке в 1982 г. Имеет региональные отделения в Лондоне и Токио. Агентство выпускает специальный журнал. Пользователям предоставляется биржевая и финансовая информация со всего мира, большое внимание уделено новостям. С 1994 г. введен раздел «Состояние ценных бумаг на рынке России». Система дает пользователю возможность работы с десятью различными методами технического анализа. Ос-

новным достоинством предоставляемых агентством «Блумберг» услуг является высокое качество и скорость доведения до потребителя финансовой информации.

Российские информационные агентства и службы, предлагающие продукты и услуги в сфере биржевой и финансовой информации

*Информационное агентство «РосБизнесКонсалтинг»
(RosBusinessConsulting)*

Ведущий российский интернет-холдинг «РосБизнесКонсалтинг» (РБК) был создан в 1992 г. и на сегодняшний день уверенно лидирует на рынках предоставления информации массовому пользователю и разработки интернет-решений для среднего и крупного бизнеса.

Информационный портал агентства — www.rbc.ru. Ежедневно на сервер приходят более 200 тыс. посетителей. Ежемесячная аудитория превышает 700 тыс. человек. Это квалифицированные специалисты, менеджеры, руководители компаний, журналисты и многие другие, которым необходима оперативная финансовая, экономическая и политическая информация.

РБК предоставляет большой объем финансовых данных по России, странам СНГ и Запада. Постоянно обновляются котировки, курсы, индикаторы по всем сегментам финансового рынка (валютный, кредитный, государственных ценных бумаг, акций и векселей), а профессионалы могут следить за трансляцией хода торгов на ведущих биржевых площадках. Специальные терминалы позволяют эффективно работать с информацией в режиме реального времени, интерактивные информационные системы дают возможность участникам рынка самостоятельно выставлять котировки. Круглосуточно на сервере www.rbc.ru в режиме реального времени выпускаются оперативные новости, освещающие все важнейшие события. «РосБизнесКонсалтинг» располагает обширной корреспондентской сетью и всю информацию получает непосредственно из первоисточников, в число которых входят государственные органы, крупнейшие российские и международные финансовые институты и коммерческие организации. Несколько раз в день обновляются аналитические материалы, комментарии и прогнозы, публикуются мнения специалистов, а также тематические статьи, над которыми работает большой штат аналитиков. РБК регулярно проводит on-line-интервью с ведущими российскими политическими деятелями и предпринимателями. «РосБизнесКонсалтинг» активно сотрудничает с известными международными информационными структурами, среди которых можно назвать Reuters Business Briefing, Bridge Telerate (бывший Dow Jones Markets), Tenfore, CQG и AFX News.

АО «АК&М»

Ведет свою историю с декабря 1990г, содержит свыше 150 сотрудников московского офиса, корпункты в 45 регионах, более 20 регулярных изданий новостного и аналитического характера. Информацию агентства использует аппарат правительства РФ, комитеты Государственной Думы РФ и Совета Федера-

ции, Администрация Президента РФ, Министерство экономического развития и торговли РФ, Центральный банк РФ, страховые и инвестиционные компании, крупнейшие банки и предприятия, а также СМИ.

«ФИНМАРКЕТ»

Информационное агентство «ФИНМАРКЕТ» специализируется на информационном обеспечении профессионалов финансовой отрасли. Фирма была образована в 1994 г.

Лекция 16

Глава 3

Глобальные сети

Совокупности вычислительных машин, объединенных коммуникационной средой, охватывающей значительные по расстоянию территории, получили название *глобальных компьютерных сетей*.

Лидером глобальных сетей является Интернет. Его основу составляют высокоскоростные магистральные сети. Независимые сети подключаются к магистральной сети через точки сетевого доступа NAP (Network Access Point). Автономные сети могут образовывать компании, специализирующиеся на предоставлении услуг доступа в сети Интернет – провайдеры. Например, в России – компания Relcom.

