

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

**СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**сборник учебно-методических материалов**

по специальности

37.05.01 – Клиническая психология

2017 г.

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

*Составители:* Попова А.М., Лебедь О.А.

**Современные информационные технологии:** сборник учебно-методических материалов по специальности 37.05.01 – Клиническая психология. – Благовещенск: Амурский гос. ун-т, 2017.

© Амурский государственный университет, 2017

© Кафедра общей математики информатики, 2017

© Попова А.М., Лебедь О.А., составление

## *ВВЕДЕНИЕ*

Методические указания предназначены для проведения лабораторных и практических занятий по дисциплине «Современные информационные технологии» для студентов факультета социальных наук по специальности «Клиническая психология».

Цель занятий состоит в приобретении практических навыков работы в Microsoft Excel.

Процесс по освоению всей совокупности теоретического и практического материала по дисциплине должен быть реализован в течение одного семестра, необходима проработка теоретического материала лекций.

Для закрепления полученных на занятии знаний студенты выполняют соответствующие упражнения и получают домашние задания. Полученные оценки за выполненные упражнения и домашние задания являются основой для выставления промежуточной и итоговой аттестации. Итоговой аттестацией в семестре является зачет.

Контроль выполненных заданий осуществляется либо непосредственно на занятиях, либо на консультациях.

Рейтинговая система студента по дисциплине складывается из оценки за работу в семестре – максимально 60 баллов и зачетной оценки – максимально 40 баллов. Минимальное значение рейтинговой оценки, набранной студентом по результатам текущего контроля по всем видам занятий, при которой студент допускается к сдаче зачета, составляет 40 баллов.

Если к моменту проведения зачета студент набирает 51 и более баллов, они могут быть выставлены ему в виде поощрения в ведомость и в зачетную книжку без процедуры принятия зачета. Устранение задолженности студента по отдельным контролируемым темам дисциплины может проходить в течение семестра в часы дополнительных занятий или консультаций, установленных в расписании по соответствующим видам занятий по данной дисциплине.

## 1 КРАТКОЕ ИЗЛОЖЕНИЕ ЛЕКЦИОННОГО МАТЕРИАЛА

Лекция – одна из базовых форм обучения обучающихся. С помощью лекций, которые читаются профессорами, доцентами, старшими преподавателями, обучающиеся знакомятся с основными научно-теоретическими и практическими положениями, проблемами того или иного учебного курса, получают направление и рекомендации по самостоятельной работе с учебниками, монографиями, учебными пособиями и первоисточниками. Лекция, особенно проблемного характера, дополняет учебники и учебные пособия. Она оказывает существенное эмоциональное влияние на обучающихся, пробуждает мысль, формирует интерес и желание глубоко разобраться в освещаемых лектором проблемах.

### Тема 1: Электронные таблицы Microsoft Excel.

#### План:

1. Стандартные функции.
2. Списки и базы данных.
3. Форма для работы со списками. Сортировка списков.
4. Поиск с помощью автофильтра. Поиск с помощью расширенного фильтра.
5. Анализ данных.
6. Автоматическое подведение итогов.
7. Мастер сводных таблиц.
8. Консолидация данных: по расположению; по категории.

**Цель:** изучить возможность MS Excel, позволяющие производить слияние данных, с целью упрощения однотипной работы и автоматизации процесса создания электронных документов.

#### Задачи:

- умение выполнять слияние данных,
- подготавливать базу данных необходимую для выполнения данного процесса.

#### Ключевые вопросы:

Некоторый диапазон таблицы можно рассматривать как базу данных. Столбцы этой таблицы называются полями, а строки называются записями.

Существуют ограничения, накладываемые на структуру базы данных:

- первый ряд базы данных должен содержать неповторяющиеся имена полей;
- остальные ряды базы данных содержат записи, которые не должны быть пустыми рядами;
- информация по полям (столбцам) должна быть однородной, т.е. только цифры или только текст.

Основная функция любой базы данных – поиск информации по определенным критериям. С увеличением количества записей поиск определенной информации затрудняется. Microsoft Excel позволяет упростить этот процесс путем фильтрации данных.

#### Фильтрация данных

Основное назначение любой базы данных – это оперативный поиск необходимой информации по какому-либо запросу. При этом часть базы данных, удовлетворяющая запросу, называется *выборкой*.

Запросы в Microsoft Excel реализованы с помощью фильтров. *Фильтрация* списка – это процесс, в результате которого в списке скрываются все строки, не удовлетворяющие критериям фильтрации, а видимыми остаются только те, которые соответствуют условиям запроса (остаётся выборка).

Microsoft Excel располагает двумя инструментами фильтрации: *автофильтром* и *расширенным фильтром*. С помощью автофильтра реализуются простые запросы, содержащие не более двух условий поиска. Расширенный фильтр позволяет выполнять запросы практически любой сложности.

Для отключения инструмента *Автофильтр* следует из меню *Данные* выбрать команду *Фильтр* и щелкнуть на команде *Автофильтр*.

Кроме того, при создании текстовых критериев можно использовать символы шаблона: символ «\*» для обозначения последовательности произвольной длины, состоящей из любых символов; символ «?» для обозначения одного символа, стоящего на определенном месте.

### Расширенный фильтр

Команда *Расширенный фильтр* позволяет отыскивать строки с помощью более сложных критериев.

При использовании *Расширенного фильтра* необходимо сначала определить (создать) три области:

1. Исходный диапазон – это область базы данных.
2. Диапазон условий отбора (или интервал критериев) – это область, где задаются критерии фильтрации.
3. Диапазон, в который при желании пользователя Microsoft Excel помещает результат выборки (интервал извлечения) – это та область, в которой будут появляться результаты фильтрации.

Имена полей во всех интервалах должны точно совпадать.

Для выполнения действий по фильтрации необходимо выполнить команды *Данные* → *Фильтр* → *Расширенный фильтр*. В диалоговом окне *Расширенный фильтр* необходимо указать координаты интервалов (рис. 1).

