Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ставропольский государственный аграрный университет»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению практических занятий
по дисциплине «Компьютерные, сетевые и информационные технологии»

для магистров направления подготовки
35.04.06 Агроинженерия
Направленность (профиль)
«Электрооборудование и электротехнологии в сельском хозяйстве»

Ставрополь, 2022

Методические указания разработаны в соответствии с требованиями ФГОС ВО в части содержания и уровня подготовки выпускников направления подготовки 35.04.06 Агроинженерия**.**

В методических указаниях даны рекомендации по организации практических занятий студента при изучении дисциплины «*Компьютерные, сетевые и информационные технологии*» при подготовке ко всем видам занятий, сделаны указания на моменты, требующие особого внимания, определен порядок подготовки и сдачи экзамена и приведен список рекомендуемых литературных источников.

Составитель к-т. экон. наук, доцент Сосин А.И.

Содержание

1. ВВЕДЕНИЕ 3
2. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 4
3. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 7
4. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3 11
5. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 23
6. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 27
7. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №6 33
8. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7 38
9. ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8 44
10. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ 46

Введение

Дисциплина «Компьютерные, сетевые и информационные технологии» изучается студентами направления подготовки 35.04.06 Агроинженерия по профилю подготовки «Электрооборудование и электротехнологии в сельском хозяйстве». Правильное сочетание теоретических знаний с практикой обеспечивает высокое качество подготовки выпускников.

В результате освоения теоретического и практического материала соответствующих разделов дисциплины «Компьютерные, сетевые и информационные технологии» образовательной программы студент приобретает следующие компетенции (ПК-4):

Знать: современные методы научных исследований в области проектирования информационных систем и технологий управления промышленным предприятием.

Уметь: осуществлять анализ и выбор инструментария проектирования и управления информационными системами для решения прикладных задач.

Владеть: навыками применения современного программного и методического инструментария для решения задач проектирования информационных систем и технологий управления промышленным предприятием.

Практическое занятие № 1

Рассылка почты при помощи «Ассистента слияния»

Цель работы: научить пользователя быстрым и экономичным способам рассылки почты.

Основные понятия

Если вам необходимо разослать большое количество писем, которые отличаются только несколькими словами, например, во вступлении одного: Уважаемый господин Иванов, а в другом: Уважаемая госпожа Голубка, то вы можете использовать для автоматизации этого процесса *Ассистент слияния* – специальное диалоговое окно, с помощью которого можете напечатать персональные письма и список имен, фамилий и адресатов получателей.

Принцип слияния

Индекс
Город
Организация
Должность
Фамилия
Имя
Отчество

Приглашаем Вас принять участие в конференции « Проблемы экологии в мировом сообществе», которая состоится с 1 по 7 августа 2007 года в здании Президиума Российской Академии Наук.

Заявки на участие должны быть направлены в Оргкомитет не позднее 1 мая 2004 г. по адресу : Москва, Кутузовский проспект 32а.

Председатель Оргкомитета
академик РАН А.И. Петровский

Рис.1. 1. Шаблон письма

На рис.1.1 вы видите письмо, в котором содержится приглашение на конференцию, и, которое следует разослать по многим адресам. При таком количестве создание отдельных персональных писем становится очень трудной задачей. Но с помощью функции слияние эта задача решается очень быстро. Что же еще требуется для создания тиража персональных писем?

117526, Москва, Институт Международных Отношений, профессор, Петров Анатолий Борисович.
117411, Москва, МГУ, доцент, Кириллов Иван Ильич.
141700, Долгопрудный, ЦАО, снс, Иванов Алексей Александрович.
312413, Новосибирск, НГУ, доцент, Брусилова Наталья Леонидовна.
617321, Киев, КГУ, профессор, Иванчук Георгий Ефимович.
411732,Обнинск, снс, Калягина Ирина Федоровна.

Рис. 1.2. Источник данных

Перед началом печати персональных писем вы должны создать два разных файла: один – с шаблоном письма, другой – с данными для подстановки в текст шаблона именами, фамилиями, адресами и т.п. (рис.1.2.)

Задание к практическому занятию

1. Создайте текст письма – приглашения (основной документ).

2. Создайте файл с адресами корреспондентов (источника данных).

3. Вставьте в основной текст поля, которые будут заменяться в процессе слияния на нужные слова, например на имена и фамилии.

4. Осуществите слияние и создайте тираж персональных писем.

Пример выполнения работы

1.Создание основного документа.

1.1. Из основного меню выберите *Рассылки→Начать слияние→Пошаговый мастер слияния.*

1.2. В окне *Выбор типа документа* укажите *Письма* и нажмите кнопку «Далее».

1.3. Наберите текст основного письма (рис.1.1).

1.4. В окне *Выбор документа* укажите *Текущий документ* и нажмите кнопку «Далее».

1.5. Сохраните файл под именем Основной документ.

2. Создание структуры источника данных.

2.1. В окне *Выбор получателей* выберите *Создание списка* и нажмите кнопку «Создать».

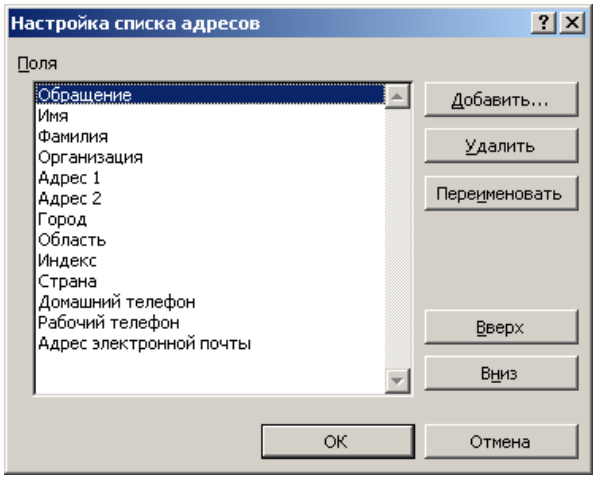


Рис.1.3. Создание списка получателей

2.2. В открывшемся окне *Новый список адресов* нажмите кнопку «Настройка столбцов».

2.3. В окне *Настройка списка адресов* удалите лишние поля и добавьте нужные.

В нашем примере должны быть следующие поля: ***индекс, город, организация, должность, фамилия, имя, отчество, пол.*** После того, как вы добавите нужные поля и удалите ненужные, упорядочьте расположение полей с помощью кнопки «Вверх» и нажмите кнопку «Ok».

2.4. Введите данные в нужные поля (рис. 1.3). В поле *Пол* введите символы **М** для лиц мужского пола, **Ж** для всех лиц женского пола.

2.4. Сохраните файл под именем Источник данных.

3. Вставка полей слияния.

3.1. Установите текстовый курсор в том месте письма, где вы хотите расположить индекс получателя, оставив пробел после слова ***Индекс.***

Щелкните по кнопке «Вставить поле слияния» и выберите из открывшегося списка поле *Индекс.*

Установите текстовый курсор на следующую строку. Щелкните по кнопке «Вставить поле слияния» и выберите из списка поле *Город.* Таким образом, выставьте все необходимые поля (поле *Пол* выставлять не надо – оно необходимо только для формирования обращения к адресату).

4. Вставка условий в текс письма.

4.1. *Примечание.* Чтобы в тексте письма появилось обращение к адресату Уважаемая или Уважаемый, вставим в текст условие IF-THEN-ELSE и будем проверять пол адресата.

Установите текстовый курсор в начале строки, в которой должно находится обращение к адресату. Щелкните по кнопке «Правила» и затем выберите из списка строку IF-THEN-ELSE.

Появится диалоговое окно. В окне *Поле* из списка полей выберите слово ***Пол***.

В окне *Оператор* установите ***Равно***.

В окне *Значение* введите **М**.

В окне *Вставить* следующий текст введите: **Уважаемый.**

Уважаемая.

Нажмите кнопку «Ok». Нажмите клавишу <Пробел> после вставленного обращения. Щелкните по кнопке «Вставить поле слияния». Выберите из списка поле *Имя***.** Нажмите клавишу <Пробел>. Щелкните по кнопке «Вставить поле слияния». Выберите из списка поле *Отчество.* Поставьте восклицательный знак.

5. Слияние документов.

5.1. Щелкните по кнопке «Автопоиск ошибок» (рис. 1.4., на кнопке нарисовано письмо с загнутым уголком и галочкой). На экране появится окно.

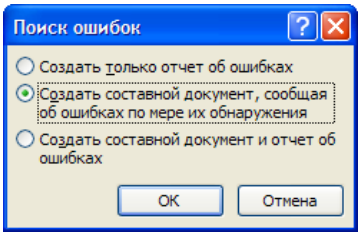


Рис.1.4. Поиск ошибок

Отметьте в нем пункт *Создать составной документ, сообщая об ошибках по мере обнаружения.*

Нажмите кнопку «Ok».

После обработки WORD создаст новое окно документа, в котором будут находиться все 5 писем, написанных различным адресатам, созданным в результате слияния.

Контрольные вопросы

1. Для чего используется Ассистент слияния.
2. Какие файлы следует создать для рассылки серийной корреспонденции.
3. Для чего используется кнопка «Вставить поле слияния».
4. Как удалить лишние поля и добавить новые в окне *Новый список адресов.*
5. Назовите кнопки панели *Пошагового* мас*тера слияния***.**

Практическое занятие № 2

Поиск решения

Цель работы: научиться использовать разные методы решения уравнений или различные задачи оптимизации в Microsoft Excel.

Основные понятия

Настройка **Поиск решения** позволяет использовать разные методы решения уравнений или различных задач оптимизации. Это могут быть, например:

определение ассортимента производимой продукции в целях максимизации прибыли;

определение маршрута перевозок между транспортными узлами в целях минимизации транспортных издержек;

формирование портфеля ценных бумаг;

проектирование электрических цепей;

составление соответствующего плана бесперебойного производства и т.д.

Диалоговое окно вызывается командой меню ***Данные*** ➔ ***Поиск решения.***

Элементы диалогового окна " Поиск решения". ***Оптимизировать целевую функцию.*** Служит для указания целевой ячейки, значение которой необходимо максимизировать, минимизировать или установить равным заданному числу. Эта ячейка должна содержать формулу.

До. Служит для выбора варианта оптимизации значения целевой ячейки (максимум, минимум, значения). Чтобы установить число, введите его в поле.

Изменяя ячейки переменных. Служит для указания ячеек, значения которых изменяются в процессе поиска решения до тех пор, пока не будут выполнены наложенные ограничения и условие оптимизации значения ячейки, указанной в поле *Оптимизировать целевую функцию.*

Ограничение. Служит для отображения списка граничных условий поставленной задачи.

Добавить. Служит для отображения диалогового окна *В соответствии с ограничением.*

Изменить. Служит для отображения диалогового окна *Изменить ограничение.*

Удалить. Служит для снятия указанного ограничения.

Найти решение. Служит для запуска поиска решения поставленной задачи.

Параметры. Служит для отображения диалогового окна *Параметры поиска решения,* в котором можно загрузить или сохранить оптимизируемую модель и указать предусмотренные варианты поиска решения.

Задание к практическому занятию

Имеются три производственных предприятия, которые находятся в Орле, Белгороде, и Воронеже. Ежемесячные производственные возможности (мощности) составляют соответственно: 7 000, 3 000, 5 000 шт. Ежемесячная потребность складов, которые находятся в Липецке, Тамбове, Смоленске и Туле, составляют соответственно: 2 000, 3 000, 4 000, 5 000 шт. Издержки по перевозке с предприятий в расчете на 1 единицу продукции составляют:

	Липецк	Тамбов	Смоленск	Тула
Орел	2	1	4	6
Белгород	7	9	3	8
Воронеж	2	5	8	4

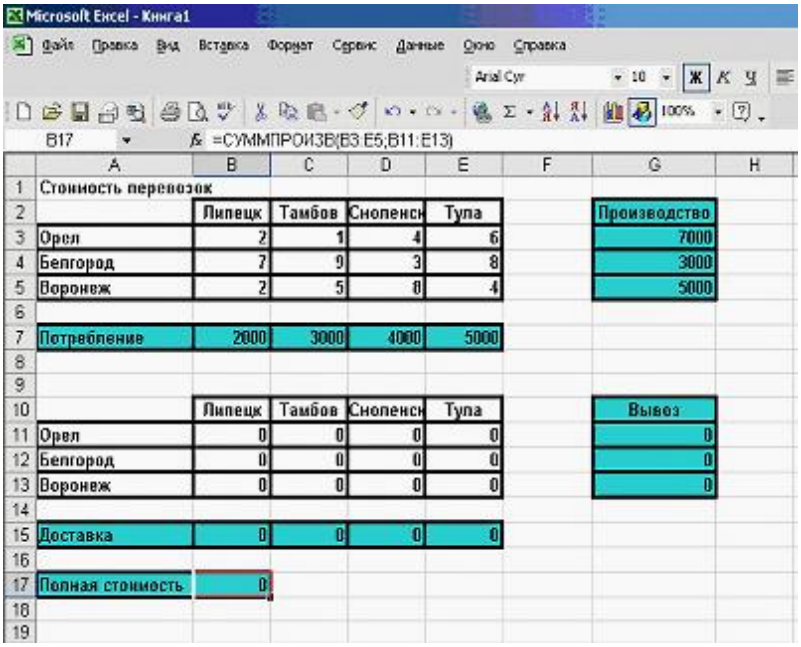
Задача заключается в выборе маршрутов таким образом, чтобы суммарные издержки перевозки товаров от производственных предприятий до складов были бы минимальными. Одновременно следует помнить, что невозможно превысить производственные мощности каждого завода и необходимо удовлетворить потребности каждого склада.

Пример выполнения работы

Вышеуказанную задачу мы можем представить в следующем виде:

У нас две таблицы (рис.2.1). Одна представляет расходы на единицу изделия, вторая – количество товара, которое будет доставлено из предприятия на склад. Кроме того, в ячейках:

- от G3 до G5 определен максимально возможный объем производства;
- от B7 до E7 определено требуемое потребление товара;
- от G11 до G13 находится формула, суммирующая количество в штуках, вывезенное из отдельных предприятий (например: в G11 формула «=СУММ(B11:E11»);



	A	B	C	D	E	F	G	H
1	Стоимость перевозок							
2		Липецк	Тамбов	Смоленск	Тула		Производство	
3	Орел	2	1	4	6		7000	
4	Белгород	7	9	3	8		3000	
5	Воронеж	2	5	8	4		5000	
6								
7	Потребление	2000	3000	4000	5000			
8								
9								
10		Липецк	Тамбов	Смоленск	Тула		Вывоз	
11	Орел	0	0	0	0		0	
12	Белгород	0	0	0	0		0	
13	Воронеж	0	0	0	0		0	
14								
15	Доставка	0	0	0	0			
16								
17	Полная стоимость	0						
18								
19								

Рис. 2.1. Расходы на единицу изделия, количество товара, которое будет доставлено из предприятия на склад

– от B15 до E15 находится формула, суммирующая количество в штуках, доставленное на отдельные склады (например: в ячейке B15 находится формула «=СУММ(B11:B13)»);

– в B17 определена функция цели, т.е. сумма произведений расходов на перевозку и объема перевозок по каждому возможному маршруту (формула «=СУММПРОИЗВ(B3:E5;B11:E13))».

А сейчас загрузите надстройку **Поиск решения,** вызывая одноименную команду из меню *Данные***,** и введите соответствующие параметры:

– в поле *Установить целевую функцию* мы указываем адрес итоговой ячейки, которая должна содержать формулу (в нашем случае $B$17);

– в поле *До* устанавливаем переключатель Минимум (что означает минимизацию функции цели);

– в поле *Изменяя ячейки* переменных указываем адреса каждой изменяемой ячейки: это может быть диапазон или разделенные запятыми адреса несмежных ячеек (для нашей задачи – это совокупность ($B$11:$E$13). Именно в этих ячейках мы должны получить результат решения.

– В поле *В соответствии с ограничениями* мы указываем все условия, наложенные на нашу задачу, т.е. нельзя превышать производственные возможности каждого из промышленных предприятий ($G$11:$G13:<=$G$3:$G$5), следует удовлетворить потребность каждого склада ($B$15:$E$15>$B$7:$E$7), величина объема перевозок не может быть отрицательной ($B$11:$E$13>=0).