Каждый компьютер в Интернет имеет уникальный IP-адрес. IP-адрес состоит из четырех байтов и записывается в виде четырех десятичных чисел, разделенных точками, например:
194.85.120.64

IP-адрес состоит из двух логических частей: номера сети и номера узла. В зависимости от количества байтов выделяют несколько классов IP-адресов.

В сети Интернет в основном используется семейство протоколов TCP/IP.

3.1. Протокол TCP/IP

Протокол TCP/IP предназначен для соединения сетей с разнородным оборудованием. Это единственное средство коммуникации, позволяющее связываться рабочим станциям всех типов – PC, Macintosh, Unix. Он необходим для выхода в Internet. Фактически TCP/IP является набором протоколов. Наиболее известные из них TCP и IP.

Протокол IP (Internet Protocol) функционирует на сетевом уровне, предоставляя различным станциям стандартный набор правил и спецификаций для межсетевой маршрутизации с помощью IP адресов.

Протокол управления передачей данных TCP (Transmission Control Protocol) работает на транспортном уровне модели OSI, обеспечивая прием сетевой и преобразование информации в требуемый на данном уровне формат. Таким образом, IP задает правила установления соединения и обеспечивает соедине-

ние портов компьютера, TCP отвечает за интерпретацию данных.

Протокол TCP устанавливает между двумя компьютерами *дуплексное соединение* типа «точка-точка». Программы на каждом конце соединения используют собственный порт. Комбинация IP-адреса и номера порта называется *socket* (socket). Соединение устанавливается путем *трехкратного квитирования* (three-way handshake)

После установления соединения каждая программа может посылать другой программе поток байтов. Программа-отправитель не ожидает подтверждения каждого сегмента, а посылает несколько сегментов вместе и ждет первого подтверждения. Если программа-получатель должна отослать данные обратно отправителю, она может совместить подтверждения и данные в одних и тех же сегментах. Номера последовательности программы-отправителя являются не индексами сегментов, а индексами в потоке байтов. Программа-получатель отсылает обратно номера последовательности (в поле номера подтверждения), удостоверяя тем самым, что все байты приняты и объединены в правильной последовательности. Программа-отправитель заново пересылает неподтвержденные сегменты.

Каждая программа со своей стороны закрывает TCP-соединение, что должно быть подтверждено программой на другой стороне соединения. Программа не может получать байты по соединению, которое закрыто на другой стороне.

3.2. Управление электронной почтой в сети масштаба предприятия

Для многих электронная почта – наиболее популярная сетевая служба. Если все сети предприятия пользуются одним пакетом программ электронной почты, то проблем с совместимостью нет. Базовым протоколом на протяжении нескольких лет является простой протокол пересылки почты (SMTP), входящий в комплект TCP/IP. SMTP состоит всего из 14 команд. Он эффективен, но его возможности ограничены. В то время, как электронная почта существенно усовершенствовалась, особенно для ЛВС, SMTP практически стоял на одном месте. Для организации взаимодействия систем электронной почты, работающих на разных сетевых платформах, комитет CCITT выпустил ряд стандартов X.400 на средства электронной почты, которые стали частью модели OSI. X.400 выпущен в 1984 году, а в 1988 был опубликован комплект изменений, проходящий в настоящее время испытания. Изменения предусматривают возможность совместной передачи текстовых и нетекстовых элементов, например, электронных таблиц и графических файлов, аудио- и видеоинформации. Интерфейс пользователя сети с электронно-почтовой системой называется пользовательским агентом. Этот интерфейс позволяет сетевому пользователю отправлять и получать почту. Пользовательский агент строит «конверт» стандарта X.400 и помещает в соответствующие поля заголовка адресную информацию. Он ищет необходимые адреса и при необходимости создает списки рассылки. Затем пользовательский агент передает конверт с заголовками и сообщением агенту пересылки сообщений, который следует рассматривать как почтовые отделе-

ния, оказывающие типовой набор почтовых услуг при передаче электронных сообщений, включая передачу с промежуточным накоплением. Агент пересылки сообщений также выполняет контроль ошибок и обеспечивает правильность форматирования конверта и заголовков. Служба передачи сообщений предоставляет расширенные услуги по доставке, например доставку с проверкой или без проверки.

