

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 20.01.2021 10:01:53

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabb75a943d14a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационных систем и технологий

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

« 18 » 01 2021 г.



ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУЧНОЙ И ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Методические указания по выполнению лабораторных работ
для студентов направления 43.04.03 Гостиничное дело

УДК 004.6

Составитель Д.О. Бобынцев

Рецензент

Кандидат технических наук, доцент Халин Ю.А.

Информационное обеспечение и компьютерные технологии в научной и образовательной деятельности: методические указания к выполнению лабораторных работ / Юго-Зап. гос. ун-т; сост.: Д.О. Бобынцев. Курск, 2021. 68 с. Библиогр.: с. 68.

Содержит методические указания к выполнению практических и лабораторных работ по дисциплине «Информационное обеспечение и компьютерные технологии в научной и образовательной деятельности». Указывается порядок выполнения работ, контрольные вопросы. Предназначен для студентов направления подготовки «Гостиничное дело».

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. 3,95. Уч.-изд. л. 3,58. Тираж 100 экз. Заказ 443 Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Содержание

1. Работа с презентациями в Microsoft Power Point.
2. Использование математического пакета MathCad.
3. Проектирование интернет-сайта.
4. Разработка базы данных.

Работа с презентациями в Microsoft Power Point

Цель работы: закрепить навыки создания презентаций в Microsoft PowerPoint для представления результатов НИР на научных мероприятиях и при защите квалификационных работ.

Современные способы организации презентаций средствами PowerPoint

В настоящее время существуют прикладные программы для подготовки выступлений или создания презентаций (демонстрационных материалов) с использованием компьютерных слайдов. К таким приложениям относится Microsoft PowerPoint, входящее в комплект Microsoft Office.

Каждая страница презентации называется слайдом. Презентация состоит из множества слайдов, которые хранятся в одном файле. Расширение файла «.ppt» или «.pptx». Презентации можно представлять в электронном виде, распечатывать в виде раздаточного материала (копии всех слайдов) или распространять через интернет. Для размещения презентации на сайте, необходимо сохранить ее как веб-страницу.

Основными элементами презентации являются слайды. С помощью редактора PowerPoint можно создавать слайды, в которых текст сочетается с таблицами, диаграммами, графическими объектами, картинками, рисунками, фотографиями, фильмами и звуком, видео клипами.

Каждый слайд презентации обладает свойствами, которые влияют на его отображение во время демонстрации:

- размер слайда;
- разметка слайда (расположение заголовков, текста и объектов на слайде);
- шаблон оформления (дизайн слайда);
- эффект перехода от слайда к слайду.

Презентацию можно создать несколькими способами:

- новая презентация (без разметки или на базе: макетов текста, макетов содержимого или макетов текста и содержимого);
- из шаблона оформления;
- из мастера автосодержания (на базе шаблонов презентации);
- из имеющейся на компьютере презентации.

Способы вывода презентации (стили презентации):

- Презентации на экране (для показа презентации используется компьютер или компьютер и мультимедийный проектор);
- WEB-страницы для размещения презентации на сайте;
- чёрно-белых прозрачек (для черно-белых иллюстраций к презентации);
- цветных прозрачек (для цветных иллюстраций к презентации);
- 35 - мм слайдов (пленки размером 35 мм).

По умолчанию приложение PowerPoint (рис. 1) открывается в режиме «Обычный». В левой части окна приложения находится область Структура или Слайды для переключения между режимами Слайды и Структура. По умолчанию в области Структура / Слайды устанавливается режим Слайды, т.е. отображается панель Слайды. В этом режиме в этой области отображаются миниатюрные изображения слайдов, входящих в презентацию.

В режиме Структура в этой области отображается иерархическая структура, содержащая заголовки и тексты слайдов презентации. Перед заголовком каждого слайда стоит номер и значок. Основной текст, включающий до пяти уровней отступов, расположен после каждого заголовка.

В центре приложения находится область слайда, в которой отображается слайд. Режим обычный - это основной режим для создания, редактирования и форматирования отдельных слайдов.

Ниже главного окна находится область заметок. В этой области к каждому слайду можно добавить заметки докладчика, которые не отображаются в режиме показа слайдов.

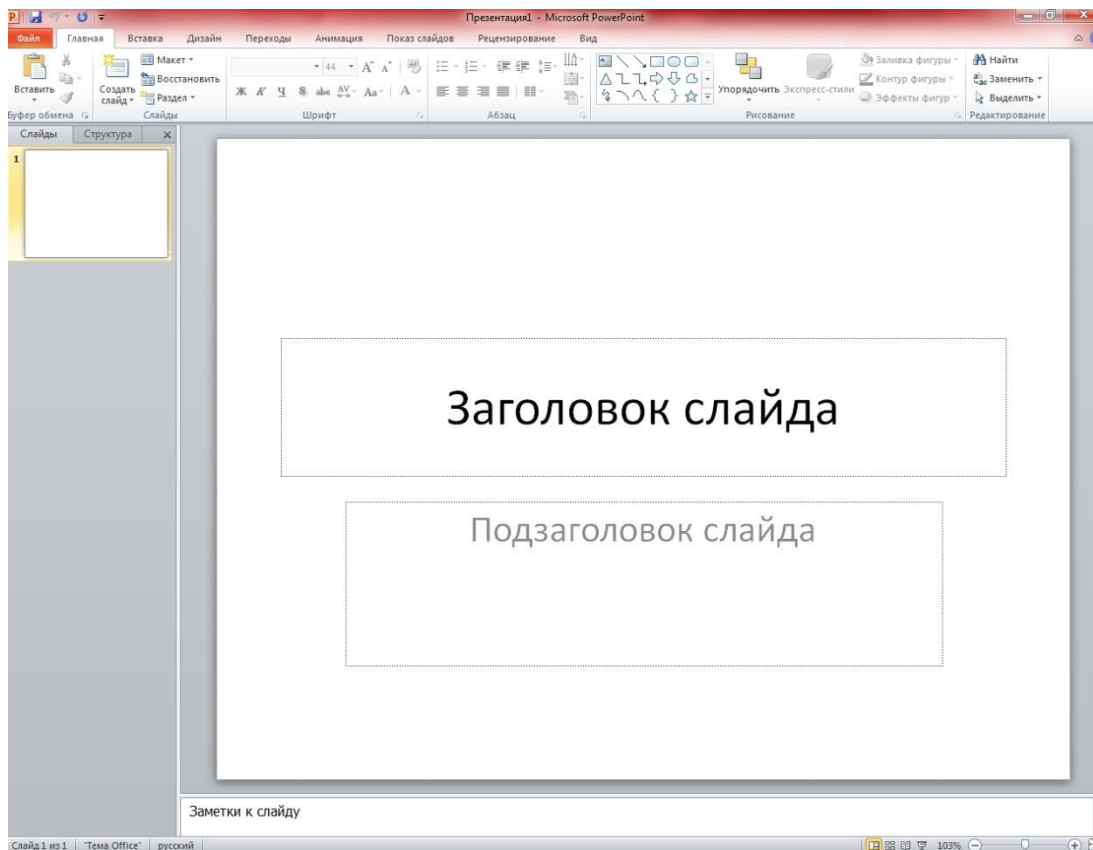


Рис. 1

Панель меню предоставляет доступ ко всем важным командам программы PowerPoint. Панели инструментов предоставляют быстрый доступ к используемым командам.

На панели «Главная» размещены основные инструменты создания слайда, форматирования, рисования и редактирования. При выборе кнопки «Создать слайд» в области задач отображаются варианты макета слайда (Титульный слайд, Заголовок и объект и т.д.) в зависимости от типа создаваемого слайда.

Бегунок линии прокрутки позволяет переходить между слайдами, а не по тексту в пределах одного слайда. Кроме того, во время перетаскивания бегунка редактор показывает номер и название каждого слайда.

Часть инструментов редактирования и оформления слайдов расположена в следующей панели «Вставка». С её помощью помимо стандартных объектов можно добавлять к презентации различные мультимедийные средства: от обычных рисунков до видео и звука.

Панель «Дизайн» позволяет задавать слайдам различные темы оформления с настройкой цветов, шрифтов, эффектов и стилей фона.

С помощью панели «Переходы» можно настроить для каждого слайда свой способ воспроизведения в режиме показа. Для каждого вида анимации, предлагаемого Microsoft, доступны индивидуальные настройки, вызываемые кнопкой «Параметры эффектов» справа от списка. В этой панели настраивается звук и командно-временные параметры смены слайдов презентации.

С помощью панели «Анимация» также настраивается воспроизведение слайдов, но уже применительно к конкретным объектам слайда. Здесь также доступна команда «Параметры эффектов» аналогично панели «Переходы». Кроме этого, командно-временные параметры анимации – способ старта, длительность и задержку анимации можно настроить с помощью триггера. При выборе анимации справа от области слайда в окне «Область анимации» появляется значок, кликнув по которому правой кнопкой, можно вызвать все необходимые параметры.

Панель «Показ слайдов» служит для настройки показа презентации. С её помощью можно определить, с какого слайда начинать показ, скрывать тот или иной слайд, настраивать управление показом слайдов,

На панели «Вид» расположены кнопки режима, которые позволяют быстро переключиться в один из режимов просмотра Power Point (Обычный режим, Режим сортировщика слайдов, Страницы заметок, Режим чтения). Для эффективного применения PowerPoint при создании и редактировании презентаций необходимо использовать различные режимы просмотра документов. Режимы представляют собой разные способы отображения слайдов на экране. К основным режимам, применяемым в PowerPoint, относятся: обычный режим и режим сортировщика слайдов.

Режимы отображения слайдов:

1. Режим «Обычный». В этом режиме в окне приложения отображаются три области: Структура/Слайды; область Слайда; Заметки к слайду. Размеры областей можно изменять, перетаскивая их границы.

2. Режим «Сортировщик слайдов» – это режим, в котором все слайды презентации отображаются виде миниатюр. В этом режиме можно легко перемещать слайды, изменяя порядок их следования в презентации.

3. Режим «Чтение» - это режим, с помощью которого можно просмотреть презентацию на экране.

4. Режим «Страницы заметок» – режим просмотра, в котором к каждому из слайдов можно добавить заметки докладчика. В верхней половине страницы появляется уменьшенное изображение слайда, а в нижней половине отображается большая панель для текста заметок.

Кроме режимов просмотра презентации доступна группа режимов образцов слайдов, образцов выдач и образцов заметок, позволяющие дополнительно открыть панели конструкторов.

Создание и форматирование презентации

Презентацию можно создать несколькими способами. Можно создать презентацию на основе шаблона, определяющего дизайн (но не содержание) презентации. Шаблоны можно найти через меню «Файл» или по кнопке Office в левом верхнем углу, далее выбрав команду «Создать». Также можно открыть имеющуюся презентацию и на ее базе создать новую презентацию.

Кроме того, можно создать новую презентацию без разметки, т.е. презентацию на базе пустых слайдов или применить разметку к пустым слайдам (макеты текста, макеты содержимого или макеты текста и содержимого). На пустые слайды можно самостоятельно вводить текст, рисунки, таблицы, диаграммы, звуки и т.д., а также самостоятельно разработать дизайн слайда. Этот способ используется, когда пользователь ясно представляет себе, как будет выглядеть презентация.

На слайд можно добавлять текст четырьмя способами:

- ввести текст в рамку (в поле с пунктирными границами на слайдах) вместо текстового заполнителя;
- добавить на слайд Автофигуру, а в нее ввести текст;
- добавить на слайд объект Надпись, а в нее ввести текст;
- добавить объект WordArt.

Такие макеты слайдов как макеты текста, макеты текста и содержимого содержат рамки для текста. В соответствующие рамки вводится текст заголовков, подзаголовков, списков и основной текст. Для ввода текста в рамку необходимо щелкнуть на ней левой клавишей мыши и набрать текст с клавиатуры или вставить текст с буфера обмена, если он был скопирован из другого документа.

Необходимо отметить, что эти рамки можно перемещать и изменять их размеры. Если текст не помещается в рамке, то необходимо изменить ее размеры или изменить размер шрифта, можно также создать новый слайд и переместите текст на него. В области Структура отображается только текст, введенный в рамки. Текст в объектах надпись или автофигура, а также текст WordArt не отображается в области Структура, поэтому его можно редактировать только на слайде. Объекты «Автофигура», «Надпись» и WordArt расположены на вкладке «Вставка».

Форматирование текста осуществляется теми же способами, что и в программе Microsoft Word.

К форматированию слайда относится операция изменения шаблона оформления или дизайна слайда. Шаблон оформления можно применить не только в момент создания презентации, но и после ее создания.

Для этого необходимо перейти на вкладку «Дизайн» и выбрать тему.

В результате этой операции изменится дизайн всех слайдов презентации, но можно изменить дизайн одного или нескольких выделенных слайдов, если из раскрывающегося списка на шаблоне оформления выбрать команду «Применить к выделенным слайдам».

Для изменения цветовой схемы (цвета и интенсивности) слайда необходимо в области задач на панели Дизайн слайда в разделе Цвета щелкнуть на требуемом эскизе цветовой схемы.

С помощью команды «Стили фона» можно изменить только фон слайдов презентации (не цветовую схему), а также разработать собственный дизайн фона слайдов презентации.

Для изменения стандартного форматирования текста на слайдах необходимо эти изменения внести в образец слайдов. Образец слайдов обычно форматируется в следующих случаях:

- 1) При изменении шрифтов и маркеров.

2) Для вставки картинок, которые должны появиться на слайдах презентации.

3) Изменение расположения, размера и формата рамок.

Известно, что все слайды презентации основываются на образце слайдов и образце заголовков. Для форматирования образца слайдов и образца заголовков необходимо перейти в режим образца на вкладке «Вид».

Вставка, удаление и перемещение слайдов выполняется стандартными способами. Команда создания нового слайда находится на вкладке «Главная».

Расширенные возможности PowerPoint

Панель «Вставка» предоставляет широкие возможности добавления демонстрационного материала: таблицы, содержимое графических файлов, снимок экрана, создание презентации на основе набора рисунков, автофигуры, стандартные графические объекты SmartArt, диаграммы, а также видео и звук.

Команда Видео – Видео из файла позволяет добавить в презентацию видеоклип из файла или вебсайта. С помощью команд группы «Звук» можно добавлять готовые звукозаписи, добавлять звук из коллекции программного пакета и записывать специальные аудиозаписи посредством микрофона.