– При параметрах, определенных таким образом, задачу можно решить, нажав кнопку «Найти решение» (рис.2.2).

В последующем окне (рис.2.3) выведена информация о результате поиска решения. Если найдено оптимальное решение, в верхней части окна видна надпись **Решение найдено**, а после нажатия кнопки «ОК» происходит актуализация данных в электронной таблице, если переключатель установлен в позицию **Сохранить найденное решение**, или возврат к исходным значениям, если включена опция **Восстановить исходные значения.**

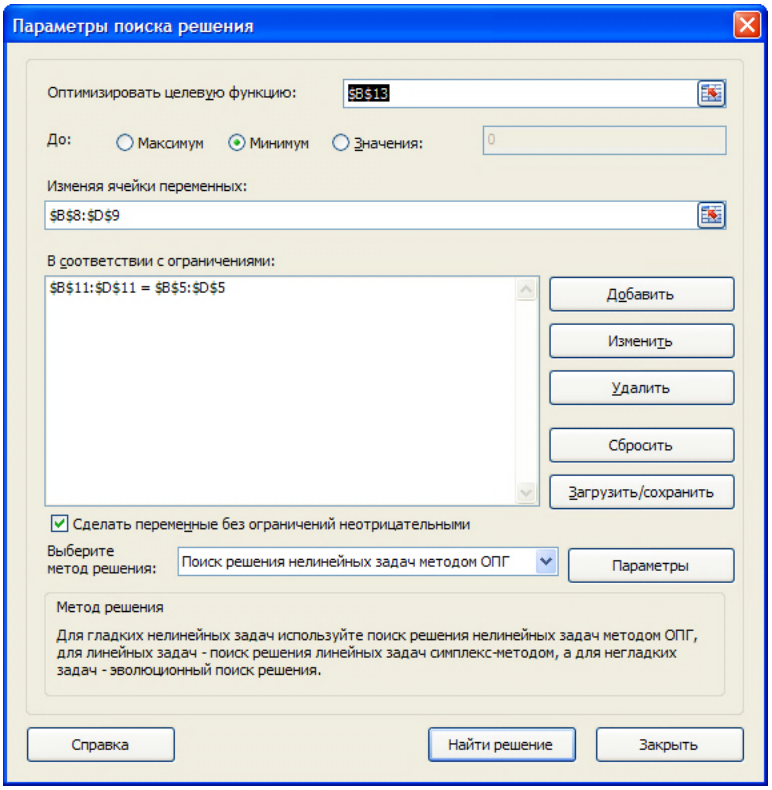


Рис. 2.2. Окно Поиск решения

Кнопка **«Отмена»** вызывает состояние, когда результат не сохраняется, а работа надстройки **Поиск решения** будет завершена. Нажатие кнопки **«Сохранить сценарий»** вызывает сохранение исходных величин, которые позднее можно будет ввести в электронную таблицу.

Ели будет найдено решение, можно генерировать три отчета: **Результат, Устойчивость,** а также **Пределы**. Они позволяют проанализировать результаты (например: можно ответить на вопрос, изменился ли график транспорта, если стоимость единицы продукции из Белгорода до Липецка вырастет в два раза).

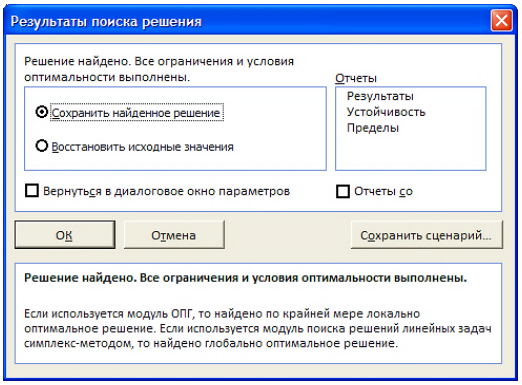


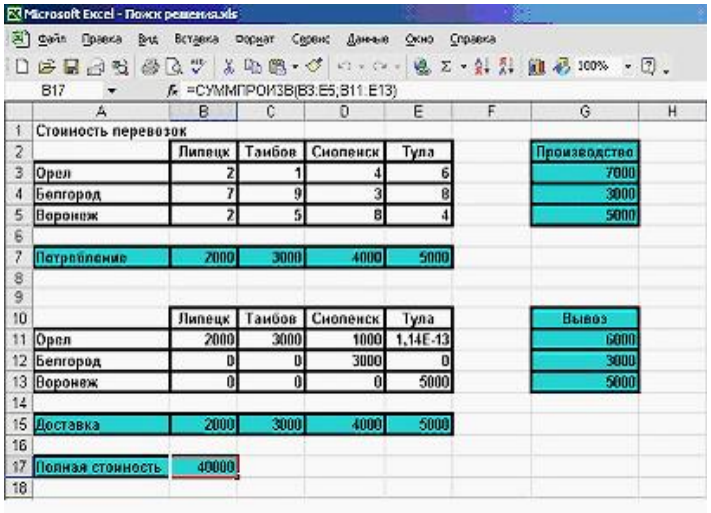
Рис. 2.3. Окно Результаты поиска решения

Оптимальное решение нашей задачи выглядит следующим образом (рис.2.4):

Перевозим из Орла до Липецка 2 000 шт., из Орла до Тамбова 3 000 шт., из Орла до Смоленска – 1 000 шт.

Полные расходы на транспортировку товара составили 40 000 руб.

Из Белгорода транспорт идет только до Смоленска (3 000 шт.), а Тула снабжается из Воронежа (5 000 шт.



	A	B	C	D	E	F	G	H
1	Стоимость перевозок							
2		Липецк	Тамбов	Смоленск	Тула		Производство	
3	Орел	2	1	4	6		7000	
4	Белгород	7	9	3	8		3000	
5	Воронеж	2	5	8	4		5000	
6								
7	Потребление	2000	3000	4000	5000			
8								
9								
10		Липецк	Тамбов	Смоленск	Тула		Вывоз	
11	Орел	2000	3000	1000	1,14E-13		6000	
12	Белгород	0	0	3000	0		3000	
13	Воронеж	0	0	0	5000		5000	
14								
15	Доставка	2000	3000	4000	5000			
16								
17	Полная стоимость	40000						
18								

Рис. 2.4. Таблица с результатами найденного решения

Контрольные вопросы

1. Какие задачи позволяет решать надстройка **Поиск решения?**
2. Какая функция называется целевой?
3. Зачем служит элемент **Равно** диалогового окна **Поиск решения**?
4. Какой метод поиска был применен для решения задачи оптимизации?
5. Зачем служит элемент **Предположить** диалогового окна **Поиск решения?**
6. Какие типы отчетов возможны в результате найденного решения?
7. Как в работе определяется функция цели?

Практическое занятие №3

Microsoft Excel. Пакет анализа

Цель работы: с помощью программы Excel научиться анализировать экспериментальные данные с использованием метода наименьших квадратов.

Основные понятия

На практике часто приходится сталкиваться с задачей о сглаживании экспериментальных зависимостей или задачей аппроксимации. Аппроксимацией называется процесс подбора эмпирической формулы $\varphi(x)$ для установленной из опыта функциональной зависимости $y = f(x)$. Эмпирические формулы служат для аналитического представления опытных данных.

Одна независимая переменная. Обычно задача аппроксимации распадается на две части. Сначала устанавливают вид зависимости $y = f(x)$ и, соответственно, вид эмпирической формулы, то есть решают, является ли она линейной, квадратичной, логарифмической или какой-либо другой. После этого определяются численные значения неизвестных параметров выбранной эмпирической формулы, для которых приближение к заданной функции оказывается наилучшим. Если нет каких-либо теоретических соображений для подбора вида формулы, обычно выбирают функциональную зависимость из числа наиболее простых, сравнивая их графики с графиком заданной функции.

После выбора вида формулы определяют ее параметры. Для наилучшего выбора параметров задают меру близости аппроксимации экспериментальных данных. Во многих случаях, в особенности, если функция $f(x)$ задана графиком или таблицей (на дискретном множестве точек), для оценки степени приближения рассматривают разности $f(x_i) - \varphi(x_i)$ для точек $x_0, x_1, \ldots x_n$. Существуют различные меры близости и, соответственно, способы решения этой задачи. Некоторые из них очень просты, быстро приводят к результату, но результат является сильно приближенным. Другие более точные, но и более сложные. Обычно определение параметров, при известном виде зависимости осуществляют по методу наименьших квадратов. При этом функция $\varphi(x)$ считается наилучшим приближением к $f(x)$, если для нее сумма квадратов невязок δ_i или отклонений «теоретических» значений $\varphi(x_i)$ найденных по эмпирической формуле, от соответствующих опытных значений y

$$z = \sum_{i=0}^{n} [f(x_i) - \varphi(x_i)]^2 \to \min \qquad (3.1)$$

Имеет наименьшее значение по сравнению с другими функциями, из числа которых выбирается искомое приближение.

Используя методы дифференциального исчисления, метод наименьших квадратов формулирует аналитические условия достижения суммой квадратов отклонений (3.1) своего наименьшего значения. Так, если функция $\varphi(x)$ вполне определяется своими параметрами k,l,m,…, то наилучшие (в указанном смысле (3.1)) значения этих параметров находятся из решения системы уравнений. Например, в простейшем случае, когда функция $\varphi(x)$ представлена линейным уравнением $y = ax + b$, система имеет вид:

$$\begin{aligned} a \cdot \sum_{i=1}^{n} x_i^{\,2} + b \cdot \sum_{i=1}^{n} x_i &= \sum_{i=1}^{n} x_i \cdot y_i, \\ a \cdot \sum_{i=1}^{n} x_i + b \cdot n &= \sum_{i=1}^{n} y_i. \end{aligned} \qquad (3.2)$$

В простейшем случае задача аппроксимации экспериментальных данных выглядит следующим образом.

Пусть есть какие-то данные, полученные практическим путем (в ходе эксперимента или наблюдения), которые отображает таблица:

x	$x1$……	x_n
y	$y1$……	y_n

На основе этих данных требуется подобрать функцию $y = \varphi(x)$, которая наилучшим образом сглаживала бы экспериментальную зависимость между переменными и по возможности точно отражала общую тенденцию зависимости между x и y, исключая погрешности измерений и случайные отклонения. Это значит, что отклонения $y_i - y_i(x_i)$ в каком-то смысле были бы наименьшими. Например, в смысле формулы (3.1).

Выяснить вид функции можно либо из теоретических соображений, либо анализируя расположение точек $(x_i; y_i)$ на координатной плоскости. Например, пусть точки расположены так, как показано на рис.3.1, 3.2.

Учитывая то, что практические данные получены с некоторой погрешностью, обусловленной неточностью измерений, необходимостью округления результатов и т.п., естественно предположить, что здесь имеет место линейная зависимость $y = a \cdot x + b$.

Чтобы функция приняла конкретный вид, необходимо каким-то образом вычислить a и b. Для этого можно решить систему (3.2).

Расположение экспериментальных точек в виде кривой на рис. 3.1. наводит на мысль, что зависимость обратно пропорциональна и функцию $\varphi(x)$ нужно подбирать в виде $y = a + b/x$. Здесь также необходимо вычислить параметры a и b.

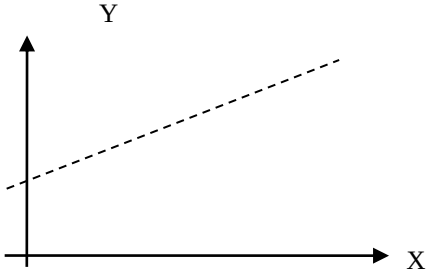


Рис. 3.1. Возможный вариант расположения экспериментальных точек

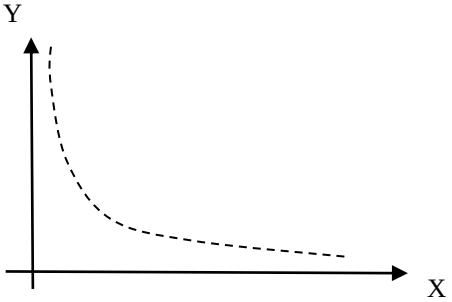


Рис. 3.2. Другой вариант расположения экспериментальных точек

Таким образом, расположение экспериментальных точек может иметь самый различный вид, и каждому соответствует конкретный тип функции.

Построение эмпирической функции сводится к вычислению входящих в нее параметров, так чтобы из всех функций такого вида выбрать ту, которая лучше других описывает зависимость между изучаемыми величинами. То есть сумма квадратов описывает зависимость между табличными значениями функции в некоторых точках и значениями, вычисленными по полученной формуле, должна быть минимальна.

В MS Excel аппроксимация экспериментальных данных осуществляется путем построения их графика (x - отвлеченные величины) или точечного графика (x - имеет конкретные значения) с последующим подбором подходящей аппроксимирующей функции (линии тренда). Возможны следующие варианты функций:

1. ***Линейная – y = ax + b.*** Обычно применяется в простейших случаях, когда экспериментальные данные возрастают или убывают с постоянной скоростью.

2. ***Полиномиальная – $y=a_0 + a_1 x + a_2 x^2 + \ldots\ldots + a_n x^n$,*** где n изменяется до шестого порядка включительно (n<=6), a_i –константы. Используется для описания экспериментальных данных, попеременно возрастающих и убывающих. Степень полинома определяется количеством экстремумов (максимумов или минимумов) кривой. Полином второй степени можно описать только один максимум или минимум, полином третьей степени может иметь один или два экстремума, четвертой степени – не более трех экстремумов и т.д.

3. ***Логарифмическая – y = a lnx + b***, где a и b – константы, ln –функция натурального логарифма. Функция применяется для описания экспериментальных данных, которые вначале быстро растут или убывают, а затем постепенно стабилизируются.

4. ***Степенная – $y=bx^n$*** , где a и b – константы. Аппроксимация степенной функцией используется для экспериментальных данных с постоянно увеличивающейся (или убывающей) скоростью роста. Данные не должны иметь нулевых или отрицательных значений.

5. ***Экспоненциальная – $y=be^{ax}$***, где a и b – константы, e – основание натурального логарифма. Применяется для описания экспериментальных данных, которые быстро растут или убывают, а затем постепенно стабилизируются. Часто ее использование вытекает из теоретических соображений.

Степень близости аппроксимации экспериментальных данных выбранной функцией оценивается коэффициентом детерминации (R^2). Таким образом, если есть несколько подходящих вариантов типов аппроксимирующих функций, можно выбрать функцию с большим коэффициентом детерминации (стремящимся к 1).

Для осуществления аппроксимации на диаграмме экспериментальных данных необходимо щелчком правой кнопки мыши вызвать выплывающее контекстное меню и выбрать пункт *Добавить линию тренда.* В

появившемся диалоговом окне *Линия тренда* на вкладке *Тип* выбирается вид аппроксимирующей функции, а на вкладке *Параметры* задаются дополнительные параметры, влияющие на отображение аппроксимирующей кривой.

Пример 1. Исследовать характер изменения с течением времени уровня производства некоторой продукции и подобрать аппроксимирующую функцию, располагая следующими данными:

Год	Производство продукции
2011	17,1
2012	18,0
2013	18,9
2014	19,7
2015	19,7

Решение

1. Для построения диаграммы, прежде всего, необходимо ввести данные в рабочую таблицу. Вводим в ячейку A1слово *Год.* Затем в ячейки A2:A6 последовательно вводим годы, начиная с 2010. Далее в ячейку B1 заносим слово *Продукция* и устанавливаем табличный курсор в ячейку B2. Здесь должно оказаться значение 17,1 соответствующее значению года в ячейке A2. Аналогично заполняем ячейки B3:B6.