Комплект спецификаций X.400 включает протокол P1, описывающий конверт. Конверт X.400 имеет заголовок, который содержит сведения об отправителе, получателе, предмете сообщения и списки рассылки копий. Пользовательские агенты взаимодействуют с агентами пересылки сообщений посредством протокола P2, регламентирующего структуру сообщений и порядок доставки.

Для организации связи между пользователями сетей, работающих в разных пунктах на разных файловых серверах, требуется служба каталогов. ССІТТ разрабатывает стандарт на службы каталогов (X..500). Каталоги построены в виде информационного дерева каталогов. Корень этого дерева – чисто концептуальный объект. Под ним находятся страны, еще ниже – организации, а затем практически все внутрифирменные сведения. Компонентами дерева каталога управляют отдельные системные агенты каталога, которые могут быть распределенными или централизованными - это зависит от метода создания базы каталога.

Лекция 17

3.3. Протокол передачи гипертекста (НТТР). Поиск информации в Интернете.

Протокол передачи гипертекста (НТТР) – стандартный протокол для передачи документов между серверами и броузерами в системе World Wide Web.

Клиент устанавливает соединение с сервером по указанному номеру порта. Если в качестве клиента выступает браузер, номер порта указывается в URL-запросе. Если номер не указан, по умолчанию указывается порт 80.

Команды НТТР-клиента принято называть методами.

В Интернете существует два типа поисковых систем – классификаторы и поисковые машины.

Классификаторы хранят упорядоченные списки ссылок на Web-узлы. Их удобно использовать для достаточно общей информации.

Поисковые машины просматривают страницы, размещенные в Интернете, и составляют индексы используемых слов. Основным элементом поисковых машин является индексатор. К элементам поисковой машины также относятся индекс и аппарат поиска.

В последнее время Web-серверы, предназначенные для поиска информации сочетают возможности классификаторов и поисковых машин.

Методы индексирования делятся на статические, теоретико-информационные и вероятностные.

В статических методах документы рассматриваются как точки информационного пространства. Чем ближе группы методов, составляющие документ, тем ближе находятся отображающие их точки. В качестве терминов индексации выбираются понижающие плотность пространства документов.

Теоретико-информационные методы основаны на предположении о том, что наибольшую информационную ценность имеют наименее часто встречающиеся слова. Для оценки полезности термина применяются концепции теории информации.

Вероятностные методы предполагают наличие обучающего множества документов для оценки релевантности результатов обработки запросов. Обучающее множество применяется для вычисления весовых коэффициентов, получаемых путем оценки условной вероятности вхождения термина в данный документ в случае его релевантности (или нерелевантности).

При построении индекса реальные документы заменяются поисковыми образцами. При индексировании нетекстовой информации в поисковые образцы входят главным образом универсальные адреса ресурсов.

Формальную релевантность вычисляет система, на основании чего ранжируется выборка найденных документов.

Реальная релевантность – оценка самим пользователем ценности найденных документов.

Некоторые поисковые машины показывают дату, когда был проиндексирован тот или иной документ, что позволяет пользователю понять, насколько актуальным является ресурс.

Часто поисковые машины не включают определенные слова в свои индексы или в запросы пользователей. Такими словами, например, являются предлоги.

Лекция 18

3.4. Электронная коммерция

Электронная коммерция подразумевает использование технологий глобальных компьютерных сетей для ведения бизнеса. Она позволяет изменить практически все процессы, происходящие в современном бизнесе, интегрируя их в единое целое. Потребители могут искать, заказывать и оплачивать товары через Интернет, обмениваться информацией о товарах и услугах с другими пользователями. Правительственные организации используют Интернет для сбора налоговых деклараций и распространения официальной информации.