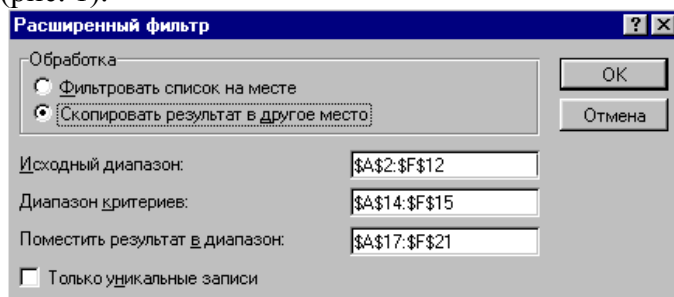


Рис. 1. Диалоговое окно *Расширенный фильтр*

Если необходимо получать результаты фильтрации в интервале извлечения, нужно поставить [\*] – Скопировать результат в другое место.

### Консолидация данных

С помощью команды консолидация можно собрать данные из нескольких исходных рабочих листов в одном итоговом листе. При этом исходные листы могут располагаться в той же самой книге, где находится итоговый лист, или в других книгах.

Команду консолидация можно использовать несколькими способами:

1. Можно связать консолидированные данные с исходными данными, чтобы последующие изменения в исходных листах отражались в итоговом листе.
2. Можно просто консолидировать исходные данные без создания связи.

Консолидировать данные можно:

1. По расположению.
2. По категории.

Если консолидировать данные по расположению, программа выбирает информацию из одинаково расположенных ячеек всех исходных листов. При консолидации данных по категории программа использует в качестве основы для объединения листов заголовки столбцов или строк.

Можно консолидировать листы, используя любую функцию в поле со списком «функция»: количество значений, среднее значение, максимум, минимум, произведение, количество чисел и т.д.

Книга, содержащая лист, в который помещаются консолидированные данные, должна быть открыта, но исходные книги могут быть закрыты при условии, что даны правильные сведения о местонахождении, иначе программа не сможет найти все файлы этих книг. Прежде, чем начать консолидацию, необходимо сохранить все исходные книги.

Консолидация предполагает использование следующей методики:

1. Указать местоположение будущих консолидированных данных.

2. Выбрать команду данные→консолидация.
3. В открывшем окне указать диапазоны данных, подлежащие консолидации.
4. Указать способ консолидации.
5. Выбрать тип консолидации, т.е. указать, какая операция будет производиться с консолидируемыми данными.
6. При необходимости указать добавление структуры – создать связи с исходными данными.

#### Сводные таблицы

Сводные таблицы представляют собой средства для группировки, обобщения и анализа данных, находящихся в списках Excel или в таблицах, созданных в других приложениях. Внешне сводные таблицы являются структурой, позволяющей размещать данные в трехмерном виде. Сводные таблицы могут использоваться:

1. Для обобщения большого количества однотипных данных.
2. Для реорганизации данных (с помощью перетаскивания).
3. Для отбора и группировки данных.
4. Для построения диаграмм.

Сводные таблицы создаются с помощью мастера сводных таблиц (данные→сводная таблица) по следующей методике:

1. Выбрать место для сводной таблицы, т.е. установить указатель ячейки в необходимое место на рабочем листе.

2. Выполнить команду данные→сводная таблица.

3. Задание типа источника данных и вида создаваемого отчета.

Тип источника данных:

- a) В списке или базе данных Excel.
- b) Во внешнем источнике данных.
- c) В нескольких диапазонах консолидации.
- d) В другой сводной таблице или сводной диаграмме.

Вид отчета:

- a) Сводная таблица.
- b) Сводная диаграмма (со сводной таблицей)

4. Указание местонахождения исходных данных.

5. Указания местонахождения будущей сводной таблицы.

Поместить таблицу:

- a) Новый лист.
- b) Существующий лист.

6. Прежде, чем указать местоположение будущей таблицы, необходимо нажать кнопку «макет» и в открывшемся окне сформировать макет сводной таблицы (т.е. задать страницу, строки, столбцы, итоговые и вычисляемые поля сводной таблицы).

7. Для определения необходимой операции для полей, помещенных в область «данные», либо задание вычисляемого поля дважды щелкнуть левой кнопкой на поле, помещенном в область «данные», и выбрать необходимые действия в окне. Вычисление поля сводной таблицы.

Операции: сумма, количество, среднее значение, максимальное значение, минимальное значение, произведение, количество чисел и т.д.

Параметры => установить необходимые параметры сводной таблицы.

Создание сводной таблицы (готово).

При создании, редактировании и работе со сводными таблицами необходимо учитывать следующее:

1. Местоположение сводной таблицы – желательно располагать ее на отдельном листе, т.к. при обновлении, группировках сводной таблицы информация, содержащаяся на рабочих листах рядом со сводной таблицей, может оказаться скрытой.

2. Местонахождение исходных данных – список Excel, внешний источник данных, диапазоны консолидации, находящиеся в другой сводной таблице

3. Необходимо при задании структуры сводной таблицы определить:

- а) поля, находящиеся в строках и столбцах таблицы;
- б) поля, по которым подводятся итоги (с выбором необходимой операции);
- с) поля для страниц, что позволяет представить информацию в трехмерном виде;

4. Сводная таблица – это средство только для отображения данных. Поэтому в самой таблице данные редактировать нельзя. Для изменения данных в сводной таблице необходимо внести изменения в источник данных, а затем обновить сводную таблицу (**кнопка !** – обновить данные панели инструментов сводной таблицы)

5. В сводных таблицах можно изменять названия полей, что не влечет изменений в полях исходных данных.

6. Сводные таблицы допускают возможность группировки элементов полей по различным уровням иерархии путем объединения в группы. Данные → **Группа и структура** => 2 кнопки:

#### **Группировать и Разгруппировать.**

Применяются следующие варианты группировки сводной таблицы:

- группировка элементов по их именам;
- группировка чисел по диапазонам;
- группировка по временным диапазонам.

7. Возможность построения диаграмм на основе сводных таблиц;

8. Сортировка элементов в сводной таблице;

9. Управление общими и промежуточными итогами;

10. Использование различных итоговых функций для анализа данных и дополнительных вычислений;

11. Использование автоформата для форматирования сводной таблицы.

## **Тема 2: База данных Microsoft Access.**

### **План:**

1. Понятие базы данных.
2. Модели организации данных.
3. Язык SQL.
4. Системы управления базами данных.
5. Основные понятия СУБД Access: поле данных, ключ поля данных, схема данных, таблицы, формы, запросы, отчеты.

**Цель:** изучить основные понятия реляционных БД и средства автоматизации проектирования БД.

### **Задачи:**

- Определить основные понятия БД, реляционных БД
- Выделить модели организации.
- Охарактеризовать основные методы и технологии управления БД.