2. Далее по введенным в рабочую таблицу данным необходимо построить диаграмму. Поскольку здесь необходимо строить динамику изменений производства продукции, не привязываясь к конкретному году, а от отвлеченных переменных – выберем диаграмму *График.*

Щелчком указателя мыши на кнопке на панели инструментов вызываем *Мастер диаграмм*. В появившемся диалоговом окне выбираем тип диаграммы *График*, вид - *Левый средний.* После нажатия кнопки <Далее> указываем диапазон данных B1:B6(с помощью мыши). Проверяем положение переключателя *Ряды в:* столбцах. Выбираем вкладку *Ряд* и с помощью мыши вводим диапазон подписей оси X:A2:A6. Нажав кнопку <Далее>, вводим название диаграммы – *Производство продукции*, название осей X и Y: *Годы* и *Условные единицы*, соответственно. Нажимаем кнопку <Готово>. Получен график экспериментальных данных.

3. Осуществим аппроксимацию полученной кривой полиноминальной функцией второго порядка, поскольку кривая довольно гладкая и не сильно отличается от прямой линии. Для этого указатель мыши устанавливаем на одну из точек графика и щелкаем правой кнопкой. В появившемся контекстном меню выбираем пункт *Добавить линию тренда*. Появляется диалоговое окно *Линия тренда* (рис. 3.3).

В этом окне на вкладке *Тип* выбираем тип линии тренда – *Полиноминальная* и устанавливаем степень – 2. Затем открываем вкладку *Параметры* (рис.3.4.) и устанавливаем флажки в поля показывать уравнение на диаграмме и поместить на диаграмму величину достоверности аппроксимации (R^2). После чего нужно щелкнуть на кнопке <Ok>.

В результате получим на диаграмме аппроксимирующую кривую (рис.3.4). Как видно из рисунка, уравнение наилучшей полиномиальной аппроксимирующей функции для некоторых отвлеченных значений x (1,2,3, …) выглядит как

$$y = -0{,}14x^2 + 1{,}5x + 15{,}6$$

При этом точность аппроксимации достаточно высока –R^2=0,986.

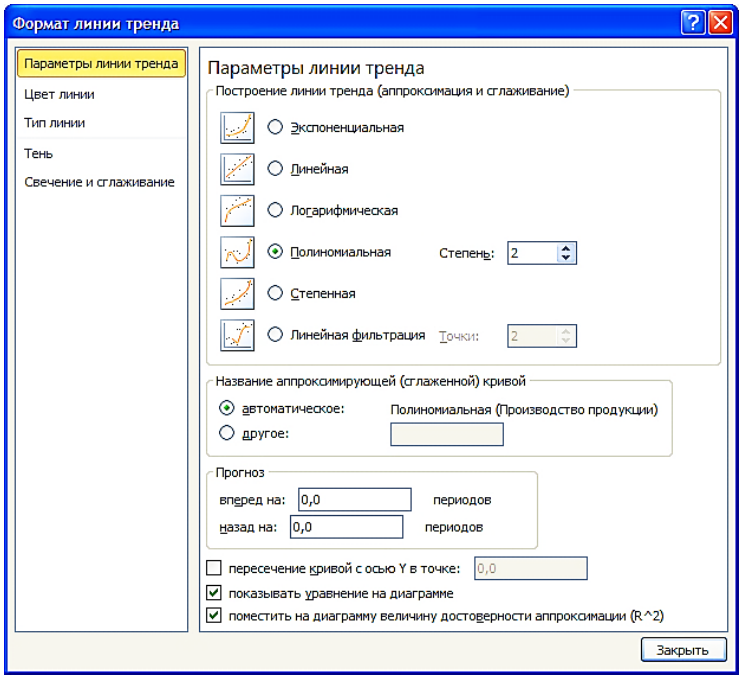


Рис. 3.3. Вкладка параметры диалогового окна Линия тренда.

Рис. 3.4. Экспериментальные данные, аппроксимированные полиномиальной кривой, из примера 1

4. Попробуем улучшить качество аппроксимации выбором другого типа функции (возможно более адекватного). Здесь возможным вариантом представляется логарифмическая функция. Для этого повторяем операции п.3. За исключением того, что в окне *Линия Тренда* на вкладке *Тип* выбираем тип линии тренда – *Логарифмическая.*

В результате получим другой вариант аппроксимации – логарифмической кривой (рис. 3.5).

Как можно видеть из рис. 3.5, уравнение наилучшей логарифмической аппроксимирующей функции несколько уступает по точности аппроксимации полиномиальной кривой – R^2=0,9716<0,986. Поэтому, если нет каких-либо теоретических соображений, то можно считать, что наилучшей аппроксимацией является аппроксимация полиномиальной функцией второй степени (из двух рассмотренных вариантов).

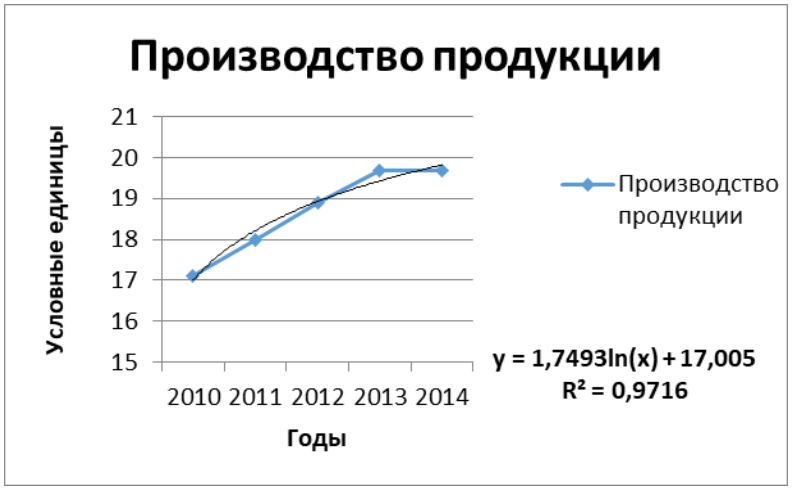


Рис. 3.5. Экспериментальные данные, аппроксимированные логарифмической кривой

Пример 2. После выброса ядовитого вещества его концентрация (мг/л) в водоеме изменялась в соответствии со следующей таблицей:

Время после выброса,ч	Концентрация вещества, мг/л
1	8,0
3	2,8
5	1,0
8	0,3

Определить вид функциональной зависимости изменения концентрации вещества от времени и оценить его концентрацию в водоеме в момент выброса.

Решение

1. Для построения диаграммы, прежде всего, необходимо ввести данные в рабочую таблицу. Вводим в ячейку A1 слово Время. Затем в ячейки A2:A5 последовательно вводим время 1,3,5,8. Далее в ячейку B1заносим слово Концентрация и в диапазон B2:B5 вводим соответствующие концентрации вещества.

2. Далее по введенным в рабочую таблицу данным необходимо построить диаграмму. Поскольку здесь необходимо строить динамику изменений концентрации вещества в соответствии с изменениями времени – будем строить диаграмму *График с маркерами*.

В меню выбираем команду *Вставка → График → График с маркерами.*

3. Осуществим аппроксимацию полученной кривой. Поскольку кривая напоминает экспоненту и из теоретических соображений наиболее вероятный закон изменения - экспоненциальный, целесообразно аппроксимировать кривую изменения концентрации экспоненциальной функцией. Для этого указатель мыши устанавливаем на одну из точек графика и щелкаем правой кнопкой. В появившемся контекстном меню выбираем пункт Добавить линию тренда. Появляется диалоговое окно *Линия тренда.*

В этом окне на вкладке *Тип* выбираем тип линии тренда – *Экспоненциальная.* Затем открываем вкладку *Параметры* и устанавливаем флажки в поля *Показать уравнение на диаграмме* и *Поместить на диаграмму величину достоверности аппроксимации (R^2).* Кроме этого, для того, чтобы оценить концентрацию вещества в водоеме в момент выброса в поле *Прогноз назад на* устанавливаем 1 периодов. После чего щелкаем на кнопке «Ok». В результате получим на диаграмме аппроксимирующую кривую (рис.3.6).

Как видно из рис. 3.6, уравнение наилучшей экспоненциальной аппроксимирующей функции для зависимости концентрации от времени выглядит как

$$y = 24{,}44e^{-1{,}088x} \qquad (3.3)$$

При этом точность аппроксимации очень высокая – R^2=0,9951, что позволяет считать описание процесса изменения концентрации вещества в водоеме экспоненциальной функцией адекватным. Расчетная оценка концентрации вещества в момент выброса, как видно из графика, составляет около 12 мг/л. Более точные цифры могут быть получены из уравнения (3.3) при $x = 0$ ($y_0 = 11{,}84 мг.л$).

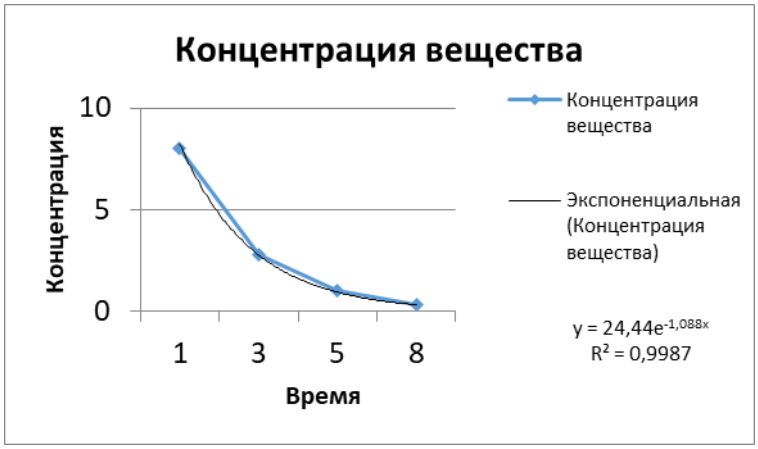


Рис. 3.6. Экспериментальные данные, аппроксимированные экспоненциальной функцией из примера 2 .

Задания для самостоятельной работы

1. В 2000-у годы уровень дефицита бюджета в России и США складывались следующим образом:

	Годы							
Страна	2008	2009	2010	2011	2012	2013	2014	2015
Россия	2,9	2,3	3,1	2,2	2,0	2,7	6,5	8,0
США	2,8	2,6	4,1	6,3	5,0	5,4	5,3	3,4

Построить функции, наилучшим образом отражающие зависимость дефицита бюджета от времени в обеих странах

2. Количество вложенных в производство средств и полученная в результате прибыль соотносятся следующим образом:

x	1,6	2,0	2,5	3,0	4,0	7,0
y	8,5	9,0	11,0	13,0	22,0	70,0

Запишите аналитическую зависимость между x и y. Проанализируйте полученный ответ. Каковы перспективы предприятия? Какая будет прибыль, если вложить 10,0 единиц?

Сколько нужно вложить средств, чтобы получить прибыль 100,0 единиц?

Несколько независимых переменных. В тех случаях, когда аппроксимируемая переменная y зависит от нескольких независимых переменных $x_1,x_2,\ldots,x_n$,

$$y=f(x_1,x_2,\ldots x_n),$$

Подход с построением линии тренда не дает решения. Здесь могут быть использованы следующие специальные функции MS Excel:

ЛИНЕЙН и ТЕНДЕНЦИЯ для аппроксимации линейных функций вида:

$$y=a_0+a_1x_1+a_2x_2+\ldots+a_nx_n, \qquad (3.4)$$

ЛФПРИБЛ и РОСТ для аппроксимации показательных функций вида:

$$y=a_0a_1^{x1}a_2^{x2}\ldots a_n^{xn}. \qquad (3.5)$$

Функции ЛИНЕЙН и ЛГРФПРИБЛ служат для вычисления неизвестных коэффициентов $a_0,a_1,\ldots,a_n$, в выражениях (3.4) и (3.5) соответственно, а также коэффициентов детерминации (R^2), значений критерия Фишера, стандартных ошибок коэффициентов a_i и ряда других показателей.

Обе функции имеют одинаковые параметры:

ЛИНЕЙН (*известные _значения_y; известные _значения_x; конст; статистика*)

ЛГРФПРИБЛ(*известные _значения_y; известные _значения _x; конст; статистика*)

Здесь:

Известные _значения_*y* – множество наблюдаемых значений y из выражений (3.4),(3.5);

Известные *_значения_x* – множество наблюдаемых значений *x1,x2,…xn*. Причем, если массив *известные _значения _ y* имеет один столбец, то каждый столбец массива известные *_значения_x* интерпретируются как отдельная переменная, а если массив известные *_значения_*y имеет одну строку, то тогда каждая строка массива известные *_значения_x* интерпретируется как отдельная переменная;

Конст – логическое значение, которое указывает, требуется ли, чтобы константа a0 ,была равна 0 (для функции ЛИНЕЙН) или 1 (для функции ЛГРФПРИБЛ).

При этом, если конст имеет значение ИСТИНА или опущено, то a0 полагается обычным образом, а если конст имеет значение ЛОЖЬ, то a0 полагается 0 или 1;

Статистика – логическое значение, которое указывает, требуется ли вычислять дополнительную статистику по регрессии, если введено значение ИСТИНА, то дополнительные параметры вычисляются, если ЛОЖЬ, то – нет (рис. 3.7).

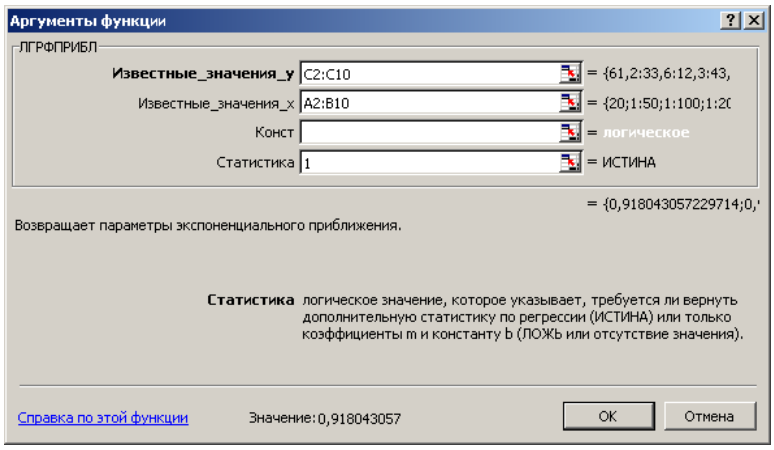


Рис. 3.7. Пример заполнения диалогового окна функции ЛГРПРИБЛ

Функции ТЕНДЕНЦИЯ и РОСТ позволяют находить точки, лежащие на аппроксимирующих кривых (3.4) и (3.5) , соответственно, для значений коэффициентов $\alpha_0,\alpha_1...\alpha_n$, найденных функциями ЛИНЕЙН и ЛГРФПРИБЛ.

Обе функции имеют одинаковые аргументы:

Тенденция (*известные _значения_y; известные _значения_x; новые _ значения _x; конст*);

Рост (*известные _значения_y; известные _значения _x; новые _значения_x; конст*).

Здесь:*Известные _значения_y* – множество значений y;

Известные _значения_x – множество значений x;

Новые _значения_x – те значения x, для которых необходимо определить соответствующие аппроксимирующие или предсказанные значения y.*Новые _значения_x* должны содержать столбец (или строку) для каждой независимой переменной, как и *известные _значения_x*. Если аргумент *новые _значения_x* опущен, то предполагается, что он совпадает с аргументом *известные _значения_x*;

Конст – логическое значение, которое указывает, требуется ли, чтобы константа a_0 , была равна 0 (для функции ТЕНДЕНЦИЯ) или 1 (функция РОСТ). При этом, если *конст* имеет значение ЛОЖЬ, то a_0 полагается равным 0 или 1 (рис. 3.8).