Анимация – это добавление к тексту или объекту специального видео- или звукового эффекта. Эффекты анимации могут применяться к таким элементам на слайде как: текст, рисунки, графики, диаграммы и других объектов. Эффекты анимации текста, как правило, можно применить к буквам, словам и абзацам.

В PowerPoint для добавления анимации применяются следующие команды: эффекты анимации и настройка анимации. Готовые эффекты анимации могут применяться ко всем элементам выделенных слайдов или всех слайдов презентации. Настройка анимации может применяться к отдельным элементам на слайде.

Для настройки анимации используется соответствующая вкладка. Чтобы выбрать вид анимации, нужно выделить объект, к которому она будет применена. Для каждого вида анимации можно настраивать эффекты. Полный список видов анимации, разделённый на группы по выполняемому действию, можно увидеть

в раскрывающемся списке «Добавить анимацию». Кроме эффектов можно настраивать временные параметры анимации: длительность и задержку. Данные параметры будут использованы непосредственно при демонстрации.

Задание на работу: разработать презентацию своей выпускной квалификационной работы бакалавра. Презентация должна содержать не более 15 слайдов, представляющих основные положения работы и демонстрационные материалы исследований при их наличии. Презентация должна позволять представить работу в течение 10-15 минут.

Контрольные вопросы

1. Какие свойства слайда влияют на его отображение во время демонстрации?
2. Какими способами можно создать презентацию?
3. Какие вы знаете режимы отображения слайдов?
4. Что такое анимация?
5. Что означает режим «Страницы заметок»?

Использование математического пакета MathCad

Цель работы: закрепить навыки работы с математическим пакетом MathCad для решения инженерных задач.

Пользовательский интерфейс среды MathCad

Вначале рассмотрим некоторые опции меню. Опции **File** (файл), **Edit** (редактирование) типичны для всех приложений Windows, поэтому их рассматривать не будем.

Пункт **View** (вид) содержит ряд пунктов, первый из которых - **toolbars** (панели инструментов). Раскроем его (Рис. 1) и рассмотрим некоторые его подпункты. Обращение к первому - **Standart** вызывает на экран стандартную панель. Эта панель во многом идентична соответствующим панелям других приложений Windows, Однако, там имеется опция **f(x)**, с помощью которой вызываются встроенные функции Маткада. Мы рассмотрим ее несколько позднее.

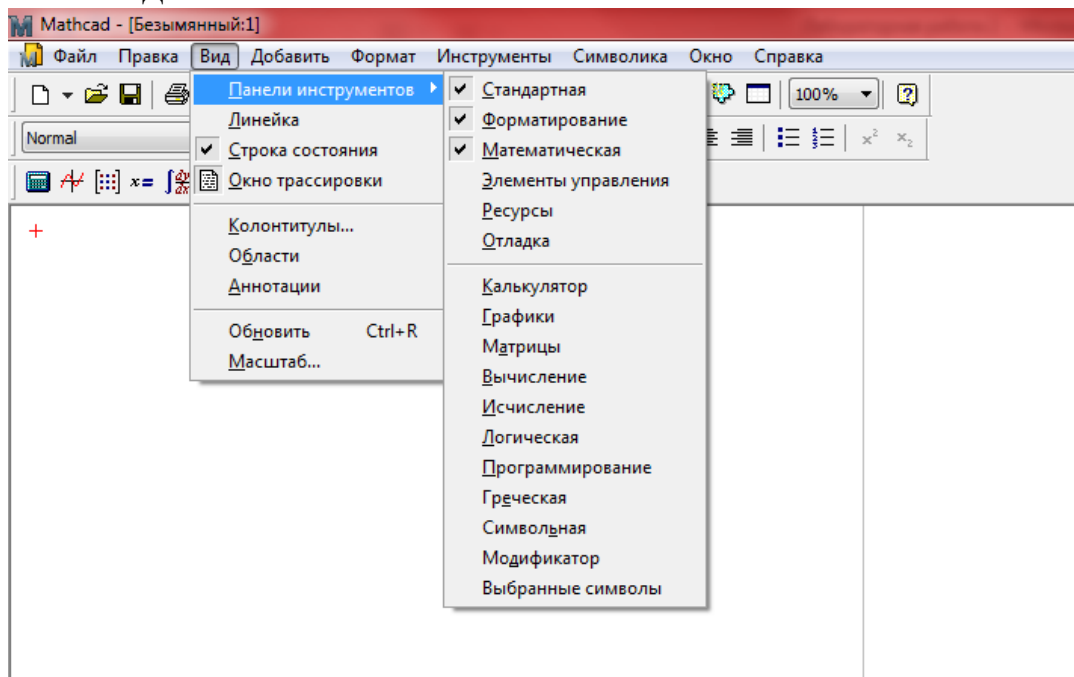


Рис. 1

Кратко рассмотрим некоторые другие пункты. **Formatting** (форматирование) аналогичен соответствующим пунктам меню других приложений. Строкой **Math** (математика) вызывается математическая панель. Она содержит кнопки, дублирующие следующие пункты меню: **Calculator** (калькулятор), **Graph** (графика), **Matrix** (матрицы), **Calculus** (вычисления), **Boolean** (Булевы функции), **Greek** (Греческий), **Programming** (Программирование), **Symbolic** (Символьные вычисления). Пункт

Graph (графика) вызывает панель Графики, пункт **Matrix** (матрицы) обеспечивает операции над матрицами, с помощью пункта **Calculus** (вычисления) производятся дифференцирование, интегрирование, суммирование и произведение, пункт **Boolean** (Булевы функции) вводит булевы функции, пункт **Greek** (Греческий) вводит греческий алфавит, пункт **Programming** (Программирование) вызывает панель программирования, пункт **Symbolic** (Символьные вычисления) делает возможным решение некоторых задач в символьном (аналитическом) виде.

Остальные пункты меню View мы не рассматриваем. Мы не рассматриваем также и другие меню.

Алфавит системы Маткад содержит: строчные и прописные латинские и греческие буквы; арабские цифры от 0 до 9; системные переменные; операторы; имена встроенных функций; спецзнаки; строчные и прописные буквы кириллицы (при работе с русифицированными документами); укрупненные элементы языка: типы данных, операторы, функции пользователя и управляющие структуры. К типам данных относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Числовые константы задаются с помощью арабских цифр, десятичной точки (а не запятой) и знака - (минус). Например: 123 - целочисленная десятичная константа; 12.3 - десятичная константа с дробной частью; $12.3 * 10^5$ - десятичная константа с мантиссой (12.3) и порядком 5.

Знак умножения * при выводе числа на экран меняется на привычную математикам точку, а операция возведения в степень (с применением спецзнака ^) отображается путем представления порядка в виде надстрочного элемента. Диапазон возможных значений десятичных чисел лежит в пределах от 10^{-307} до 10^{307} (это машинный ноль и машинная бесконечность).

Большинство вычислений система выполняет как с действительными, так и с комплексными числами, которые обычно представляются в алгебраическом виде:

$$Z = \text{Re}Z + i * \text{Im}Z$$

или

$$Z = \text{Re}Z + j * \text{Im}Z$$

Здесь $\text{Re}Z$ - действительная часть комплексного числа Z , $\text{Im}Z$ - его мнимая часть, а символы i или j обозначают мнимую единицу, т. е. корень квадратный из -1 . Такое представление характерно и для системы Маткад (за исключением того, что знак равенства не есть знак присваивания). Итак, если $\text{Re}Z = 2$, а $\text{Im}Z = 3$, то комплексная числовая константа в системе Маткад должна быть задана в виде $2 + i * 3$ или $2 + j * 3$.

Однако система не всегда знает, какой символ применить для обозначения мнимой единицы. Поэтому, перед использованием любых операций с комплексными числами, полезно вначале определить i или j как мнимую единицу (т. е. присвоить им значение квадратного корня из -1).

Имена **переменных (идентификаторы)** в системе Маткад могут иметь практически любую длину, и в них могут входить любые латинские и греческие буквы, а также цифры. Однако начинаться они могут только с буквы, например: x , $x1$, α , X , $coordinate$. Кроме того, идентификатор не должен содержать пробелов. **Строчные и прописные буквы в именах различаются!** Имя не должно совпадать с именами встроенных функций.

Для набора $+$ и $-$ используются соответствующие клавиши клавиатуры. Деление набирается клавишей $/$.

После входа в Маткад на экране появляется красный крестик, который обозначает место, где будет производиться запись. Его можно перемещать по экрану мышью.

При вводе любого символа на месте крестика появляется рамка - шаблон. **Любое введенное выражение (формула, уравнение и т.п.) должно записываться внутри одного шаблона.** При переходе к следующей записи первый шаблон исчезает. Если подвести курсор к записи без шаблона и нажать на клавишу мыши, шаблон появится снова.

Удаление записей в Маткаде можно производить несколькими способами:

1. Для удаления одного или нескольких выражений одновременно можно, нажав левую кнопку мыши, обвести все удаляемые выражения пунктиром, и затем нажать **del** или **backspace**.

2. Для удаления одного выражения можно, подведя курсор к выражению, активизировать его и, передвинув уголок в крайнее правое выражение, нажать **del**.

3. Для удаления одного выражения можно также, подведя к нему курсор, активизировать его, а затем выделить нужный текст и нажать **del**.

Одним из многих достоинств Маткад является легкость построения графиков.

Панель графиков вызывается нажатием кнопки с изображением графиков на математической панели (рис. 2).

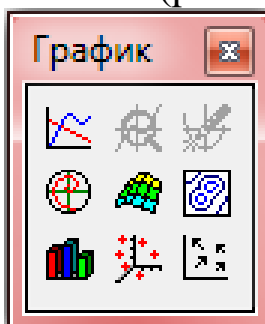


Рис. 2

На панели графиков расположены девять кнопок с изображением различных типов графиков (название графиков каждой кнопки высвечивается при подводе к ней курсора и ожидания в течение 3-5 секунд): **X-Y Plot** - графики в декартовых координатах, **Polar Plot** - графики в полярных координатах, **3D Bar Chart** - столбиковые диаграммы, **Surface Plot** - трехмерный график, **Contour Plot** - карта линий уровня (изолиний), **Vector Field Plot** - векторное поле, **3D Scatter Plot** - трехмерный точечный график. Для примера рассмотрим шаблон графика в декартовой системе координат (X-Y график).

При нажатии данной кнопки в рабочем поле появляется пустая декартова система координат (рис. 3). Пользователем вводятся необходимые параметры: переменная-аргумент, переменная-функция и числовые интервалы обоих переменных, которые должны быть отражены на графике. Для аргумента должны быть определены численные значения, переменная-функция должна быть определена как функция от переменной-аргумента. Пример построения параболы показан на рис. 4.

Вызвав правой кнопкой мыши контекстное меню графика, можно перейти к форматированию графика по команде «Формат».

Здесь можно настроить отображение графика: назначить параметры осей координат, самого графика и подписей, добавить сетку, автомасштабирование и т.д.

Большинство задач в MathCAD решаются с помощью так называемых встроенных функций, т.е. с помощью заранее составленных программ решения той или иной задачи.

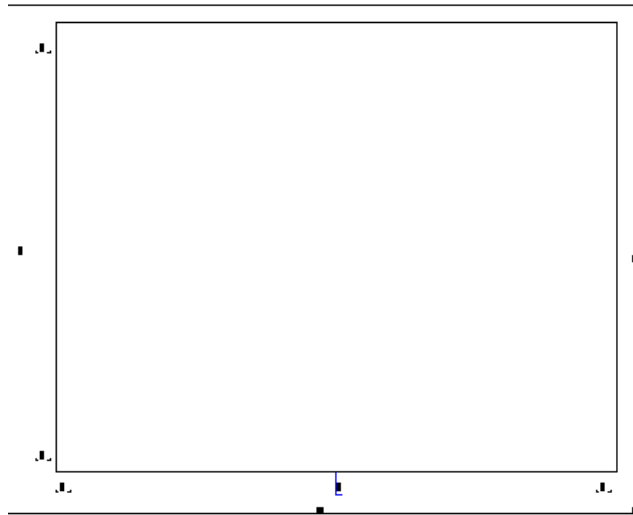


Рис. 3

$$x := -100..100$$

$$f(x) := x^2$$

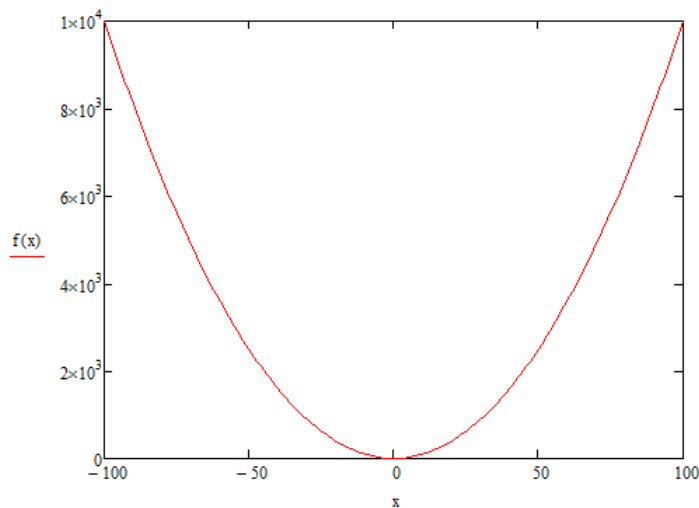


Рис. 4

Каждая такая программа имеет свое имя, по которому она вызывается. Имя можно набирать с клавиатуры, но можно и вызывать из меню Insert (Добавить) – Function (Функцию). Нажав эту кнопку, мы вызовем панель **Insert Function** (вставить функцию), в которой имеется два поля: **Function Category** (тип функции) и

Function Name (имя функции). Выбрав тип и имя, и нажав кнопку ОК, мы вызовем данную функцию.

Массивы могут записываться в виде векторов (одномерные массивы), в виде матриц (двумерные массивы) и в виде таблиц. Мы рассмотрим пока только запись матриц и векторов. Для действий над ними имеется панель Matrix (матрица), показанная на рис. 5.