### **Ключевые вопросы:**

#### Типы полей доступные в конструкторе:

*Текстовый:* используется для полей, не участвующих в вычислениях, размер не превышает 255 символов.

*Поле MEMO:* используется, когда необходимо ввести более 255 символов.

*Числовой:* используется в случаях, когда необходимо производить вычисления.

*Дата/время:* с его помощью можно выполнять действия над датами.

*Денежный:* числовой тип с фиксированным числом разрядов после запятой и указанием денежной единицы.

*Счетчик:* используется для автоматической нумерации записи.

*Логическое:* используется для хранения данных Boolean: Да и Нет.

*Поле объекта OLE:* используется для рисунков, звуковых файлов или графиков.

*Мастер постановок:* используется для создания поля, которое дает возможность выбирать значения из полей в другой таблице.

#### Свойства полей.

*Формат поля:* доступен для числовых, денежных полей, поля дата/время для определения формата вывода на экран.

*Маска ввода:* задает маску ввода, облегчающую ввод данных в элемент управления- поле. Например, удобно создать следующую маску ввода для поля «Телефон», позволяющую вводить только цифры и автоматически добавляющую промежуточные символы. Значение данного свойства определяется автоматически при использовании мастера по созданию масок ввода.

*Подпись:* задает заголовок столбца при выводе таблицы.

*Значение по умолчанию:* позволяет указать значение, автоматически вводимое в поле при создании новой записи, но доступное изменению.

*Условие на значение:* определяет требования к данным, вводимым в запись, в поле или в элемент управления.

*Сообщение об ошибке* позволяет указать текст сообщения, выводимого на экран, если введенные данные нарушают условие, определенное в свойстве Условие на значение.

*Обязательное поле:* определяет, необходимо ли вводить в поле значение.

#### *Запросы в ACCESS*

В СУБД MS Access существуют следующие виды запросов:

*запрос на выборку* – позволяет выбирать данные из взаимосвязанных таблиц и других запросов;

*запросы с параметром* – критерий отбора может задать сам пользователь, введя нужный параметр при вызове запроса;

*перекрестные запросы* – позволяют создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц;

*запросы на обновление, добавление и удаление* – являются запросами действия, в результате работы которых изменяются данные в таблицах;

*итоговые запросы* – производят математические вычисления по заданному полю и выдают результат.

#### *Конструктор запросов*

В отличие от мастера простых запросов в конструкторе запросов можно вводить критерии отбора, как это делается в фильтрах.

Для этого используются операторы:

<, >, <=, >=, <> – для задания сравнения;

In (значение1, значение2, ...) – для проверки включения значения в список;

Between...and – для проверки вхождения в интервал значений;

And, or, not – для проверки условия;

Like – вводимое значение сравнивается с образцом;

? – заменяет один любой символ;

\* – заменяет произвольное количество любых символов;

# – заменяет любую цифру;

! после первой скобки – для поиска символа, который не входит в указанный набор символов.

Ниже в таблице приведены примеры задания условных выражений.

*Таблица 1*

Условные выражения	Вывод записей, удовлетворяющих условию
За?ор	Забор, Затор
Д*нь	День, Добрый день, Длинная тень
*й	128й, Последний
#-й	5-й, 8-й
Иванов[!аы]	Иванову [но не Иванова или Ивановы]
Саратов	имеют значение Саратов
Not Саратов	не имеют значение Саратов
In (Саратов, Томск, Уфа)	имеют значение Саратов, Томск или Уфа



<M	начинаются с букв А-Л
>=M	начинаются с букв М-Я
100	имеют числовое значение 100
<=20	имеют числовое значение 20
Date()	имеют значение текущей даты
>=01.01.94	имеют значение даты позднее 01.01.94
Between 01.01.93 AND 31.12.93	имеют значение года 1993
*.02.*	имеют значение месяца Февраль
Null	содержат в поле пустое значение
Is Not Null	имеют не пустое значение в поле
Like "P*"	начинаются с буквы P

#### *Универсальные запросы с параметрами*

Для постоянного изменения какого-либо условия в запросе можно в окне конструктора в строке «Условия отбора» поля удалить текст, а затем либо непосредственно, либо в окне «Область ввода» (клавиши SHIFT+F2) задать подсказку в виде [Введите ...]. Параметр служит подсказкой.

#### *Вычисление по полям*

Чтобы произвести вычисление, запрос открывают в режиме КОНСТРУКТОРА и в строке ПОЛЕ вводится математическое выражение. Например, для вычисления среднего балла экзаменационных оценок запишем:

*сумма*: ([сессия]![оценка1]+[оценка2] +[оценка3])/3 , где *сумма* – название будущего столбца; [сессия]![оценка1] – имя таблицы, заключенная в квадратные скобки, восклицательный знак, затем имя поля.

Для вычисления используются математические операторы.

#### *Отчеты в ACCESS*

Отчеты предназначены для форматированного вывода данных на печать. Источниками данных для отчетов служат таблицы, запросы или инструкции SQL. Отображаемая информация автоматически изменяется при изменении данных в таблицах, на которых она базируется. Однако формат отчета сохраняется и изменяется только в том случае, когда макет отчета будет изменен.

Основные *виды* отчетов:

одноколонный (простой) отчет;

многоколонный отчет;

табличный отчет;

отчет с группировкой данных и подведением итогов;

отчет по связанным таблицам;

связанный отчет, т.е. отчет, содержащий другой (подчиненный отчет);

отчет слиянием с документом Word (составной документ);

перекрестный отчет.

Основные *разделы* отчета:

заголовок отчета (начало отчета);

верхний колонтитул (печатается в начале каждой страницы);

область заголовка группы (отображается перед первой записью каждой группы);

область данных (основная часть отчета);

область примечания группы (отображается после области данных последней записи каждой группы);

нижний колонтитул (печатается в конце каждой страницы);

область примечаний (печатается в конце отчета).

Отчеты можно создавать несколькими способами:

с помощью автоотчета (пользователь выбирает только источник записей и макет документа);

с помощью мастера отчетов (традиционная методика пошагового создания отчетов);

с помощью конструктора отчетов (отчет полностью формируется пользователем).

#### *Создание автоотчета*

Для создания автоотчета необходимо выбрать *Отчеты* → *Создание отчета с помощью мастера*. Так же необходимо выбрать таблицу или запрос, которые будут выступать в качестве источника данных.