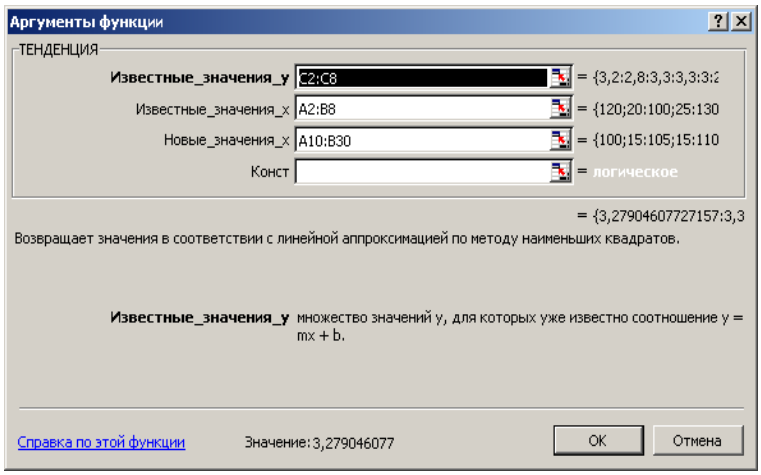


Рис. 3.8. Пример заполнения диалогового окна функции ТЕНДЕНЦИЯ

Пример 3. Источник радиоактивного излучения помещен в жидкость. Датчики расположены на расстоянии (x_1) 20,50, и 100 см от источника. Измерения интенсивности излучения (y, мРн) проводились через 1,5 и 10 суток(x_2) после установки источника. Результаты измерений (y) приведены в таблице:

x_1/x_2	1	5	10
20	61,2	43,6	23,3
50	33,6	24,0	15,6
100	12,3	8,8	5,7

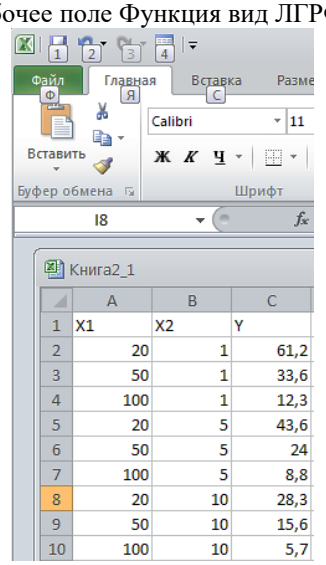
Необходимо аппроксимировать данные уравнением вида (3.5) и найти неизвестные параметры.

Решение

1. Введем данные в рабочую таблицу: в ячейку *A1* – символ x_1, в ячейку *B1* – x_2, в ячейку *C1* – *y*. В диапазон ячеек *A2:A10* внесем значения x_1, в диапазон *B2:B10* – значения x_2 и в диапазон *C2:C10* – значения *y* (рис.3.10).

2. Выделяем блок ячеек *D1:F5* под массив результатов.

3. Поскольку уравнение для вычисления интенсивности излучения имеет степенной характер (3.5) , вызываем функцию ЛГРФПРИБЛ (панель инструментов Стандартная, кнопка Вставка функции, рабочее поле Категория тип Статистические, рабочее поле Функция вид ЛГРФПРИБЛ).



	A	B	C
1	X1	X2	Y
2	20	1	61,2
3	50	1	33,6
4	100	1	12,3
5	20	5	43,6
6	50	5	24
7	100	5	8,8
8	20	10	28,3
9	50	10	15,6
10	100	10	5,7

Рис.3.9. Исходные данные из примера 3

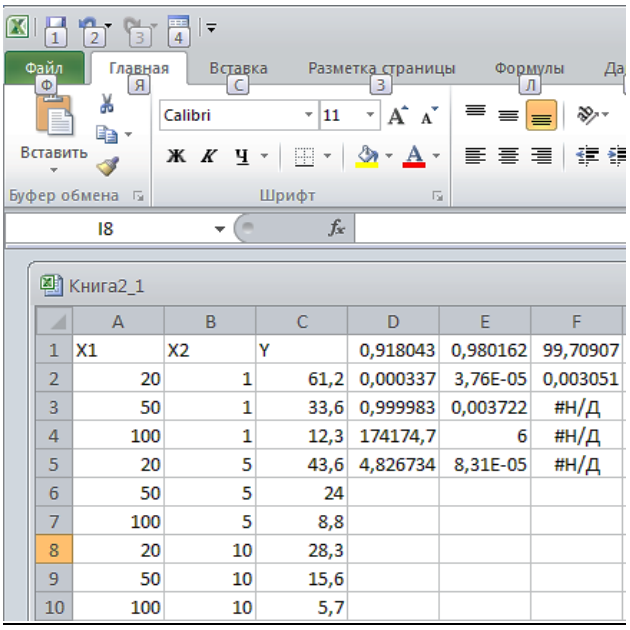
4. Заполняем рабочие поля: Изв_знач_y – *C2:C10*, Изв_знач_*x* – *A2:B10*, Стат – *1* (рис. 3.9). Нажимаем сочетание клавиш <Ctrl+Shift+Enter.>

5. В результате в диапазоне *D1:F5* получим следующие данные (рис.3.10):

Здесь первая строка – значения коэффициентов a_2,a_1,a_0, соответственно, вторая строка – стандартные ошибки этих коэффициентов, третья строка – коэффициент детерминации R^2 и стандартная ошибка y, четвертая строка – значение критерия Фишера и число степеней свободы и нижняя строка – сумма квадратов регрессии и остаточная сумма квадратов. Таким образом, искомое аппроксимирующее уравнение имеет вид:

$$y = 99{,}7 \cdot 0{,}98^{x1} \cdot 0{,}92^{x2} \qquad (3.6)$$

Причем точность аппроксимации очень высокая – R^2=0,99998.



	A	B	C	D	E	F
1	X1	X2	Y	0,918043	0,980162	99,70907
2	20	1	61,2	0,000337	3,76E-05	0,003051
3	50	1	33,6	0,999983	0,003722	#Н/Д
4	100	1	12,3	174174,7	6	#Н/Д
5	20	5	43,6	4,826734	8,31E-05	#Н/Д
6	50	5	24			
7	100	5	8,8			
8	20	10	28,3			
9	50	10	15,6			
10	100	10	5,7			

Рис. 3.10. Массив результатов из примера 3

Пример 4. В бассейне проводится ежедневная частичная смена воды. Имеются данные семидневных наблюдений изменения уровня воды в бассейне (*y)* от продолжительности заполнения водой (и времени выпуска воды (*x_2)*.

x1	x2	y
120	**20**	**3,2**
100	**25**	**2,8**
130	**20**	**3,3**
100	**15**	**3,3**
110	**23**	**3,0**
105	**26**	**2,8**
112	**16**	**3,3**

Необходимо найти значения уровня воды в бассейне в зависимости от длительностей заполнения $x_1 \in [100;130]$ и выпуска воды
$x_2 \in [15;25]$ с шагом Δ=5 минут. Построить поверхность.

Решение

1. Введем данные в рабочую таблицу: в ячейку А1 – символ x_1, в ячейку В1 – x_2, в ячейку С1- *y*. В диапазон С2:С8 – значения y.

2. Введем значения x_1 и x_2 для получения расчетных значений *y* в соответствии с заданием $x1 \in [100;130]$ в диапазон А10:А30, а $x_2 \in [15;25]$ в диапазон В10:В30 (рис.3.11).

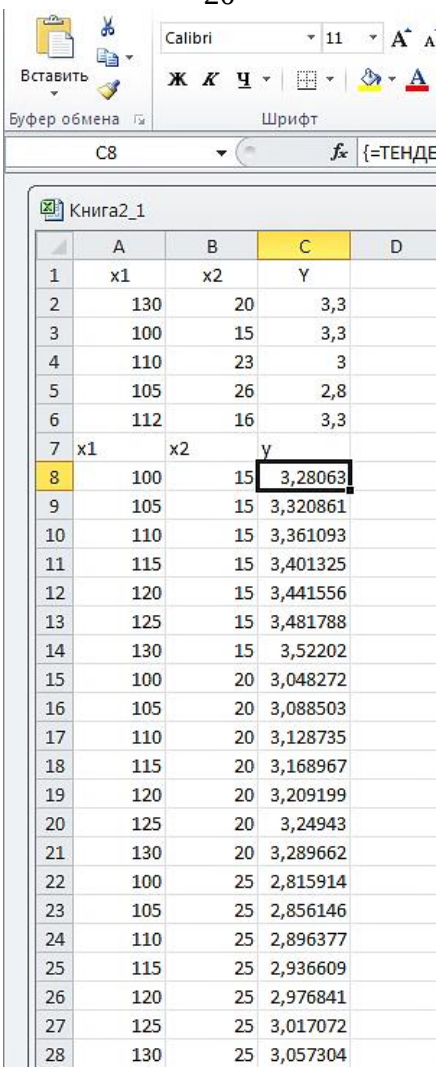
3. Выделяем блок ячеек С10:С30 под массив расчетных (предсказанных) значений *y*.

4. Поскольку уравнение для вычисления уровня воды линейное (3.4), вызываем функцию ТЕНДЕНЦИЯ (панель инструментов *Формулы,* кнопка «Вставить функцию», рабочее поле *Категория* тип *Статистические*, рабочее поле *Функция* вид *Тенденция*).

5. Заполняем рабочие поля: Изв_знач_y – С2:С8, Изв_знач_x – А2:В8, Нов_знач_x – А10:В30 (рис.10). Нажимаем сочетание клавиш <Ctrl+Chift+Enter>.

6. В результате в диапазоне С10:С30 получим предсказанные значения *y* (рис. 3.11).

7. Формируем блок данных для построения диаграммы. Для этого введем значения переменной x_1 в столбец Е. Для этого в ячейку Е1 вводим символ *x.* В диапазон ячеек Е2:Е8 – значение $x_1 \in [100;130]$ с шагом Δ =5 минут. В диапазон F1:H1 вводятся значения x_2 *15,20,30.* Затем диапазон F2:H8 заполняется соответствующими расчетными значениями *y.*



	A	B	C	D
1	x1	x2	Y	
2	130	20	3,3	
3	100	15	3,3	
4	110	23	3	
5	105	26	2,8	
6	112	16	3,3	
7	x1	x2	y	
8	100	15	3,28063	
9	105	15	3,320861	
10	110	15	3,361093	
11	115	15	3,401325	
12	120	15	3,441556	
13	125	15	3,481788	
14	130	15	3,52202	
15	100	20	3,048272	
16	105	20	3,088503	
17	110	20	3,128735	
18	115	20	3,168967	
19	120	20	3,209199	
20	125	20	3,24943	
21	130	20	3,289662	
22	100	25	2,815914	
23	105	25	2,856146	
24	110	25	2,896377	
25	115	25	2,936609	
26	120	25	2,976841	
27	125	25	3,017072	
28	130	25	3,057304	

Рис. 3.11. Расчетные значения y и соответствующие им значения x_1 и x_2 из примера 4

В результате должна быть получена следующая таблица (рис.3.12).

E	F	G	H
x	15	20	25
100	3,3	3,0	2,7
105	3,3	3,0	2,8
110	3,4	3,1	2,8
115	3,4	3,1	2,9
120	3,5	3,2	2,9
125	3,5	3,2	3,0
130	3,6	3,3	3,0

Рис. 3.12. Данные из примера 4, подготовленные для построения плоскости

8. Для построения диаграммы следует выделить все данные из полученной таблицы (рис.3.13)

	15	20	25
100	3,3	3	2,7
105	3,3	3	2,8
110	3,4	3,1	2,8
115	3,4	3,1	2,9
120	3,5	3,2	2,9
125	3,5	3,2	3
130	3,6	3,3	3

Рис.3.13. Данные из примера 4, используемые для построения плоскости

На панели инструментов выбираем: *Вставка --Гистограмма -- Все типы диаграмм. В окне Изменение типа диаграммы выбираем Поверхность, далее Проволочная поверхность.*

На вкладке *Работа с диаграммами* выбираем вкладку *Макет.* На этой вкладке нажимаем кнопку *Название осей* и подписываем ось Y, X1 и ось X2.

Правой кнопкой выделяем подписи по оси Y, из контекстного меню выбираем Формат оси и устанавливаем фиксированные значения по оси Y:минимальное значение – 2,5, а максимальное значение - 3,5 (рис.3.14).

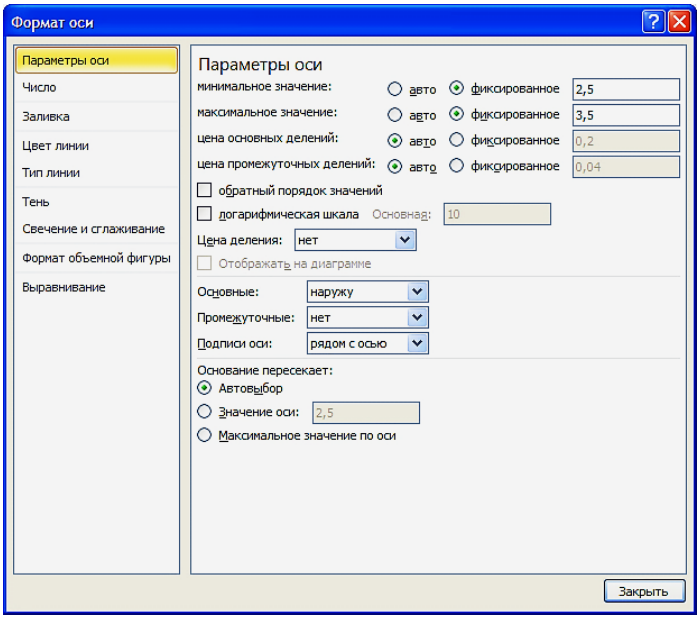


Рис. 3.14. Окно Формат оси

В результате выполненных действий будет получена диаграмма изменения уровня воды в бассейне (рис. 3.15).

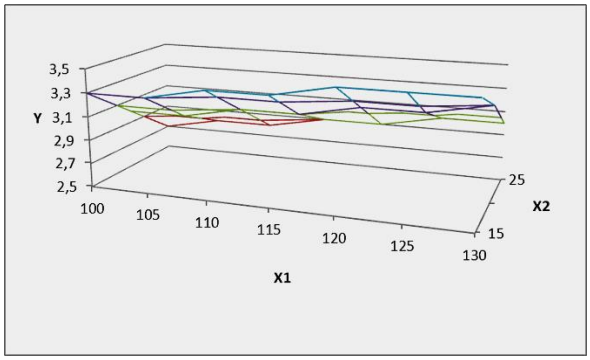


Рис. 3.15. Диаграмма изменения уровня воды в бассейне в зависимости от соотношения времени заполнения и выпуска воды (пример 4)

Задание для самостоятельной работы:

1. В условиях примера 4 найти параметры аппроксимирующего уравнения и оценить его точность.
2. В условиях примера 3 найти расчетные значения интенсивности излучения для следующих значений *x1* и *x2*:

X_1	0	0	0	200	200
X_2	0	1	3	0	1

3. Застройщик оценивает стоимость группы небольших офисных зданий в традиционном деловом районе. Оценку цены офисного здания в заданном районе застройщик предполагает осуществлять на основе следующих переменных:

y –оценочная цена здания под офис, x_1- общая площадь в квадратных метрах, x_2- количество офисов, x_3- количество входов, x_4 – время эксплуатации здания в годах. Предполагается, что существует линейная

зависимость между каждой независимой переменой (x_1,x_2,x_3 и x_4) и зависимой переменной (y), то есть ценой здания под офис в данном районе. Застройщик наугад выбирает 11 зданий из имеющихся 1500, и получает следующие данные:

x1	x2	x3	x4	y
2310	2	2	20	142000
2333	2	2	12	144000
2356	3	1,5	33	151000
2379	3	2	43	150000
2402	2	3	53	139000
2425	4	2	23	169000
2448	2	1,5	99	126000
2471	2	2	34	142900
2494	3	3	23	163000
2517	4	4	55	169000
2540	2	3	22	149000

Здесь «полвхода» (1/2) означает вход только для доставки корреспонденции. Найти параметры аппроксимирующего уравнения.

4. В условиях задания 3 с помощью функции Тенденция определить оценочную стоимость здания под офис в том же районе, которое имеет площадь 2500 квадратных метров, три офиса, два входа, зданию 25 лет

Контрольные вопросы

1. В чем заключается суть метода наименьших квадратов?
2. Виды аппроксимации по методу наименьших квадратов?
3. Для чего предназначен пакет анализа в Microsoft Excel?
4. Назначение функции ЛИНЕЙН в статистическом анализе.
5. Назначение функции ЛГРФПРИБЛ.
6. Назначение функций ТЕНДЕНЦИЯ и РОСТ.
7. Какие функции Microsoft Excel предназначены для аппроксимации линейных функций?
8. Какие функции Microsoft Excel предназначены для аппроксимации показательных функций?
9. Что показывает коэффициент детерминации?
10. Что означает понятие Линия тренда?
11. Какова максимальная степень полиноминальной функции?
12. Какие функции можно использовать для аппроксимации экспериментальных кривых?