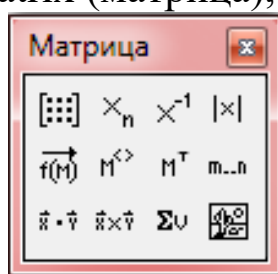


Рис. 5

При вызове кнопки «Матрица или вектор» появляется окно определения количества строк и столбцов. Над векторами определены операции сложения и вычитания, транспонирования, умножения по математическим правилам умножения матриц. Знак транспонирования следует вводить с панели Matrix (матрица).

Порядковый номер элемента, который является его адресом, называется индексом.

Нижняя граница индексации задается значением системной переменной ORIGIN, которая может принимать значение 0 или 1.

Так как MathCAD позволяет определять значения переменных в векторной или матричной форме, это позволяет строить графики по таблицам значений координат точек.

Системы компьютерной алгебры снабжаются специальным процессором для выполнения аналитических (символьных) вычислений. Его основой является ядро, хранящее всю совокупность формул и формульных преобразований, с помощью которых производятся аналитические вычисления. Чем больше этих формул в ядре, тем надежней работа символьного процессора и тем вероятнее, что поставленная задача будет решена, разумеется, если такое решение существует в принципе (что бывает далеко не всегда).

Ядро символьного процессора Mathcad - несколько упрощенный вариант ядра известной системы символьной математики Maple V фирмы Waterloo Maple Software, у которой

MathSoft (разработчик Mathcad) приобрела лицензию на его применение, благодаря чему Mathcad стал системой символьной математики.

Введение в систему Mathcad символьных вычислений придает ей качественно новые возможности. Символьные вычисления выполняются, в конечном счете, столь же просто для пользователя, как, скажем, вычисление квадрата x .

Операции, относящиеся к работе символьного процессора, содержатся в подменю позиции **Symbolic** (Символика) главного меню.

Чтобы символьные операции выполнялись, процессору необходимо указать, над каким выражением эти операции должны производиться, т.е. надо выделить выражение. Для ряда операций следует не только указать выражение, к которому они относятся, но и наметить переменную, относительно которой выполняется та или иная символьная операция.

Само выражение в таком случае не выделяется, ведь и так ясно, что если маркер ввода выделяет переменную какого-либо выражения, то это выражение уже отмечено наличием в нем выделяемой переменной.

Символьные операции разбиты на пять характерных разделов. Это операции с выражениями, операции с переменными, операции с матрицами, операции преобразования, стиль эволюции. Первыми идут наиболее часто используемые операции. Они могут выполняться с выражениями, содержащими комплексные числа или имеющими решения в комплексном виде.

Символьный процессор системы Mathcad обеспечивает проведение в символьном виде трех наиболее распространенных матричных операций транспонирования и обращения матриц, а также вычисления их детерминанта.

При символьных вычислениях, прежде всего, следует вызвать панель символьных вычислений нажатием кнопки на математической панели.

Задания на лабораторную работу

Вычисление производных в задачах геометрии и нахождение частных производных высоких порядков

I Составить уравнение касательной и нормали к линии, которая задана уравнением $y(x)=f(x)$ в точке $M(x_0, y_0)$.

1. Задать значения x_0 и y_0 в точке M .

2. Записать уравнение линии $y(x)$.

3. Определить производную от функции $y(x)$ $\frac{d}{dx} y(x) \rightarrow$, используя панель вычислений и панель символов. Присвоить значение производной функции $yy(x) := \frac{d}{dx} y(x)$.

4. Записать уравнение касательной в виде

$$\text{tang}(x) := yy(x_0) \cdot (x - x_0) + y_0,$$

5. Аналогично записать уравнение нормали

$$\text{norm}(x) := \frac{-1}{yy(x_0)} \cdot (x - x_0) + y_0$$

6. Построить графики касательной и нормали.

7. Отформатировать графики.

II. Выполнить числовое и символьное вычисление частных производных высшего порядка от функции трех переменных.

1. Записать функцию, от которой будут вычисляться производные второго порядка.

2. Обратиться к панели вычислений и выбрать оператор дифференцирования.

3. В соответствующем месте заполнения оператора записать функцию, переменную для дифференцирования и порядок дифференцирования.

4. Нажать правой кнопкой мыши на знак оператора дифференцирования и в контекстном меню выбрать View Derivative As (Показать производную как), установить флажок Partial Derivative (Частная производная).

5. Отметить оператор дифференцирования и обратиться к панели Символика/Вычислить/В символах.

6. Задать числовые значения для переменных, от которых вычисляется производная.

7. Вычислить числовые значения производных.

Варианты заданий

Номер вариант а	Функция f(x) для определения касательной и нормали	Точка М (x ₀ ,y ₀) для определения касательной и нормали	Функция f(x,y,z) для вычисления частной производной	Точка М (x ₀ ,y ₀ ,z ₀) для числового вычисления частной производной
1	$x^2 - 3x + 5$	(2,3)	$x^2 - 3x^3y - 4y^2 + 2y - z^3$	(0,1,2)
2	$x^2 + 2x + 6$	(-1,1)	$z^2 e^{x^*x+y*y}$	(0,0,0)
3	$x^3 - 3x^2$	(3,1)	$x \cos(y) + yz^4$	(1,0,0)
4	$0.5x - \sin(x)$	(0, $\pi/3$)	$z \ln(x^2 - y^2)$	(3,1,3)
5	$(x-5)e^x$	(4,0)	$z \sin(xy) + z^2$	(1,1,1)
6	$1 - (x-2)^{4/5}$	(2,1)	$x^2 + 2y^2 - 3xy - 4z^2$	(0,0,0)
7	$x^5 + 5x - 6$	(0,-1)	$zx \cdot \ln(y) + xy^2z$	(0,2,1)
8	$(x^3 + 4)/x^2$	(2,3)	$y(x - z \cos(x))$	(0,0,0)
9	$\sqrt[3]{1-x^3}$	(0,1)	$\sin(x)(\cos(z) + \cos(y))$	(1,0,0)
10	$\sin^2(x)$	(0.5,0.5)	$x^4yz + \sin(y)$	(2,1,0)
11	$x^2 - 0.5x^4$	(0,0)	$(x-y^2)*(z^3-x)$	(1,1,1)
12	$x^3 - 3x^2$	(0, $\pi/3$)	$x^2 - 3x^3y - 4y^2 + 2y - z^3$	(0,1,2)

Вычисление интегралов в задачах геометрии и механики

I. Вычислить площадь плоской фигуры, ограниченной заданными линиями.

1. Записать уравнение кривых, которые ограничивают площадь плоской фигуры.

2. Найти точки их пересечения, для того чтобы использовать их в двукратном интегрировании.

3. Обратиться на панели Символы к функции *simplify*.

4. Ввести оператор интегрирования. В соответствующих местах заполнить имя первой переменной и границы интегрирования.

5. На месте ввода функции под интегралом ввести еще один оператор интегрирования, границы интегрирования и подынтегральную функцию

$$S = \iint_D dx dy.$$

II. Вычислить координаты центра тяжести пластины.

1. Записать уравнения кривых, которые описывают область D пластины.

2. Найти точки их пересечения, для того чтобы использовать их в двукратном интегрировании.

3. Найти площадь S однородной пластинки через двойной интеграл.

3.1. Обратиться на панели Символы к функции *simplify*.

3.2. Ввести оператор интегрирования. В соответствующих местах заполнить имя первой переменной и границы интегрирования.

3.3. На месте ввода функции под интегралом ввести еще один оператор интегрирования, границы интегрирования и подынтегральную функцию

$$S = \iint_D dx dy.$$

4. Найти аналогично статические моменты M_x и M_y пластины относительно осей Ox и Oy как двойные интегралы

$$M_x = \iint_D y dx dy, \quad M_y = \iint_D x dx dy.$$

5. Определить координаты центра тяжести как отношение подынтегральной функции, которая определяет статические моменты пластины относительно осей Ox и Oy

$$x = \frac{M_y}{S}, \quad y = \frac{M_x}{S}.$$

Варианты заданий

Номер варианта	Функции для вычисления площади фигуры	Функции для вычисления координат центра тяжести фигуры
1	$x=y^2-2y; x+y=0$	$\frac{x^2}{25} + \frac{y^2}{9} = 1; \frac{x}{5} + \frac{y}{3} = 1$
2	$y=2-x; y^2=4x+4$	$y=x^2; y=2x^2; x=1; x=2$

3	$y^2=4x-4; y^2=2x$ (извне параболы)	$y^2=x; x^2=y$
4	$3y^2=25x; 5x^2=9y$	$y=\sqrt{2x-x^2}; y=0$
5	$y^2+2y-3x+1=0; 3x-3y-7=0$	$\frac{x^2}{16}+\frac{y^2}{9}=1; \frac{x}{4}+\frac{y}{3}=1$
6	$y=4x-4x^2; y=x^2-5x$	$x^2+y^2=1; x+y=1$
7	$x=4-y^2; x+2y-4=0$	$x^2+y^2=4; x+y=2$
8	$y^2=4(x-1); x^2+y^2=4$ (извне параболы)	$\frac{x^2}{16}+\frac{y^2}{4}=1; \frac{x}{4}+\frac{y}{2}=1$
9	$x=y^2-2y; x+y=0$	$\frac{x^2}{16}+\frac{y^2}{9}=1; \frac{x}{4}+\frac{y}{3}=1$
10	$y=2-x; y^2=4x+4$	$x^2+y^2=1; x+y=1$
11	$y^2+2y-3x+1=0; 3x-3y-7=0$	$x^2+y^2=4; x+y=2$
12	$y=4x-4x^2; y=x^2-5x$	$y^2=x; x^2=y$

Интерполяция экспериментальных данных в MathCad

1. Набрать таблицу, которая соответствует варианту.
2. Осуществить линейную интерполяцию, для чего необходимо выполнить следующие действия:
 - 2.1. Ввести векторы данных x и y .
 - 2.2. Определить функцию $linterp(x, y, t)$.
 - 2.3. Вычислить значения этой функции в точках, которые выбрать самостоятельно.
3. Построить график функции.
4. Осуществить сплайн-интерполяцию, используя функцию $interp(s, x, y, t)$, для чего необходимо выполнить следующие действия:
 - 4.1. Ввести векторы данных x и y .
 - 4.2. Ввести функцию $cspline(x, y)$, которая определяет первый аргумент функции $interp(s, x, y, t)$, как векторную величину значений коэффициентов кубического сплайна.
 - 4.3. Определить функцию $interp(s, x, y, t)$.
 - 4.4. Вычислить значения этой функции в точках, которые задать такими же, как и для линейной интерполяции.
5. Построить график функции.

6. Выполнить сравнительный анализ полученных разными подходами интерполяционных графиков и значений функции в одинаковых точках.

Варианты заданий

Номер вариант а	Аргумент ы и значения	Данные						
		3						
1	х	1	2	3	4	5	6	7
	у	35,6	38, 7	39,4	40,8	43,3	42,9	41,8
2	х	1	2	3	4	5	6	7
	у	135,2	138, ,7	139, 9	141, 6	140, 1	142, 5	141, 8
3	х	1	2	3	4	5	6	7
	у	9,7	10, 3	10,8	10,2	11,9	11,4	11,4
4	х	1	2	3	4	5	6	7
	у	14,5	16, 2	16,5	17,2	19,8	17,7	17,5
5	х	1	2	3	4	5	6	7
	у	32,8	30, 2	21,7	27,8	27,5	27,2	27,9
6	х	1	2	3	4	5	6	7
	у	36,3	38, 5	39,7	39,1	39,0	38,7	40,0
7	х	1	2	3	4	5	6	7
	у	52,7	56, 5	60,7	54,8	70,4	68,1	67,8
8	х	1	2	3	4	5	6	7
	у	11,1 2	10, 6	11,3 1	11,0 2	12,0	12,7 3	11,1 2
9	х	1	2	3	4	5	6	7
	У	1,8	2,9	2,0	3,6	3,8	3,9	4,1
10	х	1	2	3	4	5	6	7
	у	9,8	10, 1	10,3	11,9	10,9	11,8	12,1

11	x	1	2	3	4	5	6	7
	y	4,7	4,6	4,6	5,3	5,3	5,5	5,6
12	x	1	2	3	4	5	6	7
	y	2,12	1,2 8	1,71	1,6	1,11	1,18	1,02

Содержание отчёта

1. Титульный лист.
2. Цель работы.
3. Задания на лабораторную работу.
4. Содержания рабочих листов из MathCad.
5. Выводы по теме работы.

Контрольные вопросы

1. Какие основные элементы размещены на экране рабочего окна?
2. Как создать новый документ Mathcad?
3. Каким образом в поле рабочего окна открыть панель Математика?
4. Что входит в состав алфавита входного языка?
5. Какие форматы представления чисел используются в пакете Mathcad?
6. Какие числовые константы имеет пакет Mathcad?
7. Как образуются имена переменных?
8. Что такое встроенная функция?
9. Что такое оператор присваивания и как его вставить в документ?
10. Какое назначение имеет в Mathcad символ = ?
11. Что такое дискретная переменная и как ее задать?
12. Как в Mathcad задать функцию пользователя?
13. Какова последовательность действий для получения таблицы значений функции?
14. Как изменить формат результата?
15. Где расположен шаблон матрицы?

Проектирование интернет-сайта

Цель работы: закрепить навыки разработки web-сайтов.

Принципы разработки сайтов

Чтобы обратить на себя внимание посетителя, у сайта есть всего несколько секунд. Удержит ли пользователя главная страница сайта? В обществе, насыщенном информацией и многочисленными сайтами разной направленности, главная страница сайта должна привлечь внимание посетителей и ответить на вопросы:

- где?
- что?
- зачем?

Обычно основное внимание уделяют дизайну страницы. В браузерах мы видим только одну страницу за один раз. Сам сайт явно не представлен на экране.

Разработка архитектуры сайта сложная задача и имеет значение более важное, чем разработка дизайна страницы.

Обычно попадая на страницу, пользователь уже спустя короткое время понимает, что там можно сделать. Но перенести пользователя на нужную страницу не так просто.

Главная страница сайта

Главная страница – это флагман сайта, и ее дизайн должен отличаться от дизайна всех остальных страниц. Конечно, первая и остальные страницы должны быть выдержаны в одном стиле, но есть и различия.

Например:

- на главной странице не должно быть кнопки “На главную”, так как не очень-то приятно нажимать на кнопку для того, чтобы попасть на ту же самую страницу;

- на главной странице обычно представлен более крупный логотип компании, ее название и название сайта.

Первая непосредственная цель любой главной страницы ответить на вопросы: “Где я нахожусь?” и “Что делает этот сайт?”