При создании отчета доступны следующие методы:

*Конструктор*. Новый отчет создается вручную.

*Мастер отчетов*. Мастер MS Access сопровождает процесс создания отчета.

*Автоотчет: в столбец*. Создается отчет, который отображает поля из таблицы в одном столбце.

*Автоотчет: ленточный*. Создается отчет, который отображает данные в табличном формате, аналогичном электронной таблице.

*Диаграмма*. Мастер сопровождает процесс вставки диаграммы в отчет.

*Почтовые наклейки*. Мастер сопровождает процесс создания отчета, форматированного для печати почтовых наклеек.

#### *Создание отчета с помощью мастера*

Мастер создания отчетов работает в восемь этапов:

*1 этап*. Выбор базовых таблиц или запросов, на которых базируется отчет.

*2 этап*. Выбор полей, отображаемых в отчете.

*3 этап*. Выбор вида представления данных.

*4 этап*. Выбор полей группировки.

*5 этап*. Выбор порядка сортировки и вычисления, выполняемые для записи.

*6 этап*. Выбор вида макета для отчета.

*7 этап*. Выбор требуемого стиля.

*8 этап*. Задание имени отчета.

*Задание 19*. Создать отчет, основанный на запросе *Для отчета*. Поля для отчета выбрать в следующей последовательности: номер, Фамилия, Имя, Отчество, город, группа, пол, дата рождения. Вид представления данных и группировку оставить без изменения. Сортировку осуществить по полю *Номер* по убыванию. Вид макета выбрать *Структура 1*. Стилль выбрать *Обычный*. Сохранить отчет с именем *Студент*.

#### *Формы в ACCESS*

Для исследования возможностей мастера форм используем таблицу СТУДЕНТЫ из БД SESS.mdb.

#### *Конструктор формы. Элементы управления*

Для создания нового поля уже в существующей форме (например, проживает в общежитии или нет) познакомимся с элементами управления.

*Элементы управления* — это графические объекты, размещенные в форме или отчете и предназначенные для изображения данных, выполнения операций или просто для красоты.

Наиболее распространенная разновидность элементов управления — текстовое поле, которое позволяет размещать текст или цифры. Его можно использовать для ввода новых данных или редактирования имеющихся. Но для выполнения поставленной задачи нас больше интересует флажок или переключатель.

Рассмотрим вид окна формы в режиме конструктора (рис. 2).

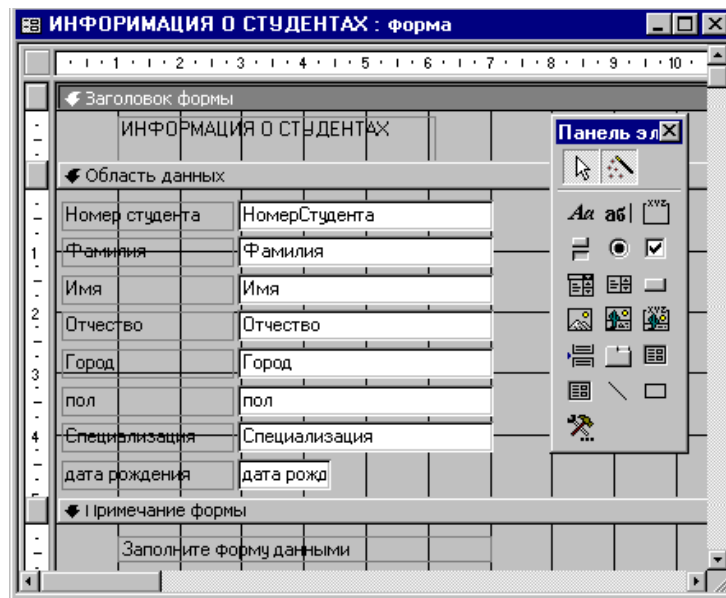


Рис. 2. Форма «Информация о студентах»

Предположим, мы хотим, чтобы результат использования переключателя в форме отражался в исходной таблице. Для этого предварительно нужно создать в таблице новое поле.

### Тема 3: Поиск информации в Интернет.

#### План:

1. Сравнение поисковых каталогов и указателей.
2. Основные зарубежные поисковые системы.
3. Отечественные поисковые системы Aport, Rambler, Яндекс, Mail.
4. Контекстный поиск. Поиск Web-узлов. Поиск ссылок.

**Цель:** изучение структуры глобальной сети Интернет и освоение основных методов поиска в сети;

#### Задачи:

- познакомить с понятием «Интернет», информации в сети «Интернет»;
- познакомиться с типами поисковых систем в сети Internet;
- научиться эффективным приемам поиска информации во всемирной паутине ;
- сравнить поисковые каталоги и указатели.

#### Ключевые вопросы:

На сегодняшний день поисковые системы можно разбить на несколько типов:

- тематические каталоги (или предметные каталоги)
- информационно-поисковые машины (или роботы-индексов)
- системы метапоиска.

Краткая характеристика различных типов поисковых систем.

#### Тематические каталоги

очень похожи на библиотечные, информация в них имеет иерархическую структуру и классифицирована по темам:

- искусство,
- наука,
- образование и т.д.

Каталогами пользуются тогда, когда пользователя интересует не что-то конкретное, а широкий круг вопросов, но по одной узкой теме. Их рекомендуется использовать для реферативно-тематических обзоров. Каталоги являются справочниками, содержащими списки адресов Интернет, сгруппированные по определенным признакам (по тематике). Каждая тема разветвляется на несколько подуровней. Особенность этих средств поиска в том, что создание структуры БД и их постоянное обновление осуществляется «вручную» коллективом редакторов и

программистов, да и сам процесс поиска требует непосредственного участия пользователя, самостоятельно переходящего от ссылки к ссылке. Наиболее популярным во все мире признан тематический каталог **Yahoo!**(<http://www.yahoo.com>) . Он представляет собой огромную базу данных адресов сайтов самой различной тематики.

Организация работы с тематическим каталогом достаточно проста: вы выбираете сначала общую тематику по вашему запросу, и далее конкретизируете, следуя подсказкам каталога. Аналогичный русскоязычный каталог **Ay!** (<http://www.au.com>). Также популярен русскоязычный каталог **List.ru** (<http://www.list.ru>).