Практическое занятие № 4

Знакомство со средой программирования VBA

Цель работы: получить представление о принципах работы в среде программирования VBA

Задание

1. Открыть MS Excel, создать и сохранить файл электронных таблиц.

2. Для запуска VBA следует добавить на панель инструментов вкладку Разработчик выполнив следующие действия: в ниспадающем меню вкладки *Файл* (рис. 4.1) выбрать *Параметры*, в окне *Параметры Excel* выбрать *Настройка ленты*.

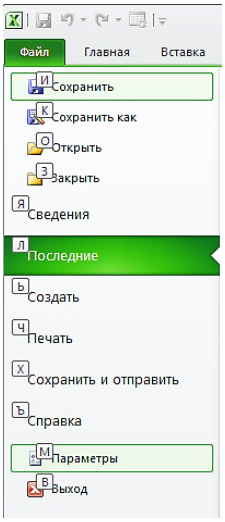


Рис. 4.1. Ниспадающее окно вкладки *Файл*

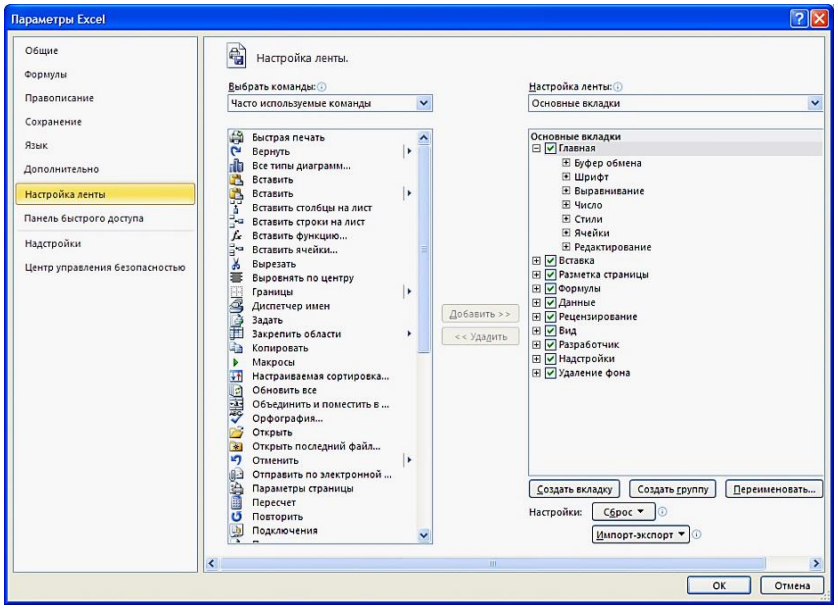


Рис. 4.2. Окно Параметры Excel

В меню со списком окна *Параметры Excel*(рис.4.2), *Настройка ленты* установить *Основные вкладки*, в окне *Основные вкладки* установить галочку в окошке *Разработчик*. После выполнения всех действий в строку меню добавится вкладка Разработчик (рис. 4.3).

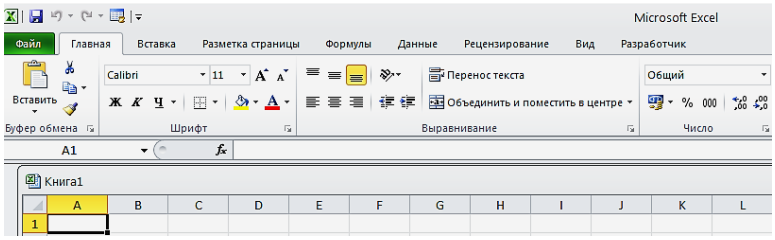


Рис. 4.3. Строка меню приложения Excel с вкладкой Разработчик

Запустить среду VBA, нажав кнопку *Visual Basic* вкладки *Разработчик.*

3. Добавить программный модуль VB*: Вставка(Insert)→ Module.*

4. Открыть окно проекта (Project-VBA Project): *Вид(View)→ Окно проекта(Project Explorer)* и найти добавленный модуль в структуре текущего файла Excel *(VBAPject(....xls)→Модули(Modules)→Модуль1(Module1)).*

5. Дважды щелкнуть на имени модуля в окне проекта и в открывшемся окне (окне программы) набрать текст процедуры:

```
Option Explicit
Sub first()
MsgBox "Добро пожаловать!", vbInformation, "Первая программа"
End Sub
```

6. Сохранить файл *(Файл(File)→Сохранить...(Save…))* и запустить модуль с процедурой first на выполнение командой: *Запуск(Run)→Запуск подпрограммы (Run Sub или F5).*

7. После завершения программы в окне модуля установить курсор на ключевом слове MsgBox и выполнить команду: *Правка(Edit)→Сведения(Quick Info)* – информация о синтаксисе функции или процедуры.

9. Установить курсор на тексте "Добро пожаловать" и выполнить команду: *Правка(Edit)→Параметры(Parameter Info)* – информация о текущем параметре функции или процедуры.

10. Добавить к проекту окно формы командой: Вставка(Insert)→UserForm. В проект будет добавлена новая форма, окно редактирования которой выводится на экран. В окне проекта будет добавлена группа Формы(Forms) с новой формой UserForm1.

11. Открыть окно свойств (Properties): Вид(View)→Окно свойств(Properties Window) для добавленной формы.

12. В окне свойств в поле Name ввести: *myforma* (в окне проекта изменится имя формы), в поле Caption ввести: *Первая программа* (в окне формы изменится ее заголовок).

13. Щелчком мыши перейти в окно формы и открыть панель элементов (ToolBox) командой: *Вид(View)→Панель элементов(ToolBox).*

14. В панели элементов щелкнуть на кнопке «Надпись(Label)» и мышью "нарисовать" в форме контур создаваемого элемента управления. В окне свойств для созданной надписи в поле Caption набрать: *Введите текст.*

15. В панели элементов щелкнуть на кнопке «Поле(TextBox)» и добавить поле для ввода текста в форму (под надписью). В окне свойств для созданного поля в свойстве Name набрать: *mytextbox.*

16. В панели элементов щелкнуть на кнопке «Кнопка(CommandButton)» и добавить управляющую кнопку в форму (под текстовым полем). В окне свойств для созданной кнопки в свойстве Caption набрать: *Завершение.*

17. Выделить все добавленные элементы в форме (надпись, текстовое поле и кнопку) и применить к ним команду: *Формат(Format)→Выровнять(Align)→По левому краю(Lefts).*

18. Выделить в форме добавленную кнопку и вызвать программу для обработки связанных с ней действий командой: *Вид(View)→Программа(Code)* (или *окно проекта→кнопка Программа(View Code)*).

19. В окне программы в открывшемся шаблоне набрать программу:

```
Option Explicit
Private Sub CommandButton1_Click()
Dim mytext As String
mytext = mytextbox.Text
MsgBox "Введено: " & mytext
Unload myforma
End Sub
```

Шаблон процедуры для кнопки (Private Sub CommandButton1_Click() ... End Sub) был добавлен автоматически. Процедура срабатывает при щелчке (Click) по объекту – кнопке (CommandButton1). В окне программы процедуры для различных объектов выбираются из двух списков (Объект(Object) и Процедура(Procedure)).

20. В окне программы для модуля (Module1) перед строкой:

```
End Sub
```

добавить строку:

myforma.Show

21. Сохранить файл и запустить модуль с процедурой first на выполнение.

22. Перейти в окно программы для кнопки (CommandButton1) и выполнить команду: *Отладка(Debug)→Добавить контрольное* значе*ние(Add watch).* В открывшемся окне в поле Выражение(Expression) набрать имя переменной: mytext. После нажатия кнопки «Ok» отобразится окно Контрольное значение(Watch): *Вид(View)→Окно контрольного значения(Watch Window).*

23. Запустите программу в режиме отладки *(Отладка(Debug)→Шаг с заходом(Step Into)* или F8). В запущенном окне диалога введите текст и нажмите кнопку «Завершение». В открывшемся окне VB желтым цветом отмечается текущая операция. Для выполнения следующей операции нужно нажать F8. Выполняя, т.о., программу по шагам, можно установить с помощью окна контрольного значения при выполнении какой строки кода переменной mytext будет присвоено значение.

24. После завершения выполнения программы в ее тексте замените mytext на mytex в строке MsgBox "Введено: " & mytext. Запустите программу на выполнение (F5).

25. Введите текст в поле ввода и нажмите кнопку «Завершение». Прочитайте сообщение об ошибке и нажмите в его окне кнопку «Ok». В открывшемся окне VB желтым цветом будет отмечена процедура, содержащая ошибку, а синим – место ошибки. Ошибку можно устранить (заменить mytex на mytext), не завершая текущий запуск программы (исправить ошибку и нажать кнопку «F5 (Продолжить(Continue)))» или завершив его (*Запуск(Run)→Сброс(Reset)*) для дальнейшего редактирования.

26. После завершения выполнения программы и устранения ошибки в процедуре CommandButton1_Click() установите курсор перед ключевым словом MsgBox и выполните команду *Отладка(Debug)→Точка останова(Toggle Breakpoint)* или щелкните на поле слева от строки. Запустите программу на выполнение (F5).

27. Введите текст в поле ввода и нажмите кнопку «Завершение». Процедура будет приостановлена перед строкой отмеченной желтым цветом (точка останова). Убедитесь в наличии значения в переменной mytext (окно контрольного значения). Завершите выполнение программы (F5 или F8(по шагам)).

28. Очистите все точки останова (щелчок слева от строки с остановом или *Отладка(Debug)→Снять все точки останова(Clear All BreakPoints))* и контрольные значения (в окне контрольного значения команда контекстного меню *Удалить контрольное значение(Delete Watch)).* Сохраните файл.

29. Закройте проект и вернитесь в Excel *(Файл(File)→Закрыть и вернуться в Microsoft Excel(Close and Return to Microsoft Excel)).*

30. Запустите процедуру first командой оболочки Excel: *Сервис→Макрос→Макросы→first (в списке Имя макроса)→Выполнить.*

31. В оболочке Excel выполните команду*: Вид→Панели инструментов→Элементы управления.* Выберите на панели элемент управления *Кнопка* и нарисуйте его контур на листе Excel.

32. В контекстном меню кнопки выполните команду *Исходный текст.*

33. В открывшемся окне VB внутри шаблона процедуры новой кнопки вызовите процедуру first:

Private Sub CommandButton1_Click()

first

End Sub

34. Сохраните файл и вернитесь в Excel.

35. Выполните команду*: панель Элементы управления→Режим конструктора или Вид→Панели инструментов→Visual Basic* и далее панель *Visual Basic→Выход из режима конструктора*.

36. Нажмите на добавленную кнопку на листе Excel.

37. Выполните команду оболочки Excel: *Вид→Панели инструментов→Настройка→Панели инструментов→Создать.* Назовите новую панель и нажмите кнопку « Ok».

38. В закладке *Команда* окна *Настройка* выберите категорию *Макросы.* В списке *Команды* выберите *Настраиваемая кнопка* и перетащите ее на созданную панель.

39. С помощью контекстного меню на кнопке новой панели инструментов выполните команду *Назначить макрос*. В предложенном списке выберите процедуру first и нажмите кнопку « Ok». Закройте окно Настройка.

40. Нажмите на кнопку созданной панели инструментов.

Контрольные вопросы

1. Каким образом открыть редактор Visual Basic в MS Excel?
2. Каким образом добавить програмный модуль и сохранить проект Visual Basic MS Excel?
3. Как получить инфорацию о синтаксисе текущей процедуры или функции в Visual Basic?
4. Как вызывается и для чего используется Окно свойств Visual Basic (пример)?
5. Как добавить форму в проект Visual Basic? Как добавить элемент управления в форму?
6. Как вызвать окно программного кода для элемента управления?
7. Как добавить контрольное значение в Visual Basic?

8. Как запустить программу с остановкой на каждом шаге?
9. Как добавить точки останова в программу Visual Basic?
10. Перечислите и поясните способы запуска программ Visual Basic в MS Excel.

Практическое занятие № 5

Создание пользовательской формы
Элементы управления
Встроенные функции VBA. Основы программирования
Оператор выбора. Встроенные диалоговые окна

Цель работы: изучить:

1. Создание пользовательской формы.
2. Свойства элементов управления кнопка и поле.
3. Встроенные функции VBA (математические, проверки типов, преобразования форматов, обработки строк).
4. Отладка программ.
5. Типы данных, описание переменных, допустимые имена, область определения переменных, условный оператор If.

Основные понятия

Интегрированная среда разработки. Интегрированная среда разработки состоит из нескольких составляющих, название и назначение которых приведены в табл. 5.1.

Составляющие интегрированной среды разработки

Таблица 5.1

Наименование окна	Команда View(Вид)	Описание
1	2	3
Project Explorer (проект)	Project Explorer (Окно проекта)	Предназначено для отображения всех
Toolbox (панель элементов)	Toolbox (панель элементов)	Содержит панель элементов для конструирования форм
UserForm	Object(Объект)	Используется для создания форм путем размещения на них элементов управления
Code(Программа)	Code (Программа)	Предназначено для просмотра, написания и редактирования программы на языке VBA.
		Поскольку среда разработки является многооконной, то для каждого модуля проекта можно открыть отдельное окно
Propertis (Свойства)	Propertis (Окно свойств)	Отображает свойства выделенных объектов. В этом окне можно задавать новые значения свойств формы и элементов управления
Object Browser (Просмотр объектов)	Object Browser (Просмотр объектов)	Отображает классы, свойства, методы, события и константы различных библиотек объектов. Используется для быстрого получения информации об объектах
Immediate (Проверка)	Immediate (Окно отладки)	Предназначено для быстрого выполнения вводимых в него инструкций. В данном окне также выводятся и результаты выполнения вводимых инструкций
Locals (Локальные переменные)	Locals (Окно локальных переменных)	Автоматически показывает все переменные данной процедуры
Watches (Контрольные значения)	Watches (Окно контрольных значений)	Применяется при отладке программ для просмотра значений выражений

Создание формы и размещение в ней элементов управления. Чтобы добавить форму в проект, щелкните правой кнопкой мыши на соответствующем проекте. Затем выберите в раскрывающемся контекстном меню команду *Insert→UserForm*. После этого появится изображение формы.

Объектная модель библиотеки объектов Ms Forms. В данной обектной модели (рис. 5.1) можно выделить три типа объектов:

Элементы управления (Controls). К ним относятся такие объекты, как TextBox (Текстовое поле),

ComboBox (Поле со списком), Label (Надпись) и т.п.

Объекты, содержащие в себе коллекции (Collections). Это объекты MultiPage и TabStrip, содержащие соответственно коллекции Pages и Tabs.

Объекты, являющиеся членами соответствующих коллекций. Например, объекты Page и Tab являются объектами, входящими в состав соответствующих коллекций Pages и Tabs.

Основные свойства формы. Свойство *Caption* определяет заголовок окна формы. Значением данного свойства может быть любая строка.

Свойство *ShowModal* при задании ему значения True позволяет сделать форму модальной. Пока модальная Фома открыта, другие окна приложения недоступны, хотя отображаются на экране.

Свойство *ScrollBars* определяет, какие полосы прокрутки присутствуют на экране. Значение данного свойства может равняться значению одной из следующих констант:

fmScrollBarsNone – полосы прокрутки отсутствуют;

fmScrollBarsHorizontal - присутствует только горизонтальная полоса прокрутки;

fmScrollBarsVertical – присутствует только вертикальная полоса прокрутки;

fmScrollBarsBoth – присутствуют обе полосы прокрутки;

Все открытые формы являются членами коллекции *UserForms.* Если открыто несколько форм, то, чтобы сослаться на конкретную форму, можно использовать ее порядковый номер в коллекции. Чтобы сослаться на первую открытую форму, можно добавить в программу следующую инструкцию:

UserForms.Item(0)

Для ссылки на форму по имени достаточно указать ее имя:

UserForm1.Height = 300

В данном примере мы задаем высоту (*Height*) формы с именем UserForm1.