Оба они требуют прямого указания полного имени. Лучше, если и по дизайну будет понятно, какой цели может служить этот сайт

для нового пользователя. Для нового пользователя, вероятно, самая важная функция первой страницы – это ответ на вопрос “Что же этот сайт делает?”, а для большинства остальных первая страница – отправная точка для навигации по сайту.

Первая страница – это также место для представления новостей или специальных предложений, которыми хозяин хочет привлечь внимание всех посетителей. Возможность поиска необходима на первых страницах, так как многие пользователи не хотят бесконечно ходить по ссылкам, чтобы попасть туда, куда им нужно.

Итак, главная страница должна предоставлять три следующие возможности:

- каталог основных содержательных разделов сайта (возможность навигации);
- краткую сводку основных новостей или специальных предложений;
- возможность поиска.

Если все это хорошо сделано, то каталог и новости помогут пользователю, пришедшему на сайт в первый раз, понять, о чем же, собственно говоря, этот сайт рассказывает. Поэтому на главную страницу всегда следует смотреть именно под таким углом: она должна отвечать на вопрос: “Что этот сайт может для меня сделать?”. А также должно быть название компании и логотип по возможности.

Элементы web-страницы

Основные элементы web-страницы:

1. Заголовок/логотип (шапка).
2. Поиск.
3. *Рекламный баннер.*
4. Контент /содержание (текстовое поле).
5. Элементы навигации.
6. Информация о разработчиках сайта.
7. *Счетчик посещаемости.*

Любая web-страница содержит определенный набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета. Безусловно, ассортимент и количество

подобных объектов могут варьироваться в зависимости от тематической направленности сайта, объема опубликованных на нем материалов, а также от целей и задач, которые ставит перед собой создатель данного ресурса. Компоновка таких элементов, проектирование их взаимного расположения и составляет одну из главных задач web-мастера.

Первым элементом web-страницы, который нам предстоит рассмотреть, является ее **Заголовок**. Он может быть выполнен как в текстовом, так и в графическом варианте, однако и в том и в другом случае он должен располагаться в верхней части документа.

Логотип или название сайта выполняет для него ту же роль, что и вывеска на входе в какое-либо учреждение. Логотип сайта обычно находится в верхней части страницы, в верхнем левом или правом углу.

Это объясняется тем, что логотип представляет весь сайт в целом, и, следовательно, он занимает самую верхнюю позицию в логической иерархии сайта. В визуальной иерархии каждой страницы можно сохранить эту доминирующую позицию двумя способами: либо сделать логотип самым заметным объектом на странице, либо поместить его так, чтобы он выполнял роль рамки для содержания страницы.

Иногда с заголовком совмещают меню выбора кодировки кириллицы и кнопки для перехода с русскоязычной на англоязычную версию сайта, если данный web-ресурс представлен на двух языках и **навигацию**.

Непосредственно под/над заголовком документа, как правило, располагается пространство, отведенное для размещения рекламного **баннера**. Включение баннера именно в верхнюю часть web-страницы в большинстве случаев является обязательным условием регистрации сайта в службах баннерного обмена – системах, рекламирующих созданный ресурс в обмен на показ на страницах сайта рекламы других участников баннерообменной сети. В некоторых случаях вместо заимствованного баннера включаются собственные баннеры промоакций.

Основную часть документа занимает так называемое **текстовое поле** - участок, где и размещается смысловое наполнение страницы: содержательный информационный текст и иллюстрации.

Перечисленные элементы еще называют "Контент". Расположение текстового поля зависит в первую очередь от того, каким образом web-дизайнер разместит остальные элементы документа.

Следующей обязательной составляющей частью web-страницы являются **элементы навигации** – гиперссылки, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, например Java-апплетов. Последние представляют собой те же кнопки, которые умеют реагировать на движения мыши, выполняя при наведении на них курсора какие-либо несложные действия (включение подсветки, создание эффекта "нажатия", изменение формы и т. д.). Располагать элементы навигации следует таким образом, чтобы они всегда были "на виду", то есть так, чтобы пользователю не приходилось "отматывать" страницу назад, если текстовое поле занимает по высоте несколько физических экранов, после чего подолгу искать ссылки на другие разделы.

Наиболее устоявшимся подходом является размещение элементов навигации у левой границы страницы и/или верхней.

В нижней части документа принято публиковать **информацию о разработчиках сайта и адрес электронной почты**, по которому посетители ресурса могут направить владельцам странички свои отклики, предложения и пожелания.

Если web-страница является стартовым документом, в нижней ее части также размещают **счетчик посещений** – небольшой сценарий, вызывающий установленный на сервере CGI-скрипт, который фиксирует каждое открытие документа в браузере пользователей, изменяя значение индикатора счетчика. Благодаря этому web-мастер без труда определит количество посетителей, навестивших его страничку в течение какого-либо времени. Счетчик посещений устанавливается только на первой странице, вызываемой при обращении к сайту, в остальных документах ресурса он отсутствует. Не рекомендуется также размещать на одной странице несколько разных счетчиков.

Решение по компоновке страницы остаётся за web-мастером. Пример компоновки страницы показан на рис. 1

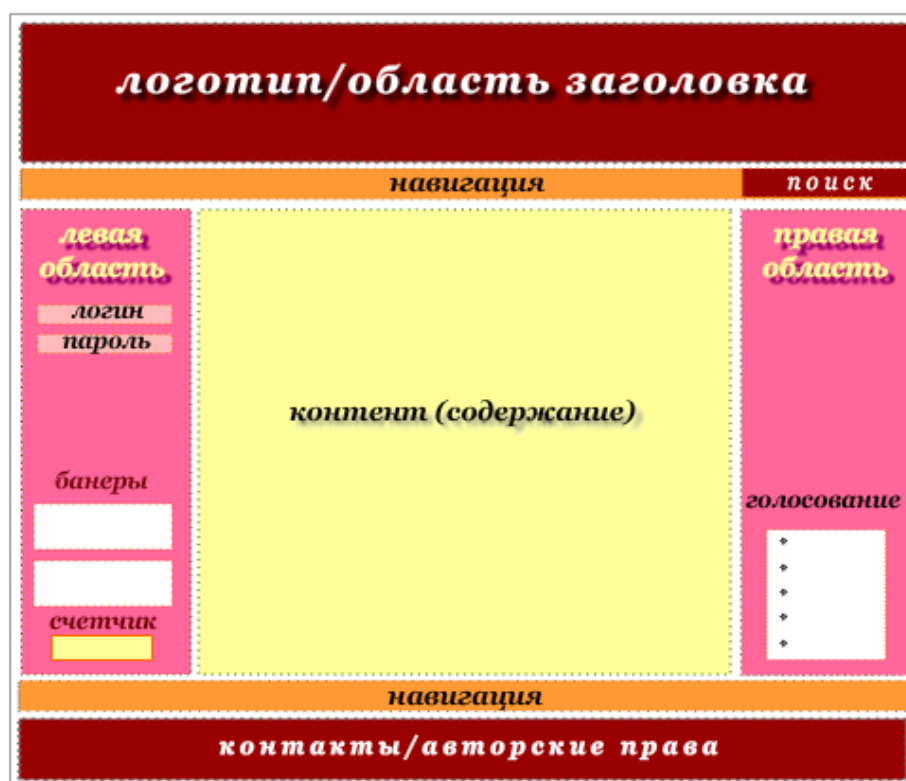


Рис. 1

Очевидно, что вариантов компоновки web-страницы может быть великое множество: конкретные решения зависят от количества составляющих ресурс разделов, объема подготовленного для размещения на сайте текста и, наконец, от фантазии самого дизайнера. Важно лишь, чтобы внешний вид сайта не вызывал нареканий у посетителей. В конце концов, только разработчик вправе проявить все свои способности и таланты и скомпоновать страницу по собственному вкусу. Создатели некоторых домашних страничек, не мучаясь сомнениями, размещают счетчик посещений в верхнем правом углу документа, название сайта пишут мелким убогим шрифтом и публикуют его под рекламным баннером, а элементы навигации почему-то неожиданно обнаруживаются прямо в середине текстового блока, между рассказом о себе и фотографиями любимой собаки автора проекта.

Поэтому необходимо ответственно подходить к проблеме компоновки страниц. В общем случае веб-страница делится на контент страницы, навигацию и сопутствующую информацию. Эти элементы в свою очередь делятся на более мелкие элементы.

Логическая и физическая структура сайта

Каждый ресурс Интернета, от домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой странице, которой присваивается имя `index.htm` (`.html`). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся `index.htm`.

Такое имя файла рекомендуется присваивать всем стартовым документам сайта, поскольку в противном случае при обращении к какому-либо разделу посредством сокращенного URL без указания названия стартовой страницы (например, `http://www.mysite.ru/photos/` вместо `http://www.mysite.ru/photos/startpage.html`) браузер отобразит не саму web-страницу, а перечень хранящихся в данной папке файлов или ошибку.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется **логической структурой сайта**.

Физическая структура подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован сайт.

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволит избежать путаницы при последующем дополнении и обновлении материалов.

Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием "Images", "img" или "pic", расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории.

Для того чтобы все гиперссылки на домашней страничке или web-сайте работали корректно, все документы открывались правильно, и браузер не выдавал ошибок при обращении к каким-либо разделам ресурса, при создании его физической структуры следует соблюдать несколько простых правил.

1. Назначать имена директорий, имена и расширения документов HTML и графических файлов с использованием символов только латинского алфавита и только в строчном регистре.

2. Не рекомендуется давать имена созданных файлов и директорий длиной более 8 символов.

3. При присвоении имен файлов документам HTML необходимо следить за тем, чтобы эти имена были "смысловыми": впоследствии легко можно забыть содержимое и назначение какой-либо web-страницы, если имена файлов будут выглядеть, например, как 1.htm, 2.htm, 3.htm и т. д.

Из всего сказанного становится очевидным, что физическая структура сайта скрыта от посетителей ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации. Отсюда следует вполне логический вывод: строение системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной логической структуре сайта.

В итоге последовательность действий по разработке web-сайта сводится к следующему несложному алгоритму:

1. Постановка целей и определение основных задач.
2. Создание списка будущих тематических разделов.
3. Разработка логической и физической структуры ресурса.
4. Подготовка эскиза дизайна, компоновки сайта, невидимой верстальной таблицы.
5. Подготовка текстовых материалов.
6. Подготовка графических материалов в векторной форме.
7. Экспорт векторных рисунков в растровый формат.
8. Оптимизация всех изображений.
9. Создание шаблонов web-страниц.

10. Сборка web-страниц и отладка кода.
11. Проверка идентичности отображения web-страниц с различным экранным разрешением и цветовой палитрой и различных браузерах.

Названия страниц

Каждая страница должна иметь название в заголовке страницы.

При подготовке информации для Интернета значительное внимание следует уделять вопросам, имеющим отношение к последующему поиску этой информации. Посетители любят пользоваться поисковыми системами как для поиска материалов в пределах текущего узла, так и для поиска в Интернете в целом. Для людей, использующих поисковые системы, тот или иной сайт существует исключительно в виде названия, отображаемого в результатах поиска.

HTML позволяет указывать в разделе заголовка страницы ее название. Названия web-страниц имеют большое значение, поскольку они нередко выступают в виде ссылок на соответствующие страницы. Кроме того, названия страниц отображаются во всевозможных меню навигации в браузерах, таких как списки закладок (избранные сайты) или ранее посещенные сайты.

Во многих из перечисленных ситуаций названия web-страниц рассматриваются в отрыве от основного контекста. Таким образом, необходимо, чтобы названия страниц были достаточно информативными и позволяли правильно судить о характере содержащихся на этих страницах материалов. С другой стороны, слишком длинные названия также неудобны. Название страницы также можно отнести к разряду информационного наполнения web-пространства. Каждое название должно представлять собой образец четкости формулировки. Автор должен объяснить потенциальному посетителю, какую информацию можно найти на соответствующей странице. Каждая из страниц должна иметь уникальное название. Представьте себе ситуацию, когда вы посетили, скажем, семь страниц с

одинаковыми названиями, а затем пытаетесь вернуться к одной из них, используя список посещенных страниц.

И наконец, названия должны быть приспособлены для беглого ознакомления. С этой целью следует помещать наиболее информативные фразы в начало названия, причем предпочтительнее всего начинать название со слова, которое с наибольшей вероятностью будет отражать потребности потенциального посетителя. Весьма часто встречается, когда в качестве названия используются такие фразы, как “Добро пожаловать на узел ХХХ”. Гораздо лучшим вариантом было бы использование в качестве названия просто “ХХХ”. Названия не обязательно должны представлять собой законченные грамматические структуры, скорее они должны быть выдержаны в духе рекламных слоганов.

При создании заголовков в Интернете необходимо руководствоваться приводимыми ниже принципами:

1. Заголовок должен четко описывать содержание соответствующей страницы.

2. Заголовок должен быть написан понятным языком.

3. Первое слово должно быть наиболее информативным, что будет способствовать оптимальному позиционированию в алфавитных списках и облегчит беглое ознакомление. Например, можно начинать заголовки с названия организации, имени автора или понятия, представляющего собой предмет обсуждения.

4. Названия страниц не должны начинаться с одного и того же слова. Такие названия будет сложно различать при беглом просмотре списка. Общие слова лучше всего помещать в конец названий.

Название страницы должно находиться в правильном месте. Другими словами, название страницы должно занимать в ее визуальной иерархии такое место, чтобы оно обрамляло расположенное на ней содержание.

Позиция, размер, цвет и шрифт названия должны ясно показывать, что это заглавие для всей страницы в целом. Поэтому, в большинстве случаев, текст названия страницы будет иметь самый большой размер.

Существует негласное правило о том, что название страницы соответствует тем словам, по которым пользователь щелкнул мышью для того, чтобы на эту страницу попасть.

Удобочитаемость

Если пользователь оказывается не в состоянии прочитать текст, все остальное – дизайн, быстрота загрузки, информационное наполнение – не имеет никакого смысла. Существует ряд основных правил, которых следует придерживаться для обеспечения разборчивости при публикации материалов.