Создание и поддержка тематических каталогов требует огромных затрат. Характерные особенности поисковых систем:

- малая скорость пополнения БД,
- сравнительно небольшой объем,
- достаточно высокая точность поиска информации.

#### Информационно-поисковые машины (роботы индексов).

К этому типу поисковых систем следует прибегать только тогда, когда ключевые слова точно известны. Например, фамилия человека или несколько специфических терминов из соответствующей области.

Поисковая машина устроена следующим образом:

это сервер с огромной базой данных адресов, которая автоматически пополняется благодаря действиям поискового робота – специальной программы. Программа-робот самостоятельно исследует пространство Интернет, обращается к страницам WWW, изучает содержимое этих страниц, формирует и прописывает ключевые слова с web-страниц в свою БД, т.е. индексирует страницы.

Все ссылки помещаются в БД сервера, в которой пользователи, используя ключевые слова, находят нужную им информацию.

Т.о. по запросу пользователя программы-роботы «перерывают» огромное количество собранных и хранящихся на сервере данных обо всех ресурсах Сети: от крупных серверов до самой маленькой и скромной домашней страницы какого-либо начинающего пользователя. Результат поиска состоит из списка рекомендованных пользователю web-страниц (их адресов, URL), оформленных в виде гиперссылок..

Найденные документы ранжируются в зависимости от местоположения ключевых слов (в заголовке, в начале текста, в первых параграфах) и частоты их появления в тексте.

Различные поисковые машины отличаются по языкам запроса, глубине поиска внутри документа, методам ранжирования и приоритетов.

Поэтому использование разных поисковых машин дает различные результаты поиска.

#### Системы мета-поиска.

Таковыми системами можно воспользоваться исходя из экономии времени и денег. Эти системы не имеют собственных поисковых инструментов и собственной базы данных. Их задача: передать запрос пользователя настоящим поисковым системам. Системы мета-поиска позволяют задействовать сразу несколько поисковых служб, при этом у пользователя нет необходимости подключаться к каждой из этих служб и многократно вводить запрос. Системой такого класса является **Search.com** (<http://www.search.com>).

Российская система мета-поиска «Следопыт» (<http://www.medialingua.ru/www/wwwsearc.htm>). Работает только с 5-ю англоязычными поисковыми машинами и одной русскоязычной. Но интересен тем, что переводит ваш русскоязычный запрос на английский язык для англоязычных поисковых машин. Итак, осуществлять поиск информации с помощью ввода только готового адреса в адресную строку не представляется возможным. Поэтому вопросы поиска информации становятся все более актуальными и с годами проблема будет только усложняться. Найти нужную информацию крайне непросто. Отсутствие единого стандарта на спецификацию задачи поиска в Сети приводит к тому, что на каждом узле предлагается свой собственный способ решения поставленной задачи. В результате пользователь, не имеющий необходимых навыков в создании синтаксических

конструкций запросов к поисковой системе, либо не полностью использует возможности каждого портала, либо, изучив функции конкретного узла, постоянно обращается к одному.

Для получения качественного результата поиска, необходимо использовать возможности различных поисковых систем. Регулярное использование только одной из поисковых систем может плохо отразиться на результатах поиска.

**К золотым правилам поиска** информации в сети можно отнести следующие требования:

- учитывать особенности естественного языка;
- не допускать орфографических ошибок;
- избегать поиска по одному слову, использовать необходимый и достаточный набор слов;
- не писать большими буквами;
- исключать из поиска ненужные слова;
- использовать возможности расширенного поиска.

К особенностям естественного языка можно отнести:

- полисемию (смысловое разнообразие слова или сочетания слов);
- омонимию (разных по смыслу, но одинаково пишущихся слов);
- синонимы (различающихся по написанию и произношению слов, но тождественных по смыслу).

#### **Список поисковых серверов и каталогов**

<b>Адрес</b>	<b>Описание</b>
<a href="http://www.aport.ru">http://www.aport.ru</a>	Апорт – русскоязычный поисковый сервер
<a href="http://www.yandex.ru">http://www.yandex.ru</a>	Яндекс - русскоязычный поисковый сервер
<a href="http://www.rambler.ru">http://www.rambler.ru</a>	Рамблер- русскоязычный поисковый сервер
<a href="http://www.excite.com">http://www.excite.com</a>	Поисковый сервер с обзорами узлов и путеводителями
<a href="http://ww.altavista.com">http://ww.altavista.com</a>	Поисковый сервер, имеются возможности расширенного поиска
<a href="http://www.hotbot.com">http://www.hotbot.com</a>	Поисковый сервер
<b>Справочные ресурсы Интернета</b>	
<a href="http://www.yellow.com">http://www.yellow.com</a>	Желтые страницы Интернета
<a href="http://monk.newmail.ru">http://monk.newmail.ru</a>	Поисковые системы различного профиля
<a href="http://www.allru.net">www.allru.net</a>	Каталог русских ресурсов Интернет
<a href="http://www.ru">www.ru</a>	Каталог русских ресурсов Интернет
<a href="http://www.open.ac.uk">http://www.open.ac.uk</a>	Открытый университет Великобритании
<a href="http://www.ntu.edu">http://www.ntu.edu</a>	Национальный университет США
<a href="http://www.translate.ru">www.translate.ru</a>	Электронный переводчик текстов
<a href="http://www.pomorsu.ru/guide.library.html">www.pomorsu.ru/guide.library.html</a>	Список ссылок на сетевые библиотеки
<a href="http://www.elibrary.ru">www.elibrary.ru</a>	Научная электронная библиотека
<a href="http://www.citforum.ru">www.citforum.ru</a>	Электронная библиотека
<a href="http://www.infamed.com/psy">www.infamed.com/psy</a>	Психологические тесты
<a href="http://www.pokoleniye.ru">www.pokoleniye.ru</a>	Web-сайт Федерации Интернет Образования
<a href="http://Mega.km.ru">Mega.km.ru</a>	Энциклопедии и словари

#### **Тема 4: Создание Web-страницы с помощью языка HTML**

**План:**

1. Теги и атрибуты. Работа с текстом: заголовки, списки (нумерованные, маркированные, определений), комментарии.
2. Таблицы.
3. Графические объекты, создание ссылок при помощи `img` и `gif`-рисунков.
4. Гиперссылки для вызова других Web-страниц.
5. Фреймы.
6. Формы

**Цель:** изучить основные возможности языка HTML для создания Web-страниц и гиперссылок.

**Задачи:**

- изучить структуру документа HTML;
- описание форматирования текста с помощью языка HTML;
- ознакомиться с основными свойствами фреймов.