Коллекция *UserForms,* как и большинство других коллекций, также имеет метод *Add,* который позволяет добавить новую форму.

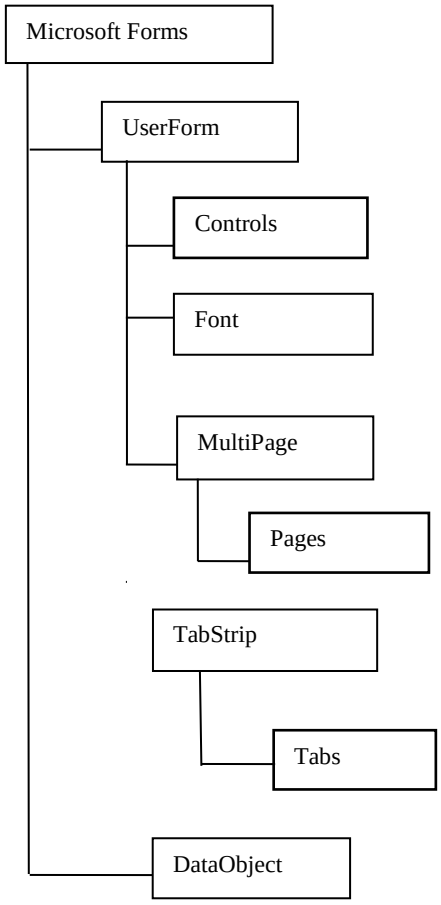


Рис. 5.1. Объектная модель библиотеки объектов Ms Forms.

Основные элементы управления. Элемент управления *Label (Надпись)* отображает текст. Обычно надписи отображают на формах справочную информацию. Например, рядом с полем можно разместить надпись с текстом, поясняющую назначение данного поля.

Элемент управления *TextBox(Поле)* обычно служит для ввода данных пользователем. В текстовом поле можно ввести любой текст.

К основным свойствам текстового поля относятся: *Text, Value, MultiLine.*

Свойство Text определяет текст в текстовом поле. Значением данного свойства может быть строковое

выражение.

Элемент управления *RadioButton (Переключатель)* имеет два состояния: включено и выключено. Переключатели объединяются в группы, причем включен может быть только один переключатель в группе.

Элемент управления *CheckBox(Флажок)* позволяет установить одно из состояний: включено или выключено.

Элемент управления *List Box(Список)* используется в том случае, когда необходим выбор одного или нескольких вариантов из списка. Список может иметь несколько столбцов, если задать соответствующие значения свойству *ColumnCount.*

Для добавления в список нового элемента следует использовать метод *AddItem.* В случае если список состоит из нескольких строк, то к нему будет добавлена новая строка.

Синтаксис :Objekt.AddItem [Item [,Index]](табл.5.2)

Параметры метода Objekt.AddItem

Таблица 5.2

Элемент	Описание
Object	Объектная переменная одного из следующих типов:ComboBox или ListBox
Item	Определяет элемент или строку списка, которая будет добавлена. Например первого элемента списка или строки соответствует 0, второго -1 и т.д.
Index	Определяет позицию добавляемого элемента в списке. Значением данного свойства может быть целое число, которое не должно превышать число элементов управления

Пример процедуры, в которой выполняется добавление элементов в список.

```
Private Sub UserForm_Activate()
ListBox1.AddItem(”Item1”)
ListBox1.AddItem(”Item2”)
ListBox1.AddItem(”Item3”)
End Sub
```

Элемент управления ComboBox. Объединяет в себе два элемента управления: TextBox (Поле) и ListBox (Список). Иногда данный элемент управления также называют комбинированным списком. Комбинированному списку присущи свойства как текстового поля, так и списка.

Элементы управления CommandButton (Кнопка)

Применяются обычно для выполнения какого-либо действия. Так, при нажатии на кнопку может открываться другая форма, выполняться какие-либо действия. Так, при нажатии на кнопку может открываться другая форма, выполняться какие-нибудь расчеты и т.д.

Как правило, в процедуру обработки события *Click* данной кнопки вставляются инструкции, выполняющие необходимые действия.

Для иллюстрации возможностей использования элемента управления *CommandButton* (Кнопка) вернемся к приведенному ранее примеру со списком. Вместо того чтобы поместить инструкции, добавляющие элементы списка в процедуру обработки события Click кнопки, как показано ниже

```
Private Sub CommandButton1_Click()
ListBox1.AddItem(”Item1”)
ListBox1.AddItem(”Item2”)
ListBox1.AddItem(”Item3”)
End Sub
```

Элементы управления Scroll Bar.

С помощью элемента управления ScrollBar (Полоса прокрутки) можно задать значение свойств другого элемента управления, например, текстового поля.

```
Private Sub ScrollBar1_Change()
TextBox1.Text=ScrollBar1.Value
End Sub
```

В приведенном примере изменение положения бегунка полосы прокрутки приводит к изменению значения, отображаемого в текстовом поле. Создана процедура обработки события Change.

В этой процедуре значению свойства *Text* элемента управления *TextBox (Поле)* с именем *TextBox1* присваивается значение свойства *Value,* Полосы прокрутки *(ScrollBar)* с именем *ScrollBar1*.

Задания к практическому занятию

Задание 1. Рассмотреть создание пользовательской формы на примере простейшего диалогового окна (рис.5.2).

1. Создание диалогового окна приложения.

1.1. Запустить среду VBA, нажав кнопку Visual Basic вкладки Разработчик.

1.2. Выполнить команду *Вставить* → UserForm. В редакторе появятся окно с пользовательской формой и панель с элементами управления.

1.3. Используя панель инструментов, заполните пользовательскую форму элементами управления, создав требуемое диалоговое окно приложения. Это окно состоит из трех полей ввода и одной кнопки (см. рис.5.2). Для отображения панели инструментов (если ее нет на экране) нажать кнопку  .

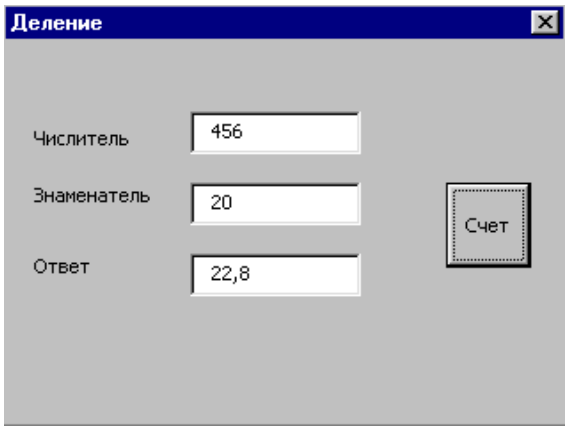


Рис. 5.2. Пример диалогового окна

1.4. Используя диалоговое окно Свойства (Properties), отображаемое нажатием кнопки

1.5.  , свойство Caption пользовательской формы определить равным Деление, а кнопки - равной Счет.

2. Написание процедуры обработки событий

2.1. Для написания процедуры обработки события - нажатия кнопки «*Счет*», дважды ее щелкнуть. Откроется редактор кода на листе модуля *UserForm1*. Более того, он откроется на том месте, где программируются действия, связанные с элементом управления, который дважды щелкнули (в данном случае с кнопкой «*Счет*»). Если код еще не набран, то при открытии редактора кода появятся инструкции заголовка и окончания процедуры, которая будет ассоциирована с активизированным элементом управления:

```
Private Sub CommandButton1_Click()

End Sub
```

Введите текст процедуры:

```
Private Sub CommandButton1_Click()
        Dim Числитель, Знаменатель, Результат As Double
        Числитель = CDbl(TextBox1.Text)
        Знаменатель = CDbl(TextBox2.Text)
        Результат = Числитель / Знаменатель
        TextBox3.Text = CStr(Результат)
End Sub
```

3. Выполнение созданной программы

3.1. Процесс создания диалогового окна и процедуры, связанной с ним, завершен. Для выполнения созданной программы нажать кнопку  или выполнить команды меню *Run → Run Sub UserForm* или нажать функциональную клавишу *F5*. На экране на фоне рабочего листа отобразится заданное диалоговое окно.

4. Проверить функционирование кнопок созданного приложения

5. Модернизировать ранее созданное приложение, которое производит деление ***Числителя*** на ***Знаменатель*** по нажатию кнопки ***Счет*** с контролем возможных ошибок. Добавить контроль: Проверка деления на 0.

5.1. Добавить следующие инструкции в процедуры обработки событий.

```
If CDbl(TextBox2.Text) = 0 Then
MsgBox "Знаменатель не может быть нулем", vbInformation, "Деление"
TextBox2.SetFocus
Exit Sub
End If
```

5.2. Выполнить созданную программу.

6. Создать приложение, в котором появляется диалоговое окно ввода с полем ввода. В зависимости от введенной информации появляются два окна с приведенными ниже сообщениями.

6.1. Диалоговое окно ввода и окна с сообщениями должны выглядеть следующим образом (рис. 5.3):

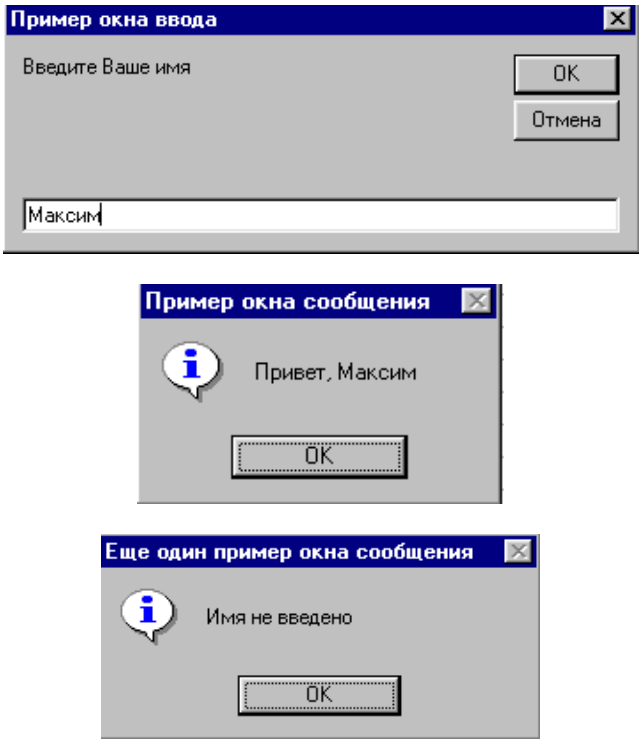


Рис.5.3. Примеры окон сообщений и диалогового окна

6.2. Написание процедуры.

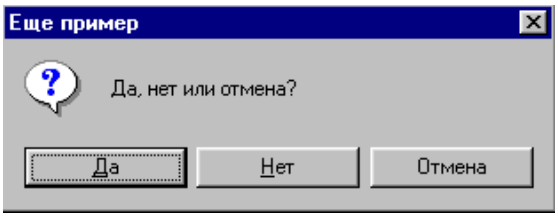
Так как для выполнения программы не создается пользовательская форма, а предусмотрена работа со встроенными диалоговыми окнами, то код программы набирается в модуле. Чтобы открыть окно для редактирования кода, следует выполнить команды *Вставка →Модуль*.

```
Private Sub ТекстОкон()
ИмяКлиента = InputBox("Введите Ваше имя", "Пример окна ввода")
If ИмяКлиента <> "" Then
MsgBox "Привет, " & ИмяКлиента, vbInformation, _
"Пример окна сообщения"
Else
MsgBox "Имя не введено " & ИмяКлиента, vbInformation, _
"Еще один пример окна сообщения"
End If
End Sub
```

6.3.Выполнить созданную программу.

7. Создать приложение, позволяющее отобразить на экране диалоговое окно с тремя кнопками ***Да***, ***Нет*** и ***Отмена*** с информационным знаком. При нажатии каждой из этих кнопок на экране появляется сообщение, подтверждающее нажатие (рис.5.4) .

Диалоговое окно с запросом и окна с сообщениями должны выглядеть следующим образом:



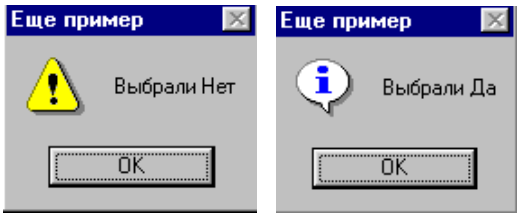


Рис.5.4. Примеры диалоговых окон

```
Private Sub ТриКнопки()
Dim Сообщение As String
Dim Кнопка As Integer
' В переменной Сообщение задается структура диалогового окна
Сообщение = vbYesNoCancel + vbQuestion
Rem В переменную Кнопка вводится целое число, возвращаемое_
Rem MsgBox при нажатии нкопки
Кнопка = MsgBox("Да, нет или отмена?", Сообщение, "Еще пример")
' В зависимости от значения переменной Кнопка,
' на экране отражается соответствующее сообщение
Select Case Кнопка
   Case vbYes
      MsgBox "Выбрали Да", vbInformation, "Еще пример"
      Case vbNo
      MsgBox "Выбрали Нет", vbExclamation, "Еще пример"
         Case vbCancel
         MsgBox "Выбрали Отмена", vbCritical, "Еще пример"
       End Select
End Sub
```

Отчет по практическому занятию должен содержать краткие теоретические сведения, диалоговые окна созданных приложений и тексты программ.

Практическое занятие №6

Оператор варианта

Оператор цикла с параметром. Массивы. Список. Поле со списком

Цель работы: изучить:
1. Оператор цикла с параметром *For*, массивы.
2. Элементы управления список и поле со списком, их свойства и заполнение.
3. Процедуру инициализации формы.

Основные понятия

Коллекция *Documents* содержит все открытые документы (объекты *Document*). Доступ к элементам данной коллекции можно получить либо по имени документа, либо по его индексу.

ActiveDocument – объект активный документ

Свойства объекта *Document* позволяют установить некоторые атрибуты документа, такие как тип документа, наличие автоматической проверки орфографии и т.п.

Paragraphs – свойство объекта *Document* , возвращает коллекцию *Paragraphs*, которая содержит все абзацы в заданном документе , диапазоне или выделении.

Add – метод объекта *Document* , используется для добавления нового пустого документа к коллекции открытых документов.

Объект *Range* представляет собой непрерывную область (фрагмент) документа. Его местонахождение определяется позициями начального и конечного символов фрагмента документа.

InsertAfter - метод объекта *Range*. Этот метод вставляет заданный текст в конец диапазона или выделения. В результате применения данного метода диапазон или выделение будет расширено настолько, чтобы включить вставлены текст.

Элемент управления *ListBox* (список) создается с помощью кнопки **Список** (*ListBox*) (рис.6.1). Элемент управления *ListBox* применяется для хранения спмска значений. Из списка пользователь может выбрать одно или несколько значений, которые в последующем будут использоваться в тексте программы.