1. Фон и текст страницы должны быть оформлены контрастными цветами. Наилучшая разборчивость обеспечивается при использовании черных символов на белом фоне (так называемого позитивного текста). Также хорошо воспринимается белый текст на черном фоне (негативный текст). Несмотря на то, что уровень контрастности в последнем случае аналогичен позитивному тексту, негативная цветовая схема способствует некоторому рассеянию внимания читателя и очень замедляет чтение. Разборчивость еще больше страдает, если цвет текста несколько светлее черного, в особенности, если при этом фоновый цвет несколько темнее белого. Наиболее **неудобны** для чтения такие цветовые схемы, как розовый цвет на зеленом фоне: эта схема характеризуется слишком низкой контрастностью и, кроме того, не позволяет читать текст пользователям, страдающим нарушением восприятия красного и зеленого цветов.

2. В качестве фона страницы следует использовать либо однотонный цвет, либо узор, имеющий минимальную контрастность. Фоновые изображения затрудняют распознавание текста глазом человека.

3. Текст должен быть набран достаточно крупным шрифтом, чтобы его могли читать даже пользователи с ослабленным зрением.

4. Текст должен быть статичным. Движущийся, мерцающий или изменяющийся в размерах текст воспринимается значительно хуже, нежели статичный.

5. В большинстве случаев текст должен быть выровнен по левому краю или по ширине. Если начало всех строк находится на одном уровне по горизонтали, чтение текста значительно

ускоряется в сравнении с текстом, выровненным по центру или по правому краю. Конечно, выравнивание по правому краю или по центру отдельных небольших фрагментов текста вполне допустимо, однако основной текст должен быть выровнен по левому краю или по ширине.

6. Аналогично, восприятие списков значительно облегчается при выравнивании первого слова каждого из элементов списка по левому краю вдоль одной линии.

7. Поскольку современные мониторы имеют сравнительно низкую разрешающую способность, текст, набранный мелким шрифтом, воспринимается значительно лучше при оформлении его с использованием рубленых шрифтов, таких как Verdana. Для четкого отображения засечек шрифта размером 10 пунктов попросту не хватает пикселей. В то же время, большинство людей предпочитает читать текст, набранный шрифтами с засечками, таким образом, мы оказываемся перед лицом своего рода парадокса. Разборчивость должна быть определяющим критерием в случае очень мелких шрифтов (9 пунктов и менее). Такой текст должен набираться с использованием рубленого шрифта.

8. Не следует набирать текст ПРОПИСНЫМИ БУКВАМИ. Чтение такого текста выполняется примерно на 10% медленнее, чем текста, набранного с использованием строчных букв, поскольку восприятие символов, имеющих одинаковую высоту, затруднено. Этого следует избегать.

Разработка информационного наполнения

В конечном итоге, основной целью посещения любого сайта является получение определенной информации. Все остальное лишь дополнение, и смысл дизайна оказать людям необходимую помощь в получении доступа к этой информации.

Характер текста веб-страницы определяет не только ее содержание, но впечатление, производимое на пользователя, поскольку посетители в первую очередь обращают внимание на текст и заголовки. Отсутствие в тексте грамматических ошибок имеет большое значение, однако также немаловажно представлять информацию в форме, позволяющей акцентировать внимание посетителя на наиболее существенных моментах повествования.

При создании текстов для Сети нужно придерживаться трех основных рекомендаций:

1. Следует соблюдать лаконичность. На странице должно размещаться не более 50% текста, который может быть использован для передачи того же материала в печатном издании.

2. Текст должен быть удобным для беглого ознакомления. Не следует заставлять посетителя читать крупные абзацы текста. Вместо этого лучше использовать небольшие абзацы, подзаголовки и маркированные списки.

3. Информацию значительного объема следует разбивать на несколько страниц, связанных между собой гиперссылками.

Исследования показали, что чтение текста с экрана монитора происходит приблизительно на 25% медленнее, чем чтение печатного текста. Даже те пользователи, которые не знакомы с результатами подобных исследований, отмечают, что они испытывают некоторые неудобства при чтении информации, представленной в электронном виде. В результате этого люди избегают чтения больших объемов текста с экрана монитора. Таким образом, объем текста следует снижать на 50% (снижение объема на 25% оказывается недостаточным), поскольку в данном случае значение имеет не только скорость чтения, но и удобство восприятия информации. Также известно, что пользователи не любят прибегать к прокрутке, что является еще одной причиной для сокращения объема страниц.

Рисунки и фотографии

1. Количество графической информации на web-страницах необходимо максимально ограничивать, поскольку загрузка графики требует значительного времени. От излишней графики следует попросту отказаться. В разряд такой графики попадает любая текстовая информация, представленная в графическом формате, за исключением текста, тесно связанного с общей концепцией оформления сайта, включение которого в состав изображения обусловлено необходимостью.

2. Несмотря на это, пользователям нужно видеть фотографии образцов предлагаемой продукции, поскольку это является единственным способом познакомиться с ней воочию. На

персональной странице целесообразно помещать фотографию или текст, говорящие о личности автора. Устранить противоречия между этими двумя требованиями помогают гипертекстовые возможности Сети. Количество графической информации на страницах верхних уровней следует сводить к минимуму. При просмотре этих страниц пользователь еще не успевает выбрать материалы, которые представляют для него интерес и нуждаются в иллюстрациях. На страницах, посвященных более узким вопросам, количество иллюстративного материала может быть увеличено.

3. На странице верхнего уровня, посвященной определенной продукции, можно поместить небольшую фотографию образца этой продукции, однако основной объем информации по-прежнему должен быть представлен текстом и табличными данными. Если пользователь действительно заинтересовался описываемым образцом, он может воспользоваться ссылками на странице описания для просмотра других фотографий этого образца. Эти фотографии должны быть достаточно крупными, чтобы давать потенциальному покупателю максимально полное представление о предлагаемой продукции.

При использовании изображений настоятельно рекомендуется проводить оптимизацию формата в специализированных графических редакторах, например, Adobe Photoshop. Пакет позволяет сжимать графические файлы с регулированием потери качества. Как правило, файлы большого размера можно существенно сжать так, чтобы при этом потеря качества изображения была незаметна. Это существенно ускорит загрузку браузером файла изображения. При публикации фотографий просмотр оригинала фотографии, то есть файла в исходном качестве, рекомендуется оставлять на усмотрение пользователя: для демонстрации на странице выкладывать сжатый вариант фотографии, а под картинку помещать гиперссылку на оригинал.

Анимированные изображения имеют право на существование в web-дизайне, однако в общем случае использование анимации лучше всего ограничивать. Кроме того, нельзя допускать воспроизведения анимации в бесконечном цикле достаточно, чтобы она воспроизводилась всего несколько раз. Отдельные пользователи полагают, что анимированные изображения выглядят эффектно и

являются показателем труда, вложенного в разработку сайта. Тем не менее, большинство пользователей отмечает, что анимация раздражает их. В частности, практически все не переносят движущийся текст и бегущие строки.

Применение анимации имеет семь целей:

1. Для передачи переходных процессов;
2. Для указания направленности действия;
3. Для передачи изменений, происходящих с течением времени;
4. Для смены, отображаемой в отдельной области страницы информации;
5. Для обогащения графического представления;
6. Для визуализации объемных структур;
7. Для привлечения внимания.

Ссылки

Ссылки – самая важная часть гипертекста. Они объединяют страницы и позволяют пользователям путешествовать по Сети. Существуют три основные формы ссылок:

- Структурные ссылки или ссылки на элементы навигации. Эти ссылки задают структуру информационного пространства, и по ним пользователь переходит к другим разделам сайта. Типичный пример, кнопка возврата на начальную страницу и ссылки на подчиненные страницы.

- Ассоциативные ссылки внутри страницы. Чаще всего это подчеркнутые слова в тексте (хотя это могут быть и изображения), указывающие на страницы, где можно найти подробную информацию о слове, играющем роль ссылки.

- Списки типа “смотрите также”. Эти ссылки нужны для того чтобы помочь пользователю найти то, что он хочет, но чего нет на данной странице. Учитывая сложность навигации в Интернете, хорошо составленные списки “смотрите также” могут сильно облегчить жизнь пользователю.

Гипертекстовые ссылки привязаны к тексту, по которому щелкает пользователь, чтобы перейти по ссылке. Этот текст не должен быть очень длинным, так как пользователи проглядывают страницы в поисках ссылок, стремясь узнать, что они могут сделать на данной странице. Ссылки выполняют ту же роль, что и выноски

в печатных журналах: именно на них останавливается взгляд пользователя при просмотре страницы. Если слишком много слов играют роль ссылок, то пользователь не сможет понять их смысл при беглом взгляде на страницу. Гиперссылками должны быть только понятия, несущие важную информацию.

Подчеркивать информационно значимые слова важно. Но еще лучше добавить краткий текст, поясняющий тип предлагаемой дополнительной информации.

Хотя сама гиперссылка не должна быть длиннее четырех слов, нелишним будет добавить описание, поясняющее смысл ссылки. Особенно это касается ссылок, которые выглядят почти одинаково. Чтобы пользователь мог понять, по какой именно ссылке он может найти нужную ему информацию, нужно дать короткое описание каждой из таких ссылок.

Начиная с браузера Internet Explorer 4.0, поддерживается возможность выводить всплывающие подсказки с описанием ссылки. Это помогает узнать, куда приведет ссылка, и улучшает возможности навигации.

Описание ссылки называется *заголовком ссылки* и его очень легко закодировать.

Если поместить указатель мыши в браузере над этой ссылкой, то примерно через секунду появится всплывающая подсказка.

Цель заголовков – помочь пользователям предугадать, что произойдет, если они перейдут по ссылке. В качестве заголовка для ссылки можно включать следующую информацию:

- Название сайта, на который ведет ссылка (если это не текущий сайт).
- Название подраздела сайта, на который ведет ссылка (если она не выводит за пределы сайта).
- Дополнительные сведения об информации той страницы, на которую указывает ссылка, или о том, как она связана с текстом ссылки и содержимым текущей страницы.

Заголовки ссылок должны быть меньше 80 символов, но они редко превышают даже 60 символов. Чем короче заголовок, тем лучше.

Кроме того, не обязательно придумывать заголовки для всех

ссылок. Если из текста ссылки и контекста понятно, куда она ведет, то ее заголовок только уменьшит удобство использования сайта, так как это будет лишний элемент, на который должен будет посмотреть пользователь. Излишне просто повторять текст ссылки в заголовке.

Заголовок ссылки не будет выглядеть одинаково для всех пользователей. На самом деле браузеры с голосовым интерфейсом прочитают заголовок ссылки вслух, а не отобразят ее на экране.

Использование заголовков ссылок не заменяет сами ссылки. Текст ссылки и контекст, в котором она приводится, должны быть понятны, даже если пользователь и не посмотрит на ее заголовок.

Большинство браузеров используют два цвета для отображения ссылок: ссылки на страницы, которые пользователь не посещал раньше, обычно отображаются синим цветом, ссылки на страницы, где пользователь уже бывал, отображаются красным или пурпурным цветом. Для удобства пользования сайтом необходимо всегда оставлять эти цвета. И хотя не обязательно использовать те же оттенки, что заданы в браузере по умолчанию, необходимо, чтобы не посещенные ссылки были бы синими, а посещенные красными.

Синий цвет сложнее читать, чем черный или красный (предполагаем, что фон белого цвета), так как в человеческом глазе меньше рецепторов, воспринимающих длину волны, соответствующую синему цвету. Но, несмотря на этот физиологический факт, рекомендуют использовать для обозначения ссылок синий цвет. Причина проста: люди привыкли к тому, что синий цвет – это цвет сыпки, поэтому они без заминок разберутся, как работать со страницей.

Когда используются нестандартные цвета для ссылок, люди не могут точно определить, какие части сайта они уже посетили, а какие еще нет. Чувство структуры и положения на сайте у пользователя сильно ослабевает, и в результате страдает его способность обдуманной навигации.

Язык html

Данные в формате HTML похожи на текстовый файл, за исключением того, что некоторые из символов интерпретируются как разметка. Разметка придает документу некую структуру.

Данные представляют собой иерархию элементов. Каждый элемент имеет имя, атрибуты и несет некую информацию. Большинство элементов представлены в документе в виде начальной метки, указывающей имя и атрибуты. Далее следует собственно содержание элемента. И наконец, заканчивает все это конечная метка.

HTML документ состоит из следующих базисных элементов:
 тип документа `<HTML> </HTML>` (начало и конец файла);
 заголовок `<HEAD> </HEAD>` (описание документа, например его имя);

имя документа `<TITLE></TITLE>` (должно быть в заголовке);

тело `<BODY></BODY>` (содержит публикуемые данные).

Например:

```
<HTML>
  <HEAD>
    <TITLE>Название документа</TITLE>
  </HEAD>
  <BODY>
    Информация, содержащаяся в документе.
  </BODY>
</HTML>.
```

Каждый элемент HTML документа начинается с метки, меткой же и заканчивается каждый непустой элемент. Начальные метки выделяются символами `<` и `>`, а конечные - символами `</` и `>`.

Имя элемента следует в метке сразу за символом открытия `<`. Имя начинается с буквы, за которой могут следовать еще 33 буквы, цифры, пробелы или дефисы. В именах игнорируется разница между прописными и строчными буквами.

Начальная метка позволяет вставить между именем и символом `>` пробелы и атрибуты. Атрибут состоит из имени, символа равенства и значения. Слева и справа от символа равенства можно оставлять пробелы.

Значение атрибута указывается в виде строки, заключенной в одинарные или двойные кавычки.

Документы должны (но не обязательно) содержать элемент HEAD, за которым следует элемент BODY.

Тэг <HEAD>

Элемент HEAD содержит всю информацию о документе в целом. Однако он не содержит какого-либо текста. Последний должен находиться в элементе BODY. В элементе заголовка HEAD можно использовать лишь строго заданный набор элементов.

Нижеприведенные элементы определяют общие свойства документа. Они должны появляться в элементе HEAD. Порядок следования элементов значения не имеет.

<TITLE> - название элемента.

<ISINDEX> - элемент, посылаемый серверу вместе с документом, предназначенным для информации к поиску.

<LINK> - элемент, определяющий связь этого документа с другими. В документе может присутствовать несколько элементов LINK.

<BASE> - запись, сделанная на языке URL при фиксировании данного , документа.

<META> - метаинформация.