**Ключевые вопросы:**

Web-страница представляет собой текстовый документ. Экранное представление Web-страницы формируется с помощью специальных команд языка HTML, интерпретируемых браузером. Web-страница отображается как комбинированный документ.

Язык HTML (HyperText Markup Language) – это язык разметки гипертекста. Гипертекст – это расширенный текст, содержащий дополнительные элементы.

HTML представляет собой достаточно простой набор команд, которые описывают структуру документа. Этот язык разметки позволяет выделить в документе отдельные логические части (заголовки, абзацы, списки и т.д.), но не задает конкретные атрибуты форматирования.

Команды языка HTML называются тегами и записываются в виде последовательных букв, заключенных в угловые скобки: <тег>. Теги могут быть одиночными или парными. Парный тег состоит из открывающегося и закрывающегося тега. Закрывающийся тег содержит ту же последовательность букв, но им предшествует косая черта: </тег>. Открывающие теги часто могут содержать атрибуты, влияющие на эффект, создаваемый тегом. Атрибуты – это дополнительные ключевые слова, отделенные от ключевого слова тега и друг от друга пробелами. Закрывающие теги никогда не содержат атрибутов. Если в один тег включается несколько атрибутов, они отделяются от наименования тега и друг от друга одиночными пробелами.

#### *Структура документа HTML*

Все документы HTML имеют строго заданную структуру. Документ должен начинаться с тега <HTML> и заканчиваться соответствующим закрывающим тегом </HTML>. Внутри он состоит из двух разделов – раздела заголовков (парный элемент HEAD) и тела документа (парный элемент BODY).

Минимальный «полный» документ HTML выглядит:

```
<HTML>
  <HEAD>
  <TITLE>
  Моя первая Web - страница
  </TITLE>
  </HEAD>
  <BODY>
  Сегодня учусь составлять HTML документ.
  </BODY>
</HTML>
```

Создав документ в Блокноте, его необходимо сохранить в виде файла с расширением \*.html (\*.htm) и открыть с помощью браузера.

В теге <BODY> можно задавать следующие цветовые атрибуты:

BGCOLOR – определяет фоновый цвет страницы;

TEXT – определяет цвет текста на странице;

LINK – определяет цвет для не просмотренных ссылок;

ALINK – определяет цвет активной ссылки – ссылки, на которую наведен указатель мыши или выделенной с помощью клавиши TAB;

VLINK – определяет цвет просмотренных ссылок, для которых уже осуществлялся переход по щелчку.

#### *Таблица цветов*

Цвет	Значение	Цвет	Значение	Цвет	Значение
Белый (WHITE)	FFFFFF	Желтый (YELLOW)	FFFF00	Серый (GRAY)	808080
Синий (BLUE)	0000FF	Зеленый (GREEN)	00FF00	Темно-синий	000080

				(NAVY)	
Фиолетовый (PURPLE)	FF00FF	Аква (AQUA)	00FFFF	Оливковый (OLIVE)	808000
Красный (RED)	FF0000	Черный (BLACK)	000000	Серебряный (SILVER)	C0C0C0

*Линейки.* Горизонтальные линейки задаются одиночным тегом <HR>. Атрибут ALIGN задает способ выравнивания линейки: LEFT – слева, RIGHT – справа, CENTER – по центру. Атрибут WIDTH определяет ширину (длину) линейки относительно страницы (знак % обязателен, напр.: <HR WIDTH="80%">). Толщину (реальную ширину) определяет атрибут SIZE, его значение задается в пикселях.

*Заголовки.* Язык HTML позволяет задать шесть разных уровней заголовков, которые задаются парными тегами от <H1> до <H6>. Выравнивание заголовков задается атрибутом ALIGN, принимает значения: LEFT, RIGHT, CENTER, JUSTIFY.

<H1> заголовок 1-го уровня </H1>

<H2> заголовок 2-го уровня </H2>

<H3> заголовок 3-го уровня </H3>

<H4> заголовок 4-го уровня </H4>

<H5> заголовок 5-го уровня </H5>

<H6> заголовок 6-го уровня </H6>

*Абзацы.* Отдельный абзац задается с помощью парного тега <P>. Сплошной текст можно разбивать при помощи тега перехода на новую строку <BR>.

*Логические разделы.* Разбиение текста на логические разделы осуществляется при помощи элемента DIV. Этот элемент похож на элемент абзаца, но предназначен для выравнивания группы блочных элементов. Выравнивание задается при помощи атрибута ALIGN, не поддерживается только выравнивание по ширине. Напр.: <DIV ALIGN="CENTER">–выравнивание текста по центру.

*Блочная цитата* описывается при помощи парного тега <BLOCKQUOTE>, т.е. образуется отступ, как у книжных цитат.

*Адресные данные.* Один из самых важных элементов в документе HTML – это элемент ADDRESS. В нем вы называете себя как автора документа. Элемент ADDRESS состоит из текста, расположенного между открывающим и закрывающим тегами <ADDRESS> </ADDRESS>.

### Форматирование текста

Для задания размера, цвета и начертания шрифта служит парный тег <FONT>, влияющий на весь текст, заключенный между открывающим и закрывающим тегами. Тег <FONT> должен иметь хотя бы один из трех возможных атрибутов:

1. SIZE – задает размер шрифта. Возможны семь заранее заданных размеров шрифта (от 1 до 7). Чем больше значение, тем крупнее шрифт. По умолчанию используется значение 3;
2. COLOR – задает цвет шрифта;
3. FACE – задает вид шрифта.

### Задание начертания шрифта

<B>текст жирный </B>

<I>текст наклонный </I>

<U> текст подчеркнутый </U>

<BIG> большой текст </BIG>

<SMALL> маленький текст </SMALL>

<STRONG> увеличенный текст </STRONG>

<STRIKE> перечеркнутый текст

<SUB> подстрочный текст </SUB>

<SUP> надстрочный текст </SUP>

### Списки

Язык HTML поддерживает 5 видов списков: нумерованный; маркированный; список определений; список каталогов; список меню.

Все списки оформляются однотипным образом. Начало и конец списка помечаются парным тегом, описывающим тип списка. Элементы списка – это блочные элементы HTML. Внутри каждого элемента списка можно использовать любые блочные или текстовые элементы HTML.