Наиболее часто используемые элементы управления *ListBox*:

ListIndex	Возвращает номер текущего элемента списка. Нумерация элемента списка начинается с нуля
ListCount	Возвращает число элементов списка
ColumnCount	Устанавливает число столбцов в списке
TextColumn	Устанавливает столбец в списке, элемент которого возвращается свойством Text
Text	Возвращает выбранный в списке элемент
List	Возвращает элемент списка, стоящий на пересечении указанных строки и столбца. **Синтаксис:** List(row, column)
RowSourse	Устанавливает диапазон, содержащий элементы списка
MultiSelect	Устанавливает способ выбора элементов списка. Допустимые значения: – fmMultiSelectSingle (выбор только одного элемента) – fmMultiSelectMulti (разрешен выбор нескольких элементов посредством либо щелка, либо нажатием клавиши <Пробел>) – fmMultiSelectExtended (разрешено использование клавиши <Shift> при выборе ряда последовательных элементов списка)
ColumnWidths	Устанавливает ширину столбцов списка. **Синтаксис:**ColumnWidths=String String – строка, устанавливающая ширину каждого из трех столбцов списка: Widths ListBox1 .ColumnCount=3 .ColumnWidths= “20;30;30” End With

Наиболее часто используемые методы элемента управления ListBox:

Clear	Удаляет все элементы из списка
RemoveItem	Удаляет из списка элемент с указанным номером. **Синтаксис:**RemoveItem (index) Index-номер удаляемого из списка элемента
AddItem	Добавляет элемент в список. AddItem ([item[, varIndex]]) Item – элемент (строковое выражение), добавляемый в список varIndex – номер, добавляемого элемента

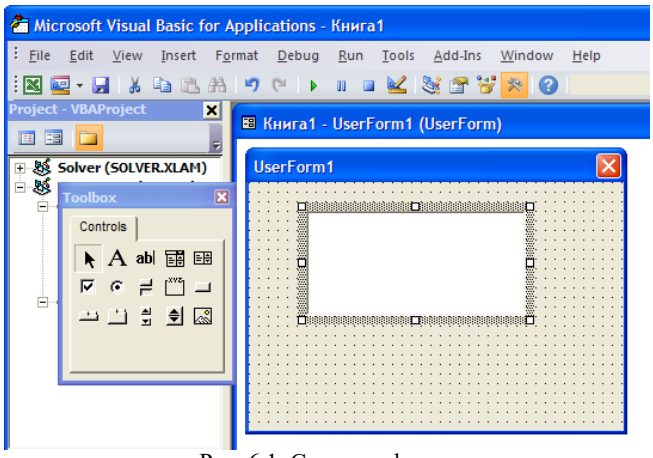


Рис. 6.1. Список в форме

Для добавления в список нового элемента следует использовать метод AddItem. В случае если список состоит из нескольких строк, то к нему будет добавлена новая строка. Рассмотрим подробнее метод AddItem. Синтаксис данного метода представлен ниже:

Object	Объектная переменная одного из следующих типов: ComboBox или ListBox
Item	Определяет элемент или строку списка, которая будет добавлена. Номер первого элемента списка или строки соответствует 0, второго – 1 и т.д.
Index	Определяет позицию добавляемого элемента в списке. Значением данного свойства может быть целое число, которое не должно превышать число элементов управления

Заполнить список можно одним из следующих способов:

Поэлементно, если список состоит из одной колонки	With ListBox1 .AddItem “ Июнь” .AddItem “Июль” .AddItem “Август” .ListIndex=0 EndWith
Массивом, если список состоит из одной колонки	With ListBox1 .List=Array(“Июнь”, “Июль”, “Август”) .ListIndex=1 EndWith
Из диапазона A1:B4, в который предварительно введены элементы списка. Результат выбора (индекс выбранной строки) выводится в ячейку C1	With ListBox1 .ColumnCount=2 .RowSourse=”A1:B4” .ControlSourse=”C1” .BoundColumn=0 EndWith

Продолжение

Поэлементно, если список состоит из нескольких колонок, например двух	With ListBox1 .ColumnCount=2 .AddItem “Июнь” .List(0,1)=”Сессия” .AddItem=”Июль” .List(1,1)=”Каникулы” .AddItem=”Август” .List(2,1)=”Каникулы” EndWith
Массивом, если список состоит из нескольких колонок, например двух	Dim A(2,1) As String A(0,0)=”Июнь” A(0,1)=”Сессия” A(1,0)=”Июль” A(1,1)=”Каникулы” A(2,0)=”Август” A(2,1)=”Каникулы” With ListBox1 .ColumnCount=2 .List=A EndWith

Задания к практическому занятию

Задание 1. Создать следующую форму: Кнопка *Запуск* определяет, является ли вводимое выражение символом, четным или нечетным числом, большой или маленькой латинской буквой. Кнопка *Выход* служит для выхода из программы.

Ход работы:

1. Создаём стандартный exe-файл.
2. Переносим на форму 2 кнопки (рис.6.2)

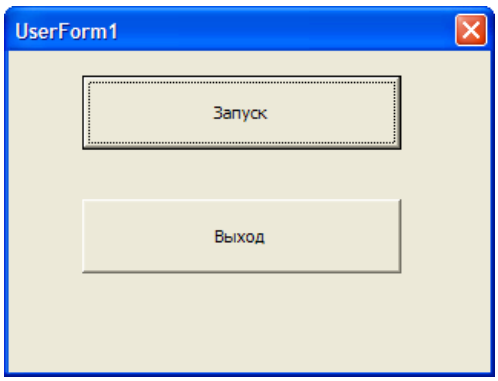


Рис. 6.2. Форма к заданию 1

3.Устанавливаем значение *Caption* первой кнопки – *Запуск*, второй – *Выход*
4. Вводим код в окно кода:

```
Private Sub CommandButton1_Click()
   Dim msg, w As String
   msg = "Введите символ"
   w = InputBox(msg)
   If Not (IsNumeric(w)) Then
      If Len(w) = 1 Then
         Select Case Asc(w)
            Case 65 To 90
               msg = "Введена большая лат.буква"
            Case 97 To 122
               msg = "Введена малая лат.буква"
            Case Else
               msg = "Это не латинская буква и не число"
         End Select
      Else
         msg = "Это не символ"
      End If
   Else
      Select Case Val(w)
            Case 1, 3, 5, 7, 9
               msg = "Это нечётная цифра"
            Case 0, 2, 4, 6, 8
               msg = "Это чётная цифра"
            Case Else
               msg = "Это не цифра и не буква"
         End Select
   End If
     MsgBox(msg)
End Sub
Private Sub Command2_Click()
End
End Sub
```

Запускаем приложение (рис. 6.3) и вводим символы.

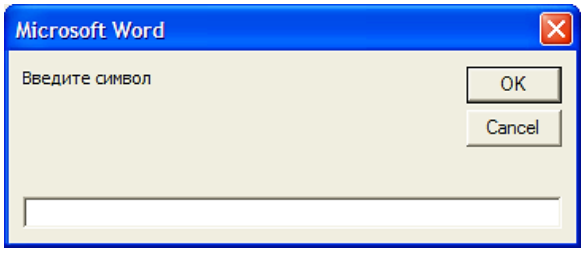


Рис. 6.3. Форма в режиме заполнения

Задание №2. Создать форму (рис.6.4), в которой:
будет создаваться массив случайных чисел, и выполняться сортировка чисел по возрастанию;
будет создаваться массив случайных чисел, и выполняться сортировка чисел по убыванию;
предусмотрена кнопка *Выход*.

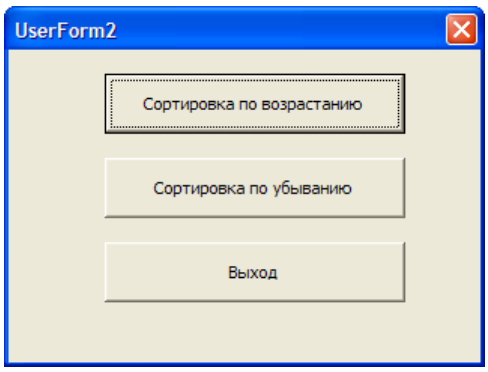


Рис.6.4. Форма 2

Ход работы:
1. Создаём проект в среде Microsoft Word, перетаскиваем на него 3 кнопки и надпись
2. Меняем следующие свойства:
   Кнопка 1 *Caption* Сортировка по возрастанию
   Кнопка 2 *Caption* Сортировка по убыванию
   Кнопка 3 *Caption* Выход
   Метка 1 *Caption*
3. Вводим код в окно кода:

```
Private Sub CommandButton1_Click()
   Dim i As Integer, b(10) As Single
Dim CurRange As Range
' Добавляет новый пустой абзац в открытый документ
ActiveDocument.Paragraphs.Add
ActiveDocument.Content.InsertAfter Text:=”Исходный массив:”
ActiveDocument.Paragraphs.Add
   For i = 1 To 10
      b(i) = Int(10 * Rnd + 10)
ActiveDocument.Content.InsertAfter Text:=Str(b(i))+’’ ”
   Next i
ActiveDocument.Paragraphs.Add
ActiveDocument.Content.InsertAfter Text:=”Отсортированный по возрастанию массив:”
Call SortMas(b(),10,2)
ActiveDocument.Paragraphs.Add
For i=1 To 10
ActiveDocument.Content.InsertAfter Text:=Str(b(i))+” ”
Next i
End Sub
Private Sub CommandButton2_Click()
   Dim i As Integer, b(10) As Single
ActiveDocument.Paragraphs.Add
ActiveDocument.Content.InsertAfter Text:=”Исходный массив:”
```

```
ActiveDocument.Paragraphs.Add
   For i = 1 To 10
      b(i) = Int(10 * Rnd + 10)
ActiveDocument.Content.InsertAfter Text:=Str(b(i))+" "
   Next i
ActiveDocument.Paragraphs.Add
ActiveDocument.Content.InsertAfter Text:="Отсортированный по убыванию массив:"
   Call SortMas(b(), 10, 1)
ActiveDocument.Paragraphs.Add
   For i = 1 To 10
ActiveDocument.Content.InsertAfter Text:=Str(b(i))+" "
   Next i
End Sub
Private Sub Command3_Click()
End
End Sub
```

4. Добавляем модуль и вводим код:

```
Sub SortMas(a() As Single, n As Integer, f As Byte)
   Dim i As Integer, j As Integer
   If f = 1 Then
      For i = 1 To n - 1
         For j = i + 1 To n
            If a(i) < a(j) Then
               a(i) = a(i) + a(j)
               a(j) = a(i) - a(j)
               a(i) = a(i) - a(j)
            End If
         Next j
      Next i
   Else
      For i = 1 To n - 1
         For j = i + 1 To n
            If a(i) > a(j) Then
               a(i) = a(i) + a(j)
               a(j) = a(i) - a(j)
               a(i) = a(i) - a(j)
            End If
         Next j
      Next i
   End If
End Sub
```

Запускаем приложение.

Отчет по практическому занятию должен содержать краткие теоретические сведения, диалоговые окна созданных приложений и тексты программ.

Практическое занятие № 7

Структура HTML-документа. Текстовое оформление страниц

Цель работы: ознакомиться с основными понятиями языка HTML, тегами, текстовым оформлением страниц, вставкой изображений.

Основные понятия

В последние два десятилетия Интернет прочно вошел в нашу жизнь. В наше время он охватывает почти все сферы жизни: культура, искусство, бизнес.

Наряду с обычными музеями, библиотеками, магазинами все большую популярность во всем мире завоевывают аналогичные Интернет-ресурсы. Даже бизнес становится виртуальным: создаются Web-представительства фирм, электронные магазины и торговые площадки, электронные банки, которые дают возможность осуществлять типичные для этих сфер операции с большей скоростью. В связи с этим профессия Web-дизайнера и Web-программиста стала востребованной и у нас в России. Интернет-технологии создания указанных web-ресурсов требует владения такими языками для написания серверных скриптов, как Perl, PHP, ASP, владения языком XML, а также умения создавать динамические страницы и сайты, используя скрипты VBScript и JavaScript, которые позволяют осуществлять вычисления, работу с датой и временем, изменение элементов страницы по желанию пользователя.

Perl – язык для написания CGI-скриптов (Common GateWay Interface). CGI-скрипт – это программа, которая выполняется на Web-сервере по запросу посетителя сайта. CGI – это не язык программирования, а специальный интерфейс, с помощью которого и происходит запуск скрипта и взаимодействие с ним.

PHP-скрипт (Hypertext Preprocessor) – скрипт, написанный на языке PHP и внедренный в HTML-документ.

ASP-скрипт (Active Server Pages) – скрипт, написанный на языке Visual Basic Scripting Edition и внедренный в HTML-документ.

Но это потом. А на первом этапе необходимо научиться создавать простые HTML-страницы. Не стоит считать, что в этом нет необходимости. Простого HTML вполне хватит для двухстраничной коммерции, которая распространяется сейчас по всему миру. HTML позволяет создать «заготовки» страниц, а уже потом можно добавлять в них различные скрипты. XML же – это надстройка над HTML. Поэтому без изучения HTML не обойтись. Итак, HTML – Hyper-Text MarkUp Language – язык разметки гипертекста. Гипертекст – расширенный текст, который может содержать в себе текст, иллюстрации, различные внедренные объекты и ссылки на другие ресурсы или на другие Web-страницы.

При передаче информации через Интернет используются как раз HTML- страницы – файлы *.htm и *.html, поскольку они содержат в себе гипертекст, имеют размер меньше, чем текстовые или иные файлы и для их просмотра нужен только браузер.

HTML-документ представляет собой обычный текстовый документ с управляющими конструкциями языка HTML – тегами, которые и производят действия «форматирования» над текстовыми блоками и осуществляют вставку различных объектов в документ.

Создавать HTML-страницы можно вручную – путем самостоятельного написания тегов в текстовом редакторе или используя Web-редакторы с автоматизацией ввода нужных тегов – например, WebEdit, Arachnophilia или же используя программы, которые создают теги сами в ответ на действие пользователя на странице – по принципу What You See Is What You Get – например, MS Word, MS Frontpage. Но они задают абсолютное форматирование и часто дописывают много лишнего кода к странице. К тому же для создания более уникальных страниц ими не обойдешься, для этого лучше писать теги HTML самому и использовать каскадные таблицы стилей. Мы предлагаем использовать для создания Web-страниц блокнот, который входит в стандартный набор любой из операционных систем Windows и не тре-бует предварительного изучения.

Наш практикум содержит в себе все основные разделы, необходимые для создания Web-страниц. В каждом разделе есть примеры использования тегов, показан результат действия этих тегов, а также представлены лабораторные работы. Практикум можно использовать для самостоятельного изучения технологии создания статических Web-страниц студентам любых специальностей или тем, кто хочет самостоятельно изучить основы языка HTML.

транице – по принципу What You See Is What You Get – например, MS Word, MS Frontpage. Но они задают абсолютное форматирование и часто дописывают много лишнего кода к странице. К тому же для создания более уникальных страниц ими не обойдешься, для этого лучше писать теги HTML самому и использовать каскадные таблицы стилей. Мы предлагаем использовать для создания Web-страниц блокнот, который входит в стандартный набор любой из операционных систем Windows и не тре-бует предварительного изучения.

Наш практикум содержит в себе все основные разделы, необходимые для создания Web-страниц. В каждом разделе есть примеры использования тегов, показан результат действия этих тегов, а также представлены лабораторные работы. Практикум можно использовать для самостоятельного изучения технологии создания статических Web-страниц студентам любых специальностей или тем, кто хочет самостоятельно изучить основы языка HTML.

Введение в язык HTML. Автономные Web-документы используют язык HTML (HypertextMarkup

Language - язык разметки гипертекста). Гипертекст, то есть расширенный текст, включает дополнительные элементы: иллюстрации, ссылки, вставные объекты.

Под разметкой понимается использование специальных кодов, легко отделяемых от смыслового содержания документа и используемых для реализации гипертекста. Применение этих кодов подчиняется строгим правилам, определяемым спецификацией языка HTML.

HTML -документ - это файл, содержащий обыкновенный текст со специальными командами. Такой файл может быть подготовлен в произвольном текстовом редакторе (существуют, однако, специальные программы-конверторы и HTML-редакторы).

HTML -документ состоит из содержимого, то есть собственно полезной информации, и команд, задающих структуру.

Каждая команда (управляющая конструкция) HTML -документа (тег) должна заключаться в угловые скобки - вот так: <тег>. Чаще всего в документе встречаются парные теги (открывающий и соответствующий ему закрывающий), так как браузеру необходимо знать область действия тега. Существуют и одиночные теги, однако, используются они только там, где область действия очевидна и дополнительной информации не требуется (ясно, например, что если мы встретили тег "начало абзаца" (**<P>**), то предыдущий абзац уже закончился). В сомнительном же случае лучше перестраховаться и поставить закрывающий парный тег, иначе документ может оказаться нечитаемым. Открывающий и закрывающий теги называются одинаково и отличаются друг от друга только символом "наклонная черта" или "слэш" - "/", который ставится сразу после открывающей угловой скобки закрывающего тега. Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

<B><I>На этот текст воздействуют два тега</I></B>

Кроме того, тег может включать атрибут, дающий дополнительную информацию браузеру. Например, при помощи атрибута можно попросить браузер изменить величину шрифта, ориентацию изображения по отношению к строке следующего за ним текста, поменять цвет фона документа и т. д. В парных тегах атрибуты добавляются только к открывающему тегу. Атрибуты представляют собой дополнительные ключевые слова, отделяемые от ключевого слова, определяющего тег, и от других атрибутов пробелами и размещаемые до завершающего тег символа ">". Способ применения некоторых атрибуто требует указания значения атрибута. Значение атрибута отделяется от ключевого слова атрибута символом "=" (знак равенства) и заключается в кавычки.