Например:

<HEAD>

<LINK REV="made" TITLE="Vasia Ivanov"
HREF="mailto:vasia@ivanov.ru">

<META NAME="KeyWords" CONTENT="ключевые слова для поисковых машин">

<META NAME="Description" CONTENT="краткое описание документа">

<TITLE>Домашняя страничка Васи Иванова</TITLE>

</HEAD>

Тэг <BODY>

В противоположность элементу HEAD элемент BODY содержит всю ту информацию, из которой собственно и состоит рассматриваемый документ.

Далее рассматриваются элементы, применяемые для представления и форматирования информации в документе.

Определение структуры текста

Для определения структуры текста используются следующие тэги.

<H?></H?> - заглавие (стандарт определяет 6 уровней).

<H? ALIGN=LEFT|CENTER|RIGHT></H?> - заглавие с выравниванием.

<DIV></DIV> - секция.

<DIV ALIGN=LEFT|RIGHT|CENTER></DIV> - секция с выравниванием.

<BLOCKQUOTE></BLOCKQUOTE> - цитата (обычно выделяется отступом).

 - выделение (обычно изображается курсивом).

 - дополнительное выделение (обычно изображается жирным шрифтом).

<CITE></CITE> - отсылка, цитата (обычно курсив).

<CODE></CODE> - код (для листингов кода).

<SAMP></SAMP> - пример вывода.

<KBD></KBD> - ввод с клавиатуры.

<VAR></VAR> - переменная.

<BIG></BIG> - большой шрифт.

<SMALL></SMALL> - маленький шрифт.

Внешний вид

Внешний вид текста определяется с использованием следующих тэгов.

 - жирный.

<I></I> - курсив.

 - верхний индекс.

 - нижний индекс.

<TT></TT> - печатная машинка (изображается как шрифт фиксированной ширины).

<PRE></PRE> - форматированный (сохранить формат текста как есть).

<PRE WIDTH=?></PRE> - ширина (в символах).

<CENTER></CENTER> - центрировать (как текст, так и графика).

 - размер шрифта (от 1 до 7).
 - изменить размер шрифта.
 - цвет шрифта.
 - выбор шрифта.

Ссылки и графика

Следующие тэги используются для организации ссылок и работы с графикой.

 - ссылка.

 - ссылка на закладку (в другом документе).

 - ссылка на закладку (в том же документе).

 - ссылка на другое окно.

 - определить закладку.

 - графика.

 - графика с выравниванием.

 - альтернатива (выводится если картинка не изображается).

 - размеры (в точках).

 - окантовка (в точках).

 - отступ (в точках).

Разделители

Следующие тэги используются в качестве разделителей.

<P></P> параграф (закрывать элемент часто не обязательно).

<P ALIGN=LEFT|CENTER|RIGHT></P> - выравнивание.

 - новая строка (одиночный перевод строки).

<BR CLEAR=LEFT|RIGHT|ALL> - убрать выравнивание.

<HR> - горизонтальный разделитель.

<HR ALIGN=LEFT|RIGHT|CENTER> - выравнивание.

<HR SIZE=?> - толщина (в точках).

<HR WIDTH=?> - ширина (в точках).

`<HR WIDTH="% ">` - ширина в процентах (в процентах от ширины страницы).

`<HR NOSHADE>` - сплошная линия (без трехмерных эффектов).

Списки

Для реализации списков используются следующие тэги.

`` - неупорядоченный (`` перед каждым элементом).

`<UL COMPACT>` - компактный.

`<UL TYPE=DISC|CIRCLE|SQUARE>` - тип метки (для всего списка).

`<LI TYPE=DISC|CIRCLE|SQUARE>` - (этот и последующие).

`` - нумерованный (`` перед каждым элементом).

`<OL COMPACT>` - компактный.

`<OL TYPE=A|a|I|i|1>` - тип нумерации (для всего списка).

`<LI TYPE=A|a|I|i|1>` - (этот и следующие).

`<OL START=?>` - первый номер (для всего списка).

`<LI VALUE=?>` - (этот и следующие).

`<DL><DT><DD></DL>` - список определений (`<DT>`= термин, `<DD>`= определение).

`<DL COMPACT></DL>` - компактный.

`<MENU></MENU>` - меню (`` перед каждым элементом).

`<MENU COMPACT></MENU>` - компактное.

`<DIR></DIR>` - каталог (`` перед каждым элементом).

`<DIR COMPACT></DIR>` - компактный.

Фон и цвета

Следующие тэги определяют фон и цвета.

`<BODY BACKGROUND="URL">` - фоновая картинка.

`<BODY BGCOLOR="#$$$$$$">` - цвет фона.

`<BODY TEXT="#$$$$$$">` - цвет текста.

`<BODY LINK="#$$$$$$">` - цвет ссылки.

`<BODY VLINK="#$$$$$$">` - пройденная ссылка.

`<BODY ALINK="#$$$$$$">` - активная ссылка.

5.9 Специальные символы

(обязаны быть в нижнем регистре)

&#?; - специальный символ (где ? это код ISO 8859-1).

< - <.

> - >.

& - &.

" - ".

® - торговая марка ТМ.

© - copyright.

 - неразделяющий пробел.

Формы

Для работы с формами используются следующие тэги.

<FORM ACTION="URL" METHOD=GET|POST></FORM> -
определить форму.

<INPUT
TYPE="TEXT|PASSWORD|CHECKBOX|RADIO|IMAGE|HIDDEN|
SUBMIT|RESET"> - поле ввода.

<INPUT NAME="***"> - имя поля.

<INPUT VALUE="***"> - значение поля.

<INPUT CHECKED> - отмечен (checkboxes и radio boxes).

<INPUT SIZE=?> - размер поля (в символах).

<INPUT MAXLENGTH=?> - максимальная длина (в
символах).

<SELECT></SELECT> - список вариантов.

<SELECT NAME="***"></SELECT> - имя списка.

<SELECT SIZE=?></SELECT> - число вариантов.

<SELECT MULTIPLE> - множественный выбор (можно
выбрать больше одного).

<OPTION>- опция (элемент, который может быть выбран).

<OPTION SELECTED> - опция по умолчанию.

<TEXTAREA ROWS=? COLS=?></TEXTAREA> - ввод текста,
размер.

<TEXTAREA NAME="***"></TEXTAREA> - имя текста.

Таблицы

Для организации таблиц используются следующие тэги.

<TABLE></TABLE> - определить таблицу.

- <TABLE BORDER=?></TABLE> - окантовка таблицы.
- <TABLE CELSPACING=?> - расстояние между ячейками.
- <TABLE CELLPADDING=?> - дополнение ячеек.
- <TABLE WIDTH=?> - желаемая ширина (в точках).
- <TABLE WIDTH="% "> - ширина в процентах (проценты от ширины страницы).
- <TR></TR> - строка таблицы.
- <TR ALIGN=LEFT|RIGHT| CENTER|MIDDLE|BOTTOM>
- выравнивание.
- <TD></TD> - ячейка таблицы (должна быть внутри строки).
- <TD ALIGN=LEFT|RIGHT| CENTER|MIDDLE|BOTTOM>
- выравнивание.
- <TD NOWRAP> - без перевода строки.
- <TD COLSPAN=?> - растягивание по колонке.
- <TD ROWSPAN=?> - растягивание по строке.
- <TD WIDTH=?> - желаемая ширина (в точках).
- <TD WIDTH="% "> - ширина в процентах (проценты от ширины страницы).
- <TD BGCOLOR="#\$\$\$\$\$\$"> - цвет ячейки.
- <TH></TH> - заголовок таблицы (как данные, но жирный шрифт и центровка).
- <TH ALIGN=LEFT|RIGHT| CENTER|MIDDLE|BOTTOM>
- выравнивание.
- <TH NOWRAP> - без перевода строки.
- <TH COLSPAN=?> - растягивание по колонке.
- <TH ROWSPAN=?> - растягивание по строке.
- <TH WIDTH=?> - желаемая ширина (в точках).
- <TH WIDTH="% "> - ширина в процентах (проценты ширины таблицы).
- <TH BGCOLOR="#\$\$\$\$\$\$"> - цвет ячейки.
- <CAPTION></CAPTION> - заглавие таблицы.
- <CAPTION ALIGN=TOP|BOTTOM> - выравнивание (сверху/снизу таблицы).

Фреймы

Фреймы (Frames) – это элементы HTML, появившиеся в браузерах версий 3.0. Позволяют разделить страницу на несколько независимых окон и в каждом из них размещать свою собственную

WEB-страничку. Возможна ссылка из одного окна в другое. Применяется в основном для организации постоянно находящихся на экране меню, в то время как в другом окне располагается непосредственно сама информация.

Для работы с фреймами используются следующие тэги.

`<FRAMESET></FRAMESET>` - документ с фреймами (вместо `<BODY>`).

`<FRAMESET ROWS=,,,></FRAMESET>` - высота строк (точки или %).

`<FRAMESET ROWS=*></FRAMESET>` - высота строк (* = относительный размер).

`<FRAMESET COLS=,,,></FRAMESET>` - ширина колонок (точки или %).

`<FRAMESET COLS=*></FRAMESET>` - ширина колонок (* = относительный размер).

`<FRAMESET BORDER=?>` - ширина окантовки.

`<FRAMESET FRAMEBORDER="yes|no">` - окантовка.

`<FRAMESET BORDERCOLOR="#$$$$$$">` - цвет окантовки.

`<FRAME>` - определить фрейм (содержание отдельного фрейма).

`<FRAME SRC="URL">` - документ.

`<FRAME NAME="***"|_blank|_self|_parent|_top>` - имя фрейма.

`<FRAME MARGINWIDTH=?>` - ширина границы (правая и левая границы).

`<FRAME MARGINHEIGHT=?>` - высота границы (верхняя и нижняя границы).

`<FRAME SCROLLING="YES|NO|AUTO">` - скроллинг.

`<FRAME NORESIZE>` - постоянный размер.

`<FRAME FRAMEBORDER="yes|no">` - окантовка.

`<FRAME BORDERCOLOR="#$$$$$$">` - цвет окантовки.

`<NOFRAMES></NOFRAMES>` - содержание без фреймов (для браузеров, не поддерживающих фреймы).

Комментарий

`<!-- *** -->` - тэг комментария (игнорируется браузером).

Задание

Создать на языке гипертекстовой разметки HTML пример Вашей персональной домашней странички.

Страница должна содержать:

Заголовок (тэг <HEAD>), включающий в себя:

- название страницы (тэг <TITLE>), содержащее Ваше имя;
- обозначение кодировки страницы (рекомендуется использовать кодировку WINDOWS-1251, тэг <meta http-equiv=Content-Type content="text/html; charset=windows-1251">);
- Описание страницы (тэг <meta name="Description">);
- Описание ключевых слов (тэг <meta name="Keywords">).

Тело страницы (тэг <BODY>), включающее в себя:

- Меню с ссылками внутри документа;
- Текст с краткой информацией о себе;
- Изображение, "обтекаемое" вышеуказанным текстом;
- Ссылку на адрес Вашей электронной почты;
- Ссылки на различные сайты (сайты должны открываться в новом окне).

При проектировании странички рекомендуется пользоваться программами Notepad или специализированным редактором Dreamweaver.

Отчет должен содержать:

1. Листинг файла-страницы на языке HTML.
2. Описание используемых тэгов.
3. Распечатку окна браузера с загруженной персональной страницей.

Контрольные вопросы

1. Из каких этапов строится работа при создании интернет-сайта?
2. Какие цели ставят перед интернет-сайтом?
3. Что такое «техническое задание»?
4. Что такое «целевая аудитория» и для чего ее нужно исследовать?
5. Как оценить эффективность интернет-сайта?

Разработка базы данных

Цель работы: закрепить навыки разработки реляционных баз данных.

1. Модель «сущность-связь».

Прежде, чем приступить к созданию системы автоматизированной обработки информации, разработчик должен сформировать понятия о предметах, фактах и событиях, которыми будет оперировать данная система. Для того, чтобы привести эти понятия к той или иной модели данных, необходимо заменить их информационными представлениями. Одним из наиболее удобных инструментов унифицированного представления данных, независимого от реализующего его программного обеспечения, является модель "сущность-связь" (entity - relationship model, ER - model).

Модель "сущность-связь" основывается на некой важной семантической информации о реальном мире и предназначена для *логического* представления данных. Она определяет значения данных в контексте их взаимосвязи с другими данными. Важным для нас является тот факт, что из модели "сущность-связь" могут быть порождены все существующие модели данных (иерархическая, сетевая, реляционная, объектная), поэтому она является наиболее общей.

Любой фрагмент предметной области может быть представлен как *множество сущностей*, между которыми существует некоторое *множество связей*. Дадим определения:

Сущность (entity) – это объект, который может быть идентифицирован неким способом, отличающим его от других объектов. Примеры: *конкретный человек, предприятие, событие и т.д.*

Набор сущностей (entity set) – множество сущностей одного типа (обладающих одинаковыми свойствами). Примеры: *все люди, предприятия, праздники и т.д.* Наборы сущностей не обязательно должны быть непересекающимися. Например, сущность, принадлежащая к набору МУЖЧИНЫ, также принадлежит набору ЛЮДИ.

Сущность фактически представляет из себя множество **атрибутов**, которые описывают свойства всех членов данного набора сущностей.

Пример: рассмотрим множество работников некоего предприятия. Каждого из них можно описать с помощью характеристик *табельный номер, имя, возраст*. Поэтому, сущность СОТРУДНИК имеет атрибуты ТАБЕЛЬНЫЙ_НОМЕР, ИМЯ, ВОЗРАСТ. Используя нотацию языка Pascal этот факт можно представить как:

```

type employe = record

    number : string[6];

    name   : string[50];

    age    : integer;

end;
```

В дальнейшем для определения сущности и ее атрибутов будем использовать обозначение вида

СОТРУДНИК (ТАБЕЛЬНЫЙ_НОМЕР, ИМЯ, ВОЗРАСТ).

Например, отделы, на которые подразделяется предприятие, и в которых работают сотрудники, можно описать как ОТДЕЛ (НОМЕР_ОТДЕЛА, НАИМЕНОВАНИЕ).

Множество значений (область определения) атрибута называется **доменом**. Например, для атрибута ВОЗРАСТ домен (назовем его ЧИСЛО_ЛЕТ) задается интервалом целых чисел больших нуля, поскольку людей с отрицательным возрастом не бывает.