Первоначально под термином электронная коммерция понимались продажи товаров и переводы денежных средств с помощью компьютерных сетей. Однако банковские системы платежей являются закрытыми, между тем Интернет – открытая система. Сейчас термин «электронная коммерция» охватывает практически все аспекты ведения бизнеса, которые возможны с использованием Интернета.

Выделяют два аспекта электронной коммерции:

- электронная коммерция как торговля через Интернет;

- электронная коммерция как рынок.

Электронный рынок по многим параметрам похож на обычный рынок, основными компонентами которого являются:

игроки (компании, поставщики, брокеры, магазины и покупатели);

продукты (товары, услуги);

процессы (поставки, производство, маркетинг, конкуренция, распространение, потребление и др.).

Только элементы электронного рынка являются виртуальными. Такой рынок является очень эффективным и конкурентным и не нуждается в регулировании или внешнем вмешательстве.

Электронные продукты имеют высокую степень настраиваемости. На таком достаточно дифференцированном и сегментированном рынке цены будут всегда стремиться к максимальным, которые покупатель согласен заплатить.

Существует две основные области применения систем электронных платежей – обмен данными (заказы, счета и пр.) и электронный перевод денежных средств, который осуществляется между банками и имеет достаточно большой объем и относительно небольшое количество передаваемых полей, а также долгосрочные связи.

Существует три типа электронных платежных средств:

1. комбинация обычных и электронных платежей – платеж делается электронным способом, а уведомление обычным, или наоборот.
2. расширение традиционного способа передачи денежных средств - передача номеров кредитных карточек электронным способом и использование смарт-карт, которые могут хранить и передавать всю информацию о владельце.
3. различные виды цифровой наличности и электронных денег.

Разница между первыми двумя типами и третьим в том, что в последнем случае действительно переводятся деньги, а не информация о них.

Электронные деньги – некоторые серийные номера, представляющие реальную сумму денег. Они могут обмениваться на настоящие деньги

Смарт-карты являются небольшими устройствами, хранящими информацию, необходимую для совершения транзакций в рамках электронной коммерции – информацию о владельце, электронные ключи и пр.

Микроплатежи – особый вид денег, необходимый для оплаты микротранзакций. Стоимость одной такой транзакции может быть очень мала, и ее невозможно выразить в обычных денежных единицах. Это особенность электронной коммерции.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ МЕЖСЕССИОННОГО КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ

Межсессионный контроль осуществляется на основе выполнения домашних заданий, лабораторных заданий, промежуточных тестов, защиты реферата.

По итогам выполнения перечисленных видов работ в сроки, установленные деканатом (как правило, на 6-ой и 12-ой неделе семестра) преподавате-

лем выставляется аттестационная оценка по пятибалльной системе.

VII. КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА

Все виды занятий по дисциплине ведет кандидат технических наук, доцент Галаган Т.А.

СОДЕРЖАНИЕ

I. РАБОЧАЯ ПРОГРАММА	3
II. ГРАФИК ВЫПОЛНЕНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ	9
III. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ И ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ	10
IV. ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ	11
V. КОНСПЕКТ ЛЕКЦИЙ	50
VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ МЕЖСЕССИОННОГО КОНТРОЛЯ ЗНАНИЙ СТУДЕНТОВ	78
VII. КАРТА ОБЕСПЕЧЕННОСТИ ДИСЦИПЛИНЫ КАДРАМИ ПРОФЕССОРСКО-ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА	79

Татьяна Алексеевна Галаган,
доцент кафедры ИиУС АмГУ

Учебно-методический комплекс по дисциплине «Мировые информационные ресурсы и сети»

Изд-во АмГУ. Подписано к печати
Тираж Заказ

Формат 60x84/16. Усл. печ. л.