Маркированный список задается парным тегом <UL> и может включать атрибут TYPE, описывающий тип маркера. Напр.: TYPE="DISC" – круглый черный маркер; TYPE="SQUARE" – квадратный черный маркер; TYPE="CIRCLE" – маркер виде колечка.

Нумерованный список задается с помощью тега <OL> и может иметь значения атрибута TYPE: TYPE="1" – арабские цифры (1, 2, 3...); TYPE="a" – латинские буквы нижнего регистра (a, b, c...); TYPE="A" – латинские буквы верхнего регистра (A, B, C...); TYPE="i" – римские цифры с использованием нижнего регистра (i, ii, iii...); TYPE="I" – римские цифры с использованием верхнего регистра (I, II, III...).

Элементы списка, задаваемые внутри маркированного и нумерованного списков, описываются парным тегом <LI>.

Список определений описывается парным тегом <DL>. Предполагается, что данный список представляет собой «словарь», в котором чередуются термины и определения. Термины выделяются обычно полужирным шрифтом, а определения отображаются с отступом от левого поля.

### ***Рисунки на Web-страницах***

Иллюстрации играют важную роль в оформлении Web-страниц. Сами рисунки хранятся в отдельных файлах вне документа HTML, но отображаются браузером внутри Web-страницы.

Для размещения рисунков в документе служит одиночный тег <IMG>, содержащий обязательный атрибут SRC, значение которого составляет адрес URL файла изображения, записанный в абсолютной или относительной форме. При загрузке документа рисунок так же загружается и отображается в том месте документа, где расположен тег <IMG>. Для задания размера рисунка используются атрибуты WIDTH (ширина) и HEIGHT (высота).

Альтернативный текст – это более или менее подробное описание изображения. Если браузер не может показать рисунок, он в место него выводит альтернативный текст. Альтернативный текст задается в теге <IMG> значением специального атрибута ALT.

Атрибут ALIGN - используется для выравнивания объектов на экране. Доступные значения:

top – верхний край объекта выравнивается по верхнему краю строки;

middle – центр объекта выравнивается по базовой линии строки;

bottom – нижний край объекта выравнивается по базовой линии строки;

left – объект выравнивается по левому краю, при этом возможно обтекание объекта текстом;

right – объект выравнивается по правому краю, при этом возможно обтекание объекта текстом.

Если данный параметр не указан, большинство браузеров располагает изображение в левой части экрана, а текст – справа от него.

Изображение на страницах Web-страницы можно использовать в качестве фонового рисунка. При отображении документа, содержащего фоновый рисунок, на компьютере рабочая область окна заполняется этим рисунком последовательно.

Фоновый рисунок задается с помощью атрибута BACKGROUND в теге <BODY>.

### **Таблицы**

В HTML для разметки таблиц используются несколько тегов (одновременно). Это напоминает разметку списков, когда один тег определяет список целиком, а другой – отдельный элемент списка.

Тег <TABLE> размещает таблицу. Закрывающий тег обязателен. Любая таблица состоит из ячеек. В HTML ячейки таблицы сгруппированы по рядам (строкам). Для разметки ряда таблицы используется тег <TR>. Для разметки отдельных ячеек в строке используется <TD>. Спецификацией HTML допускается отсутствие закрывающих тегов </TD> и </TR>.

### **Основные атрибуты элементов таблицы**



Атрибут ALIGN – выравнивание содержимого ячеек (LEFT (влево), RIGHT (вправо) и CENTER (по центру)).

Атрибут BACKGROUND – задает фоновое изображение для ячеек строки.

Атрибут BGCOLOR – задает фоновый цвет для ячеек строки.

Атрибут BORDER – задает толщину рамки, обрамляющей ячейки строки. Нулевое значение данного атрибута означает отсутствие обрамления.

Атрибут VALIGN (встречается внутри тегов <TR>, <TH> и <TD>) – определяет вертикальное выравнивание данных в ячейках: TOP (вверху), BOTTOM (внизу), CENTER (по центру), MIDDLE (по середине) и BASELINE (по базовой линии).

Атрибут COLSPAN=n определяет, сколько табличных столбцов займет ячейка в пределах строки.

Атрибут ROWSPAN=m определяет, сколько табличных строк займет ячейка в пределах столбца.

### **Гипертекстовые ссылки**

Гипертекст – информационная структура, обеспечивающая навигацию посредством гипертекстовых ссылок. Гипертекстовыми ссылками называются фрагменты текста или изображения, при активизации которых отображаются связанные с ними документы.

Гипертекстовая ссылка задается тегами <A> и </A>. Она имеет несколько атрибутов, наличие одного из двух (HREF или NAME) обязательно. Первый указывает на ссылку за пределы данного документа, например:

<A HREF="http://www.имя\_сайта.зона/.../имя\_файла.html"> название ссылки </A>. Второй позволяет перейти к определенным образом отмеченному (с помощью так называемого якоря, или закладки) месту того же самого документа, например:

<A NAME="якоря"> название ссылки </A>

### **Фреймы**

Фреймы – позволяют разбивать Web-страницы на подокна, причем каждое подокно (фрейм) может иметь собственные полосы прокрутки и функционировать независимо от других подокон либо, наоборот, управлять их содержанием. Фреймы могут значительно улучшить внешний вид Web-страниц и повысить их функциональность.

### **Свойства фреймов**

Каждый фрейм имеет свой URL-адрес, что позволяет обращаться к нему и загружать его независимо от других фреймов.

Каждый фрейм имеет собственное имя (задаваемое атрибутом NAME), позволяющее обратиться к нему из другого фрейма.

Размер фрейма может быть изменен пользователем прямо на экране мышью (если это не запрещено с помощью специального параметра).

В отдельном фрейме, который будет все время оставаться на экране, можно разместить статическую информацию, которую автор считает необходимым постоянно показывать пользователю. Это может быть графический логотип фирмы, соглашение об авторских правах, набор управляющих кнопок, оглавление всех или части Web-документов, содержащихся на Web-сервере, что позволит пользователю быстро находить интересующую его информацию.

### **Наборы фреймов**

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тега <BODY> используется парный тег <FRAMESET>:

<HTML> <HEAD>...</HEAD> <FRAMESET>...</FRAMESET> </HTML>

То есть перед тем как размещать на Web-страницы фреймы, с помощью тега <FRAMESET> создают так называемый набор фреймов, содержащий информацию о числе фреймов, их относительных и абсолютных размерах.