Ключ сущности – это группа атрибутов, такая, что отображение набора сущностей в соответствующую группу наборов значений является взаимнооднозначным отображением. Другими словами: ключ сущности - это один или более атрибутов, уникально определяющих данную сущность. В нашем примере ключом сущности СОТРУДНИК является атрибут ТАБЕЛЬНЫЙ_НОМЕР

(конечно, только в том случае, если все табельные номера на предприятии уникальны).

Связь (relationship) - это ассоциация, установленная между несколькими сущностями. Примеры:

- поскольку каждый сотрудник работает в каком-либо отделе, между сущностями СОТРУДНИК и ОТДЕЛ существует связь "работает в" или ОТДЕЛ-РАБОТНИК;
- так как один из работников отдела является его руководителем, то между сущностями СОТРУДНИК и ОТДЕЛ имеется связь "руководит" или ОТДЕЛ-РУКОВОДИТЕЛЬ;
- могут существовать и связи между сущностями одного типа, например связь РОДИТЕЛЬ - ПОТОМОК между двумя сущностями ЧЕЛОВЕК;

Здесь следует отметить, что в методике проектирования данных есть своеобразное правило хорошего тона, согласно которому сущности обозначаются с помощью имен существительных, а связи - глагольными формами. Данное правило, однако, не является обязательным.

К сожалению, не существует общих правил определения, что считать сущностью, а что связью. В рассмотренном выше примере мы положили, что "руководит" - это связь. Однако, можно рассматривать сущность "руководитель", которая имеет связи "руководит" с сущностью "отдел" и "является" с сущностью "сотрудник".

Связь также может иметь атрибуты. Например, для связи ОТДЕЛ-РАБОТНИК можно задать атрибут СТАЖ_РАБОТЫ_В_ОТДЕЛЕ.

Роль сущности в связи - функция, которую выполняет сущность в данной связи. Например, в связи РОДИТЕЛЬ-ПОТОМОК сущности ЧЕЛОВЕК могут иметь роли "родитель" и "потомок". Указание ролей в модели "сущность-связь" не является обязательным и служит для уточнения семантики связи.

Набор связей (relationship set) - это отношение между n (причем n не меньше 2) сущностями, каждая из которых относится к некоторому набору сущностей.

То число сущностей, которое может быть ассоциировано через набор связей с другой сущностью, называют **степенью связи**.

Рассмотрение степеней особенно полезно для бинарных связей. Могут существовать следующие степени бинарных связей:

- *один к одному (обозначается $1 : 1$)*. Это означает, что в такой связи сущности с одной ролью всегда соответствует не более одной сущности с другой ролью. В рассмотренном нами примере это связь "руководит", поскольку в каждом отделе может быть только один начальник, а сотрудник может руководить только в одном отделе. Данный факт представлен на следующем рисунке, где прямоугольники обозначают сущности, а ромб - связь. Так как степень связи для каждой сущности равна 1, то они соединяются одной линией.

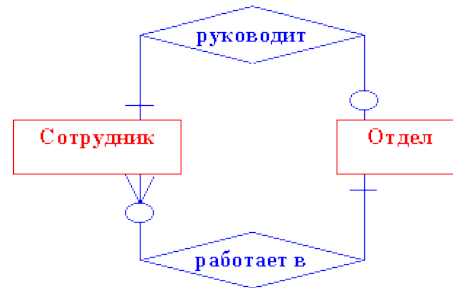


Другой важной характеристикой связи помимо ее степени является **класс принадлежности** входящих в нее сущностей или **кардинальность** связи. Так как в каждом отделе обязательно должен быть руководитель, то каждой сущности "ОТДЕЛ" непременно должна соответствовать сущность "СОТРУДНИК". Однако, не каждый сотрудник является руководителем отдела, следовательно в данной связи не каждая сущность "СОТРУДНИК" имеет ассоциированную с ней сущность "ОТДЕЛ".

Таким образом, говорят, что сущность "СОТРУДНИК" имеет *обязательный класс принадлежности* (этот факт обозначается также указанием интервала числа возможных вхождений сущности в связь, в данном случае это 1,1), а сущность "ОТДЕЛ" имеет *необязательный класс принадлежности* (0,1). Теперь данную связь мы можем описать как $0,1:1,1$. В дальнейшем кардинальность бинарных связей степени 1 будем обозначать следующим образом:

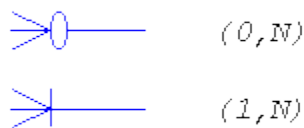


- *один ко многим ($1 : n$)*. В данном случае сущности с одной ролью может соответствовать любое число сущностей с другой ролью. Такова связь ОТДЕЛ-СОТРУДНИК. В каждом отделе может работать произвольное число сотрудников, но сотрудник может работать только в одном отделе. Графически степень связи n отображается "древообразной" линией, так это сделано на следующем рисунке.



Данный рисунок дополнительно иллюстрирует тот факт, что между двумя сущностями может быть определено несколько наборов связей.

Здесь также необходимо учитывать класс принадлежности сущностей. Каждый сотрудник должен работать в каком-либо отделе, но не каждый отдел (например, вновь сформированный) должен включать хотя бы одного сотрудника. Поэтому сущность "ОТДЕЛ" имеет обязательный, а сущность "СОТРУДНИК" необязательный классы принадлежности. Кардинальность бинарных связей степени n будем обозначать так:



- *много к одному* ($n : 1$). Эта связь аналогична отображению $1 : n$. Предположим, что рассматриваемое нами предприятие строит свою деятельность на основании контрактов, заключаемых с заказчиками. Этот факт отображается в модели "сущность-связь" с помощью связи КОНТРАКТ-ЗАКАЗЧИК, объединяющей сущности КОНТРАКТ(НОМЕР, СРОК_ИСПОЛНЕНИЯ, СУММА) и ЗАКАЗЧИК(НАИМЕНОВАНИЕ, АДРЕС). Так как с одним заказчиком может быть заключено более одного контракта, то связь КОНТРАКТ-ЗАКАЗЧИК между этими сущностями будет иметь степень $n : 1$.



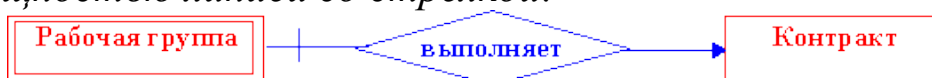
В данном случае, по совершенно очевидным соображениям (каждый контракт заключен с конкретным заказчиком, а каждый

заказчик имеет хотя бы один контракт, иначе он не был бы таковым), каждая сущность имеет обязательный класс принадлежности.

- *многие ко многим* ($n : n$). В этом случае каждая из ассоциированных сущностей может быть представлена любым количеством экземпляров. Пусть на рассматриваемом нами предприятии для выполнения каждого контракта создается рабочая группа, в которую входят сотрудники разных отделов. Поскольку каждый сотрудник может входить в несколько (в том числе и ни в одну) рабочих групп, а каждая группа должна включать не менее одного сотрудника, то связь между сущностями СОТРУДНИК и РАБОЧАЯ_ГРУППА имеет степень $n : n$.



Если существование сущности x зависит от существования сущности y , то x называется **зависимой сущностью** (иногда сущность x называют "слабой", а "сущность" y - *сильной*). В качестве примера рассмотрим связь между ранее описанными сущностями РАБОЧАЯ_ГРУППА и КОНТРАКТ. Рабочая группа создается только после того, как будет подписан контракт с заказчиком, и прекращает свое существование по выполнению контракта. Таким образом, сущность РАБОЧАЯ_ГРУППА является зависимой от сущности КОНТРАКТ. Зависимую сущность будем обозначать двойным прямоугольником, а ее связь с сильной сущностью линией со стрелкой:



Заметим, что кардинальность связи для сильной сущности всегда будет (1,1). Класс принадлежности и степень связи для зависимой сущности могут быть любыми. Предположим, например, что рассматриваемое нами предприятие пользуется несколькими банковскими кредитами, которые представляются набором сущностей КРЕДИТ (НОМЕР_ДОГОВОРА, СУММА, СРОК_ПОГАШЕНИЯ, БАНК). По каждому кредиту должны осуществляться выплаты процентов и платежи в счет его погашения. Этот факт представляется набором сущностей ПЛАТЕЖ (ДАТА, СУММА) и набором связей "осуществляется по". В том случае, когда получение запланированного кредита

отменяется, информация о нем должна быть удалена из базы данных. Соответственно, должны быть удалены и все сведения о плановых платежах по этому кредиту. Таким образом, сущность ПЛАТЕЖ зависит от сущности КРЕДИТ.



Очень важным свойством модели "сущность-связь" является то, что она может быть представлена в виде графической схемы. Это значительно облегчает анализ предметной области и позволяет на основе схемы создавать таблицы в СУБД.

2. Базовые понятия реляционных баз данных

Основными понятиями реляционных баз данных являются тип данных, домен, атрибут, кортеж, первичный ключ и отношение.

Понятие *тип данных* в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных базах данных допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных (соответствующими возможностями обладают, например, системы семейства Ingres/Postgres).

Понятие *домена* более специфично для баз данных, хотя и имеет некоторые аналогии с подтипами в некоторых языках программирования. В самом общем виде домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.

Наиболее правильной интуитивной трактовкой понятия домена является понимание домена как допустимого потенциального множества значений данного типа.

Схема отношения базы данных - это именованное множество пар {имя атрибута, имя домена (или типа, если понятие домена не поддерживается)}. Степень или "арность" схемы отношения - мощность этого множества. Если все атрибуты одного отношения определены на разных доменах, осмысленно использовать для именованния атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именованния и не устраняет различия между понятиями домена и атрибута).

Схема **базы данных** (в структурном смысле) - это набор именованных схем отношений.

Кортеж, соответствующий данной схеме отношения, — это набор именованных значений заданного типа.

Обычным представлением отношения является таблица, в заголовке которой указывают наименование отношения, а в строках — кортежи; в этом случае имена атрибутов именуют столбцы таблицы. Поэтому, когда говорят «столбец таблицы», имеют в виду «атрибут отношения». Такой терминологии придерживаются в большинстве коммерческих реляционных СУБД.

Реляционная модель данных. Модель данных с точки зрения СУБД описывает некоторый набор родовых понятий и признаков, которыми должны обладать сама СУБД и управляемые ею базы данных, если они основываются на этой модели. Наличие модели данных позволяет сравнивать конкретные реализации, используя один общий язык.

Реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной, манипуляционной и целостной.

В структурной части модели фиксируется, что единственной структурой данных, используемой в реляционных БД, является нормализованное отношение — некоторый определенный набор ограничений, свойственный этому отношению. Нормализация связана с поиском наиболее простой структуры для данного множества данных и имеет дело с зависимостью между атрибутами; она позволяет избежать потери общей информации при удалении или вводе записей. Существует несколько формальных типов нормализации (более пяти).

Манипуляционная часть модели определяет механизм манипулирования реляционными БД — реляционная алгебра и реляционное исчисление, базирующиеся на теории множеств и логическом аппарате исчисления отношений.

В целостной части реляционной модели данных фиксируются два базовых требования целостности: целостности сущностей и целостности по ссылкам.

Объекту, или сущности, реального мира в реляционных БД соответствуют записи (кортежи) отношений, и требование целостности состоит в сохранении отличий разных записей этого отношения; говорят, что любое отношение должно обладать первичным ключом.

Второе требование более сложно. Сущности реального мира представляются в реляционной БД в виде записей нескольких отношений. Для связи отношений используется атрибут, который служит внешним ключом. Отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом. Требование целостности по ссылкам состоит в том, что для каждого значения внешнего ключа должна найтись запись с таким же значением первичного ключа в отношении, на которое указывает ссылка, либо значение внешнего ключа должно быть неопределенным (т. е. ни на что не указывать). Выполнение таких требований чрезвычайно важно при модификации отношений или удалении записей. Поддержке целостности при удалении кортежа служат: запрет на удаление кортежа, на который существуют ссылки; автоматическая замена значения внешнего ключа на неопределенное во всех ссылающихся кортежах; автоматическое удаление всех ссылающихся кортежей.

Реляционные СУБД обладают рядом особенностей, влияющих на организацию внешней памяти. К наиболее важным особенностям можно отнести следующие:

- наличие двух уровней системы: уровня непосредственного управления данными во внешней памяти и языкового уровня (например, уровня, реализующего язык SQL); тогда подсистема нижнего уровня должна поддерживать во внешней памяти набор

базовых структур, конкретная интерпретация которых входит в число функций подсистемы верхнего уровня;

- информация, связанная с наименованием объектов базы данных и их конкретными свойствами (например, структура ключа-индекса), поддерживается подсистемой языкового уровня;

- регулярность структур данных во внешней памяти, поскольку основным объектом реляционной модели данных является плоская таблица;

- для выполнения операторов языкового уровня над одним или несколькими отношениями во внешней памяти поддерживаются дополнительные «управляющие» структуры — индексы;

- для выполнения требования надежного хранения баз данных поддерживается избыточность хранения данных во внешней памяти.

Следует подчеркнуть, что как бы ни были организованы индексы в конкретной СУБД, их основное назначение состоит в обеспечении эффективного прямого доступа к кортежу отношения по ключу. Обычно индекс определяется для одного отношения, и ключом является значение атрибута (возможно, составного). Организация индексов в больших БД представляет сложную проблему. Все более популярным подходом к организации индексов является использование техники хэширования. Общей идеей методов хэширования является применение к значению ключа некоторой функции свертки (хэш-функции), вырабатывающей значение меньшего размера. Свертка значения ключа затем применяется для доступа к записи. В самом простом случае свертка ключа используется как адрес в таблице, содержащей ключи и записи.

СУБД Microsoft Access

Простейшие базы данных можно создавать, не прибегая к специальным программным средствам: например, в электронных таблицах Excel. Но существует несколько отличий между ними. Во-первых, системы управления базами данных (СУБД) разрабатываются с целью обеспечения эффективной обработки больших объемов информации, намного больших, чем те, с которыми справляются электронные таблицы. Во-вторых, СУБД

может легко связывать две таблицы так, что для пользователя они будут представляться одной таблицей. Реализовать такую возможность в электронных таблицах практически невозможно.

И в-третьих, СУБД минимизирует общий объем базы данных. Для этой же цели таблицы, содержащие повторяющиеся данные, разбиваются на несколько связанных таблиц.