<H1 ALIGN="LEFT">

Язык HTML в большинстве случаев совершено равнодушен к регистру, в котором набираются теги. Скажем, браузеру совершенно все равно, наберете вы тег, служащий для рисования горизонтальной линии, как **<HR>** или **<hr>** - эффект будет один и тот же.

HTML не признает никакого дополнительного форматирования текста, кроме как с помощью тегов. В результате текст, превосходно смотрящийся в текстовом редакторе, в окне браузера сольется в единую нечитаемую массу.

Так, на месте нескольких пробелов будет лишь один пробел. Исчезнут все заголовки, пустые строки, деление текста на абзацы. Без HTML -тегов браузер просто игнорирует все элементы форматирования.

Определение HTML как языка разметки основывается на том, что при удалении из документа всех тегов получается текстовый документ, совершенно эквивалентный по содержанию исходному гипертекстовому документу. Таким образом, при отображении документа HTML сами теги не отображаются, но влияют на способ отображения остальной части документа.

<HTML>
<HEAD>

Структура HTML-документа

<TITLE> Моя домашняя страница </TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>

Первый тег, который вы здесь видите, **<HTML>**, сообщает браузеру о том, что он имеет дело именно с документом в формате HTML. Тег **<HTML>** и парный ему закрывающий тег **</HTML>** можно считать как бы "конвертом", в который помещается весь документ. Любой HTML -документ состоит из заголовка, который задается при помощи тега **<TITLE>**, и тела документа, который определяется тегом **<BODY>**. В заголовке документа размещается служебная информация, комментарии автора и заголовок страницы, заключаемый в теги **<TITLE>**. Заголовок, вписанный между тегами **<TITLE>**, в основное

текстовое поле браузеру не попадает, а, как правило, размещается в заголовке окна браузера. HTML-документ - это просто текстовый файл с расширением ***.htm** (Unix-системы могут содержать файлы с расширением ***.html**). Большинство элементов языка HTML описывает части содержания документа и помещается между тегами **<BODY>** и **</BODY>**, то есть, внутрь структурного элемента BODY. Такие элементы делят на блочные и текстовые. Блочные элементы относятся к частям текста уровня абзаца. Текстовые элементы описывают свойства отдельных фраз и еще более мелких частей текста. Теперь можно сформулировать правила вложения элементов:

1. Элементы не должны пересекаться. Другими словами, если открывающий тег располагается внутри элемента, то и соответствующий закрывающий тег должен располагаться внутри этого же элемента.
2. Блочные элементы могут содержать вложенные блочные и текстовые элементы.
3. Текстовые элементы могут содержать вложенные текстовые элементы.
4. Текстовые элементы не могут содержать вложенные блочные элементы.

Строго говоря, все правила языка HTML можно рассматривать исключительно как "пожелания". Средство, используемое для отображения Web-документа, сделает все возможное, чтобы истолковать разметку наиболее разумным образом. Тем не менее, гарантию правильного воспроизведения документа дает только неукоснительное следование требованиям спецификации языка.

Функциональные блочные элементы. В большинстве документов основными функциональными элементами являются заголовки и абзацы. Язык HTML поддерживает шесть уровней заголовков. Они задаются при помощи парных тегов от **<H1>** до **<H6>.** При ото бражении Web-документа на экране компьютера эти элементы показываются при помощи шрифтов разного размера.

Обычные абзацы задаются с помощью парного тега **<P>**. Язык HTML не содержит средств для создания абзацного отступа ("красной строки"), поэтому при отображении на экране компьютера абзацы разделяются пустой строкой.

Закрывающий тег **</P>** рассматривается как необязательный. Подразумевается, что он стоит перед тегом, который задает начало очередного абзаца документа. Например:

```
<h1>Заголовок</h1>
<p>Первый абзац
<p>Второй абзац
<h2>Заголовок второго уровня</h2>
```

Заголовок

Первый абзац

Второй абзац

Заголовок второго уровня

Следствием наличия специального тега, определяющего абзац, является тот факт, что обычного символа конца строки, вводимого по нажатию клавиши ENTER, для создания абзацного отступа недостаточно. Язык HTML рассматривает символы конца строки и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении документа рассматривается как одиночный пробел. Это, в частности, означает, что символ конца строки даже не осуществляет перехода на новую строку
(для этой цели используется текстовый элемент, задаваемый непарным тегом **
**).

Текстовое оформление страниц. С помощью тега **<font>** можно изменить параметры шрифта. Для тега используются следующие параметры: **face**, **size** и **color**.

Параметр **Face** служит для задания гарнитуры шрифтов использующихся для текста. Названий шрифтов можно указать несколько, через запятую. В этом случае, если первый указанный шрифт не будет найден, будет использоваться следующий по списку.

Пример 1. Использование параметра face

```
<font face="Arial, Helvetica, sans-serif">Текст будет написан шрифтом
Arial.</font>
```

Size задает размер шрифта в условных единицах от 1 до 7. Средний размер, используемый по умолчанию принят 3. Размер шрифта можно указывать как абсолютной величиной (например, size=4), так и относительной (например, size=+1, size=-1). В последнем случае размер изменяется относительно базового.

Пример 2. Задание размера шрифта

```
<font size=1>Шрифт размера 1</font><br>
<font size=2>Шрифт размера 2</font><br>
<font size=3>Шрифт размера 3</font><br>
<font size=4>Шрифт размера 4</font><br>
<font size=5>Шрифт размера 5</font><br>
<font size=6>Шрифт размера 6</font><br>
<font size=7>Шрифт размера 7</font><br>
```

Шрифт размера 1
Шрифт размера 2
Шрифт размера 3
Шрифт размера 4
Шрифт размера 5
Шрифт размера 6
Шрифт размера 7

Color определяет цвет текста, который можно задавать с помощью названий цветов или в шестнадцатеричном формате.

Пример 3. Изменение цвета текста

<font size=5 color=red
face=Arial>Первая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.

Первая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.

Видоизменение текста - средства его форматирования, такие как выбор начертания шрифта и использование эффектов, позволяющих менять вид текста. В таблице перечислены основные теги, которые применяются для изменения оформления текста.

Код HTML	Описание	Пример
Текст	Жирный текст	**Текст**
Текст	Курсивное начертание текста	*Текст*
<u>Текст</u>	Подчеркнутый текст	Текст
^{Текст}	Верхний индекс	$e=mc^2$
_{Текст}	Нижний индекс	H_2O
<strike>Текст</strike>	Зачеркнутый текс	~~Текст~~
<pre>Текст</pre>	Текст пишется как есть, включая все пробелы	Текст
Текст	Курсивный текст	*Текст*
Текст	Жирный текст	**Текст**

Обычно для создания верхнего или нижнего индекса используется тег **small**, делающий индекс меньше по размеру основного шрифта.

Пример 4. Создание нижнего индекса

Формула серной кислоты:
H_{<small>2</small>}SO_{<small>4</small>}

Формула серной кислоты:

H_2SO_4

Выравнивание текста

Выравнивание текста определяет его внешний вид и ориентацию краев абзаца и может выполняться по левому, правому краю, по центру или по ширине.

Код HTML	Описание	Пример
Текст	Добавляет новый параграф по умолчанию выравненный по левому краю. Перед параграфом автоматически добавляется новая строка.	Текст
Текст	Выравнивание по левому краю	Текст
Текст	Выравнивание по правому краю	Текст
Текст	Выравнивание по центру	Текст
Текст	Выравнивание по ширине	Текст

Задание 1

Оформите текст, как показано ниже:

Октябрь уж наступил,

Уж роща отряхает

Последние листы

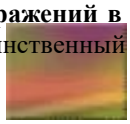
С нагих своих ветвей.

Дохнул осенний хлад,

Дорога промерзает,

Журча, еще бежит

Вставка изображений в html-страницы. Для встраивания изображения в документ используется тег **IMG**, имеющий единственный обязательный параметр **src**, который определяет адрес файла с картинкой.



Файл с рисунком, изображенным ниже, называется *sample.gif* и размещается в папке *images* корня сайта.

Для указания адреса изображения можно задавать как абсолютный, так и относительный адрес.

Пример 1. Вставка изображения в документ

```
<html>
<body>
<img src="http://www.htmlbook/images/sample.gif"> - это абсолютный адрес размещения изображения
<img src="/images/sample.gif"> - адрес размещения изображения относительно корня сайта
<img src="images/sample.gif"> - адрес размещения изображения относительно текущего HTML-документа
</body>
</html>
```

Выравнивание изображений. Для изображений можно указывать их положение относительно текста или других изображений на веб-странице. Способ выравнивания изображений задается параметром **align** тега **IMG**. В таблице перечислены возможные значения этого параметра и результат его использования.

Код HTML	Описание
1	2
<img src=”HLPBELL.GIF” align=**texttop**>	Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки.
<img src=HLPBELL.GIF align=**top**>	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.
<img src=HLPBELL.GIF align=**absmiddle**>	Выравнивание изображения по середине текущей строки.
<img src= HLPBELL.GIF align=**baseline**>	Выравнивание изображения по базовой линии текущей строки.
<img src= HLPBELL.GIF align=**bottom**>	Выравнивание нижней границы изображения по окружающему тексту.
<img src= HLPBELL.GIF align=**left**>	Выравнивание изображения по текущему краю окна.
<img src= HLPBELL.GIF align=**right**>	Выравнивание изображения по правому краю окна.

Наиболее популярные параметры – **left** и **right**, создающие обтекание текста вокруг изображения. Чтобы текст не прилегал плотно к рисунку, рекомендуется в теге **IMG** добавить параметр **hspace** и **vspace**, задающих расстояние до текста в пикселах.

Пример 2. Обтекание текста вокруг рисунка

```
<html>
<body>
<img src=”HLPBELL.GIF” width=50 height=50 hspase=10 vspase=10 align=left> Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy nibh euismod tincidunt ut lacreet dolore magna aliguam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

</body>
</html>
```

Задание 2.

Выровняйте рисунки и текст так, как показано ниже:

Этот рисунок находится слева и текст выравнивается по левой гра-
нице. Эффект достигается без таблицы

Этот рисунок находится слева и

текст выравнивается по центру. Эффект достигается без таблицы

Этот рисунок находится справа, и текст выравнивается по правой границе. Эффект достигается без таблицы.

Практическое занятие №8

Создание гиперссылок

Цель работы: получить навыки работы с абсолютными, относительными и внутренними ссылками.

Основные понятия

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега A, который имеет единственный параметр **href**. В качестве значения используется адрес документа (**URL**).

Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка.

Пример 1. Использование абсолютных ссылок

```
<html>
<body>
<a href=www.yandex.ru>Поисковая система Яндекс</a>
</body>
</html>
```

Относительные ссылки, как следует из их названия, построены относительно текущего документа или адреса. Примеры таких адресов:

1. /
2. /demo/
3. /images/pic.gif
4. ../help/me.html
5. manual/info.html

Первые две ссылки называются неполные и указывают веб-серверу загружать файл index.html (или default.html) находящемуся в корне сайта (пример 1) или папке demo (пример 2). Если файл index.html отсутствует, браузер, как правило, показывает список файлов, находящихся в данном каталоге. Слэш перед адресом говорит о том, что адресация начинается от корня сайта (пример 2),

двоеточие - перейти на уровень выше в списке каталогов сайта (пример 4).

Пример 2. Использование относительных ссылок

```
<html>
<body>
<a href=images/xxx.jpg>Посмотрите на мою фотографию!</a><br>
<a href=tip.html>Как сделать такое же фото?</a>
</body>
</html>
```

Ссылки внутри страницы. Большие документы читаются лучше, если они имеют оглавление со ссылками на соответствующие разделы. Для создания ссылки следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи параметра name тега **A**.

Пример 1. Создание внутренней ссылки

```
<html>
<body>
<a name=top></a> Друг уронил утюг в унитаз. И разбил его. Причем так разбил, что по назначению унитаз и использовать никак нельзя. Мгновением назад только что вот все было хорошо и вот уже дыра, да такая, что можно забыть, что есть такой предмет в доме. Махнул рукой нечаянно, а потом мучайся...
<a href=#top>Наверх</a>
</body>
</html>
```

Между тегами **** и **** отсутствует текст, так как требуется лишь указать местоположение перехода по ссылке, находящейся внизу страницы. Имя ссылки на закладку начинается символом # , после чего идет название закладки. Название выбирается любое, соответствующее тематике.

Можно, также, делать ссылку на закладку, находящуюся в другой вебстранице и даже другом сайте. Для этого в адресе ссылки надлежит указать ее адрес и в конце добавить символ решетки # и имя закладки.

Пример 2. Ссылка на закладку из другой веб-страницы

```
<html>
<body>
<a href=text.html#bottom>Перейти к нижней части текста</a>
</body>
</html>
```

Ссылка на новое окно

Если требуется сделать ссылку на документ, который открывается в новом окне браузера, используется параметр **target=_blank** тега **A**.

Создание нового окна обычно требуется в случаях, когда делается ссылка на другой сайт, в остальном лучше открывать документы в текущем окне, поскольку обилие окон может сбить читателя с толку.

Так как ссылки на текущее или новое окно ничем не отличаются друг от друга, на некоторых сайтах рядом со ссылкой ставят специальную иконку, показывающую, что документ открывается в новом окне.

Пример 1. Создание ссылки на новое окно

```
<html>
<body>
<a href=www.bsu.edu.ru>Обычная ссылка на сайт www.bsu.edu.ru</a><br>
<a href=www.bsu.edu.ru target=_blank>Ссылка открывает новое окно на сайт
www.bsu.edu.ru</a>
</body>
</html>
```

Обычная ссылка на сайт www.bsu.edu.ru

Ссылка открывает новое окно на сайт www.bsu.edu.ru

Задание 1.

Используя внутренние ссылки, создать следующий словарь терминов:

Словарь терминов

А Б В Г Д Е

А

АВТЕНТИЧЕСКИЙ КАДАНС

кадансовый оборот, в котором заключительная тоническая гармония предваряется доминантовой

АЛИКВОТНЫЕ СТРУНЫ

резонирующие струны, к которым исполнитель не прикасается во время игры

АТАКТА

гармонический элемент на басу нижнего или верхнего вводного тона

В начало

Б

БАГАТЕЛЬ

небольшая нетрудная для исполнения пьеса

БАРТОКОВСКОЕ ПИЦЦИКАТО

сильный щипок струны с последующим ударом струны о гриф

БОНАНГ

набор из 10-12 гонгов разного размера

В начало

В

В начало

Г

В начало

Д

В начало

Список рекомендуемой литературы

1. Исакова А. И. Информационные технологии [Электронный учебник]: учебное пособие / Исакова А. И.. - Эль Контент, Томский государственный университет систем управления и радиоэлектроники, 2012. - 174 c. - Режим доступа: http://iprbookshop.ru/13938

2. Тимченко С. В. Информатика [Электронный учебник] : учебное пособие / Тимченко С. В.. - Эль Контент, Томский государственный университет систем управления и радиоэлектроники, 2011. - 160 c. - Режим доступа: http://iprbookshop.ru/13935

3. Власова Е. З. Информационные технологии [Электронный учебник] : учебно-методическое пособие / Власова Е. З.. - Российский государственный педагогический университет им. А.И. Герцена, 2011. - 251 c. - Режим доступа: http://www.iprbookshop.ru/19321

4. Губарев В. В. Информатика [Электронный учебник] : прошлое, настоящее, будущее Учебник / Губарев В. В.. - Техносфера, 2011. - 432 c. - Режим доступа: http://iprbookshop.ru/13281