Несмотря на то, что в мире существует множество систем управления базами данных, все они имеют некоторые сходства. Основное назначение СУБД Microsoft Access 2007 из пакета Microsoft Office – дать пользователю простое и доступное средство для создания таких баз данных, которые ему нужны. Несмотря на простоту этой СУБД, она позволяет создавать приложения довольно сложной структуры. При желании систему можно развивать и настраивать собственными силами, используя язык программирования Visual Basic.

Еще одним достоинством Access является ее интегрированность с Word, Excel и другими программами пакета Microsoft Office. Данные легко импортируются и экспортируются из одного приложения в другое.

Объекты базы данных Access

К объектам базы данных Access относятся:

1. *Таблицы* – предназначены для упорядоченного хранения данных.
2. *Запросы* – предназначены для поиска, извлечения данных и выполнения вычислений.
3. *Формы* – предназначены для удобного просмотра, изменения и добавления данных в таблицах.
4. *Отчеты* – используются для анализа и печати данных.
5. *Страницы доступа к данным* – предназначены для просмотра, ввода, обновления и анализа данных через сеть или из любого места компьютера.
6. *Макросы* – используются для выполнения часто встречающегося набора макрокоманд, осуществляющих обработку данных.
7. *Модули* – предназначены для описания инструкций и процедур на языке VBA.

Основным объектом базы данных является таблица, которая состоит из записей (строк) и полей (столбцов). На пересечении записи и поля образуется ячейка, в которой содержатся данные.

Каждому полю таблицы присваивается *уникальное имя*, которое не может содержать более 64 символов. В каждом поле содержатся данные одного типа.

Типы данных

Типы данных	Описание
Текстовый	Используется для хранения символьных или числовых данных, не требующих вычислений. В свойстве <i>Размер поля</i> задается максимальное количество символов, которые могут быть введены в данное поле. По умолчанию размер устанавливается в 50 знаков. Максимальное количество символов, которые могут содержаться в текстовом поле, – 255
Поле МЕМО	Предназначено для ввода текстовой информации, по объему превышающей 255 символов; может содержать до 65536 символов
Числовой	Предназначен для хранения числовых данных, используемых в математических расчетах. На вкладках <i>Общие</i> и <i>Подстановка</i> можно установить свойства числового поля, среди которых <i>Размер поля</i> , <i>Формат поля</i> , <i>Число десятичных знаков</i>
Дата/время	Используется для представления даты и времени. Выбор конкретного формата даты или времени устанавливается в свойстве <i>Формат даты</i>
Денежный	Предназначен для хранения данных, точность представления которых колеблется от 1 до 4 знаков после запятой. Целая часть может содержать до 15 десятичных знаков

Счётчик	Предназначен для автоматической вставки уникальных последовательных (увеличивающихся на 1) или случайных чисел в качестве номера новой записи. Номер, присвоенный записи, не может быть удален или изменен. Поля с этим типом данных используются в качестве ключевых полей таблицы.
Логический	Предназначен для хранения одного из двух значений, интерпретируемых как «Да/Нет», «Истина/Ложь», «Вкл./ Выкл.»
Поле объекта OLE	Содержит данные, созданные в других программах, которые используют протокол OLE. Это могут быть, например, документы Word, электронные таблицы Excel, рисунки, звуковые и видеозаписи и др. Объекты OLE связываются с базой данных Access или внедряются в нее. Сортировать, группировать и индексировать поля объектов OLE нельзя
Гиперссылка	Специальный тип, предназначенный для хранения гиперссылок
Мастер подстановок	Предназначен для автоматического определения поля. С его помощью будет создано поле со списком, из которого можно выбирать данные, содержащиеся в другой таблице или в наборе постоянных значений

В Microsoft Access полностью автоматизировано создание и редактирование таблиц и всех типов связей между ними. Однако пользователю разрешено самостоятельно писать SQL-запросы.

Запросы являются основным средством просмотра, отбора, изменения и анализа информации, которая содержится в одной или нескольких таблицах базы данных.

Существуют различные виды запросов, но наиболее распространенными являются *запросы на выборку*.

Формы – это объекты базы данных, предназначенные для просмотра данных из таблиц и запросов, для ввода данных в базу, корректирования существующих данных и выполнения заданных

действий. Форма может содержать графики, рисунки и другие внедренные объекты.

Можно вносить данные в таблицы и без помощи каких-либо форм. Но существует несколько причин, которые делают формы незаменимым средством ввода данных в базу:

1) при работе с формами ограничен доступ к таблицам (самоу ценному в базе данных);

2) разные люди могут иметь разные права доступа к информации, хранящейся в базе. Для ввода данных им предоставляются разные формы, хотя данные из форм могут поступать в одну таблицу;

3) вводить данные в форму легче, чем в таблицу, и удобнее, так как в окне формы отображается, как правило, одна запись таблицы;

4) в большинстве случаев информация для баз данных берется из бумажных бланков (анкет, счетов, накладных, справок и т. д.). Экранные формы можно сделать точной копией бумажных бланков, благодаря этому уменьшается количество ошибок при вводе и снижается утомляемость персонала.

Создавать формы можно на основе нескольких таблиц или запросов с помощью Мастера, используя средство автоформы, «вручную» в режиме Конструктора, сохраняя таблицу или запрос как форму. Созданную любым способом форму можно затем изменять в режиме Конструктора.

Задание на лабораторную работу: разработать в Microsoft Access базу данных в соответствии с индивидуальным заданием.

Варианты заданий

1. Коллекция видеофильмов MPEG4 на локальном компьютере. Написать базу, в которой хранятся названия фильмов, жанр (комедия, боевик и др.), режиссер, основные артисты, краткая аннотация, сведения о наградах, год выпуска фильма, продолжительность и путь к файлу (файлам). База предназначена для поиска фильмов по названию, автору, режиссерам, наградам, жанрам, артистам, ключевому слову. Предусмотреть возможность просмотра выбранного фильма.

2. Предприятие занимается прокатом автомобилей гражданам и организациям. База предназначена для хранения сведений об автомобилях, клиентах и договорах проката. Пользователями базы являются владелец проката и клиенты. Владелец хранит информацию об имеющихся автомобилях, обращавшихся к нему клиентам и историю проката машин. В базе зарегистрированы имеющиеся автомобили: модель, цвет, номер, фотография автомобиля, год выпуска, номер двигателя, номер кузова, дата приобретения, дата прохождения последнего техосмотра, сведения о ремонтах и др. При оформлении договоров проката фиксируется: если клиентом является физическое лицо – Ф.И.О. клиента, паспортные данные, контактный телефон, адрес; если клиентом является юридическое лицо – название организации, ИНН, контактный телефон, юридический адрес; а также: номер договора, дата заключения договора, период проката, номер автомобиля, стоимость услуг.

Клиенты с помощью базы могут выбрать понравившуюся модель, просмотрев характеристики и фотографии, имеющихся автомашин, найти среди них те, которые в данный момент свободны и не находятся в ремонте, для остальных определить срок их возврата в пункт проката.

3. Разработать многопользовательскую базу для пункта проката. Пользователями базы являются операторы проката, регистрирующие выдачу и возврат кассет и дисков; и клиенты, производящие поиск интересующих фильмов и музыки.

Клиенты могут взять напрокат ресурсы двумя способами: 1) оставляя в залог документ или деньги, получают номерок, по которому регистрируется заказ; 2) оформляются по паспорту, получая карточку постоянного клиента, и далее заказ регистрируется на карточку. При прокате в базу заносится: номерок или номер карточки постоянного клиента, список взятых кассет и дисков, срок начала и окончания проката.

Клиентам предоставляется возможность поиска по базе интересующих их фильмов и музыки: название кассет и дисков, наличие на текущий момент (если отсутствуют, то предполагаемый срок возврата), инвентарные номера. Музыка ищется по: исполнителю, автору, жанру, году издания альбома.. Видеофильмы

ищутся по названию, жанру, фамилии режиссера, фамилиям актеров. Предусмотреть возможность поиска по шаблону и по типу носителя (кассета, CD, DVD). Для каждого фильма, кроме всего прочего, в базе хранится краткая аннотация.

С целью предсказания спроса предусмотреть возможность статистического анализа: какие жанры музыки и фильмов пользуются наибольшим спросом, вывести десять наиболее популярных фильмов и исполнителей музыки.

4. Разработать базу данных спортивного клуба, в которой будет храниться информация о тренерах, спортсменах, соревнованиях, наградах спортсменов, расписании тренировок и командировок. О тренерах в базе хранится: ФИО, квалификация, имеет ли право на судейскую деятельность; список спортсменов, которых он тренирует, контактный телефон и адрес. О спортсменах хранится: ФИО, разряд, дата рождения, телефон, адрес, список наград, расписание тренировок и участия в соревнованиях, и пр.

База используется:

- для составления личного расписания тренеров (тренировки, судейство, выезд в командировки со своими подопечными);
- для составления личного расписания спортсменов;
- для вывода биографии и послужного списка спортсмена;
- для поиска судей, которые находятся на момент проведения соревнований в городе, и в соревнованиях не участвуют их подопечные.

5. Электронный каталог предназначен для учета и поиска статей в периодических изданиях, которые выписывает библиотека. Учет проводится по нескольким категориям областей знаний. Одна статья может относиться к нескольким категориям. Для каждой статьи хранятся выходные данные (автор, название, название журнала/газеты, год, номер, начальная страница), список ключевых слов и, по возможности, ее краткое содержание.

Поиск статей производят посетители библиотеки. Поиск может вестись по категориям, автору, названию и по ключевым словам. Поисковая система в результате формирует список статей с полными выходными данными и кратким содержанием. Второй тип поиска: вывести список журналов и газет, которые выписывала библиотека в заданный период времени. В списке указываются

номера и годы выпуска. Третий тип поиска: для выбранного журнала и газеты (название может быть задано по маске) вывести список номеров, имеющих в библиотеке.

6. Автотранспортное хозяйство осуществляет перевозки людей и грузов на заказ. Имеющийся транспорт: легковые автомобили, грузовики и автобусы. Каждый из водителей (категорий С, D, E) закреплен за конкретным транспортным средством, но может замещать отсутствующего коллегу в случае, если собственный транспорт находится на ремонте. Все перевозки регистрируются в базе данных: заказчик, время и дата заказа (с точностью до половины дня), тип перевозок, пункт назначения, транспортное средство, водитель. Сроки ремонтов транспорта также протоколируются. Выходные документы:

- полная информация о заказах, выполненных в заданный промежуток времени;
 - "история" перевозок транспортного средства в заданный промежуток времени: куда ездило, кто был водитель, кто заказчик, и пр.;
 - список транспортных средств определенного типа (например, автобусов с числом посадочных мест не менее 20), свободных на заданный интервал времени;
- списки водителей с указанием категории, которые на текущий день остались без работы по причине ремонта их транспорта.

7. База данных издательства хранит информацию о статьях, опубликованных журналистами в журнале, персональных сведениях о журналистах, а также реальное содержимое журнала. О журналистах хранится: ФИО, контактный телефон, адрес постоянного проживания, образование, стаж, штатный/нештатный корреспондент. О статьях: название; год, номер и страницы журнала; число знаков текста, авторы (может быть несколько), жанр (аналитический обзор, интервью, исторический очерк, новости и т.д.), рейтинг (числовая мера качества статьи). Файл с текстом журнала (в формате издательской системы) также помещается в базу данных. База данных применяется для поиска статей по автору и жанру; содержащиеся в ней сведения также могут использоваться поиска областей, в которых наиболее полно раскроется творческий потенциал работника. Выходные документы:

- для заданного журналиста вывести перечень его статей в журналах;
- для каждого работника определить три жанра, статьи в которых обладают наивысшим рейтингом;
- перечень журналистов в порядке убывания их среднего рейтинга.

8. В базе данных хранится информация об автомобилях и водительских удостоверениях граждан. Для автомобиля хранится: модель, цвет, текущие номера, дата постановки на учет, дата снятия с учета, номера двигателя и кузова, владелец, срок последнего техосмотра. Также имеется список автомобилей, находящихся в розыске. При постановке и снятии с учета проверяется, чтобы автомобиль не был в розыске.

При выдаче водительского удостоверения в базу заносится: номер удостоверения, дата выдачи, категория, ФИО, дата рождения, паспортные данные, прописка, фотография, вождение в очках. Также в базе хранится информация обо всех дорожных авариях, вовлеченных в них автомобилях и наложенных штрафах.

Выходные документы:

- для указанного владельца машины вывести все дорожные происшествия, произошедшие с его машинами (машиной) за последние N месяцев;
- список автомобилей с просроченной датой техосмотра.

Содержание отчёта:

1. Титульный лист.
2. Цель работы.
3. Задание.
4. Модель «сущность-связь».
5. Примеры выполнения запросов.

Контрольные вопросы

1. Из каких фаз состоит анализ предметной области?
2. Как выбираются информационные объекты?
3. Как выявляются связи между объектами?
4. Что такое ключ?
5. Что включает концептуальная модель?
6. Что понимается под сущностью?
7. Что такое атрибут?

Библиографический список

1. Киселев, Г.М. Информационные технологии в педагогическом образовании : учебник / Г.М. Киселев, Р.В. Бочкова. – 3-е изд., стер. – Москва : Дашков и К°, 2020. – 304 с. : ил. – (Учебные издания для бакалавров). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=573270> (дата обращения: 07.02.2021).
2. Лёвкина (Вылегжанина), А.О. Компьютерные технологии в научно-исследовательской деятельности: учебное пособие для студентов и аспирантов социально-гуманитарного профиля : [16+] / А.О. Лёвкина (Вылегжанина). – Москва ; Берлин : Директ-Медиа, 2018. – 119 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=496112> (дата обращения: 07.02.2021).
3. Шандриков, А.С. Информационные технологии : учебное пособие : [16+] / А.С. Шандриков. – 3-е изд., стер. – Минск : РИПО, 2019. – 445 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=463339> (дата обращения: 07.02.2021).
4. Калугян, К.Х. Информатика. Информационные технологии и системы : учебное пособие : [16+] / К.Х. Калугян ; Ростовский государственный экономический университет (РИНХ). – Ростов-на-Дону : Издательско-полиграфический комплекс РГЭУ (РИНХ), 2018. – 80 с. : схем., табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=567017> (дата обращения: 07.02.2021).
5. Компьютерные технологии в научных исследованиях : учебное пособие / Е.Н. Косова, К.А. Катков, О.В. Вельц и др. ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2015. – 241 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=457395> (дата обращения: 07.02.2021). – Текст : электронный.