*Общий синтаксис набора фреймов:*

<FRAMESET COLS="value" | ROWS="value"><FRAME SRC="url"> <FRAME ...>...</FRAMESET>

Общий контейнер <FRAMESET> описывает все фреймы, на которые делится экран. Можно разделить экран на несколько вертикальных или несколько горизонтальных фреймов. Для описания каждого фрейма в отдельности служит тег <FRAME>.

Между тегами <FRAMESET> и </FRAMESET> могут находиться теги <FRAME>, вложенные теги <FRAMESET> и </FRAMESET>, а также контейнер из тегов <NOFRAME> и </NOFRAME>, который позволяет продублировать документ для браузеров, не поддерживающих фреймы.

Тег <FRAMESET> допускает вложения и может иметь атрибуты COLS, ROWS, BORDER, FRAMEBORDER, причем атрибуты ROWS и COLS взаимоисключающие.

*Количество и размеры столбцов* в создаваемом наборе фреймов задает атрибут COLS. Значения ширины различных фреймов перечисляются в кавычках через запятую в пикселях, в процентах (%) или в связанных масштабных коэффициентах (\*), например: <FRAMESET COLS="число, число%, число\*">

*Количество и размеры строк* в создаваемом наборе фреймов в пикселях, в процентах (%) или в связанных масштабных коэффициентах (\*) задает атрибут ROWS, например:

<FRAMESET ROWS="число, число%; число\*">

Задание значений атрибута ROWS или COLS в пикселях – далеко не самый лучший способ описания высоты фрейма, поскольку пользователи могут произвольным образом менять размер рабочего поля браузера, не говоря уже о различных разрешениях экрана.

Если значения атрибута ROWS или COLS указываются в процентах (от 1% до 100%) и общая сумма размеров описываемых фреймов превышает 100%, то размеры всех фреймов пропорционально уменьшаются, чтобы в результате получилась сумма в 100 %. Если, соответственно, сумма меньше 100 %, то размеры пропорционально увеличиваются.

Задание одного из значений атрибута ROWS или COLS с единичным масштабным коэффициентом (в этом случае указывается просто звездочка без числа) означает, что все оставшееся место будет принадлежать данному фрейму. Если единичные масштабные коэффициенты указываются для двух или более фреймов, то оставшееся пространство делится поровну между этими фреймами. Если перед звездочкой стоит цифра, то она задает масштабный коэффициент для данного фрейма (во сколько раз он должен быть больше аналогичного фрейма, описанного звездочкой без числа).

Количество фреймов определяется количеством значений в списке. Общая сумма значений высоты всех фреймов (при любом способе задания значений атрибута) должна составлять высоту всего окна. Отсутствие атрибута ROWS означает один фрейм высотой с окно браузера.

*Толщину обрамления в пикселях* для всех фреймов, входящих в данный набор фреймов задает атрибут BORDER, например: <FRAMESET BORDER="число">

*Отменить вывод обрамления* у фреймов независимо от значения атрибута BORDER позволяет атрибут FRAMEBORDER. Значение 1 соответствует наличию, а 0 – отсутствию обрамления, например: <FRAMESET FRAMEBORDER="1"> <FRAMESET FRAMEBORDER="0">

### ***Вставка фрейма в документ с набором фреймов***

Как уже отмечалось, для вставки в документ отдельного фрейма служит непарный тег <FRAME> с атрибутами NAME, SRC, MARGINWIDTH, MARGINHEIGHT, SCROLLING, NORESIZE, FRAMEBORDER.

*Имя фрейма* задает атрибут NAME. Имя позволяет обратиться к этому фрейму в программе (это делается с помощью атрибута TARGET), например загрузить в него содержимое нового HTML-файла. Синтаксис: NAME="имя"

*Документ, который должен отображаться внутри данного фрейма*, задает атрибут SRC, например: <FRAME SRC="10.html"> Если атрибут SRC отсутствует, отображается пустой фрейм.

*Величину вертикальных разделительных полос* между фрейма позволяет указать атрибут MARGINWIDTH: MARGINWIDTH="value" Значение value указывается в пикселях и не может

быть меньше единицы. Значение, устанавливаемое по умолчанию, зависит от используемого браузера.

*Величину горизонтальных разделительных полос* между фреймами позволяет указать атрибут MARGINHEIGHT: MARGINHEIGHT="value". Значение value указывается в пикселях и не может быть меньше единицы.

Выводом *полос прокрутки фрейма* управляет атрибут SCROLLING:

SCROLLING="yes | no | auto"

Значение yes указывает, что фрейм будет иметь полосы прокрутки в любом случае, значение no – что полос прокрутки не будет. Значение auto, заданное по умолчанию, говорит о том, что полосы прокрутки появятся только при необходимости.

Атрибут NORESIZE позволяет создавать *фреймы без возможности изменения их размеров*. По умолчанию размер фрейма можно изменить мышью так же просто, как и размер окна Windows. Атрибут NORESIZE запрещает такую возможность. Если у одного из фреймов установлен атрибут NORESIZE, это лишает пользователя возможности перетаскивать границы не только данного фрейма, но и прилегающие к нему границы других фреймов.

Атрибут FRAMEBORDER задает обрамление у фреймов и может принимать значения 1 (есть обрамление) или 0 (нет обрамления).

### **Формы HTML-документов**

Пользовательская форма – это совокупность стандартных HTML-конструкций ввода текстовой и прочей информации и программы-обработчика этой информации, работающей на Web-сервере. Иными словами, пользовательская форма (или HTML-форма) служит для передачи информационных данных серверу.

Результат конструкций языка разметки HTML интерпретируется браузером, с помощью которого пользователь электронного документа получает информацию. Таким образом, объединив все эти формулировки, можно сказать, что HTML-форма выступает в роли посредника между пользователем и сервером.

Посетитель Web-страницы вводит в HTML-форму определенные данные, которые обрабатываются программой и отсылаются на Web-сервер. Все эти действия укладываются в три стадии:

1. Ввод пользователем информации.
2. Обработка введенной информации программой, установленной на сервере.
3. Получение результата отправления введенной информации на Web-сервер (открытие нового HTML-документа, переадресация на предыдущую страницу и пр.).

В качестве программы-обработчика чаще всего выступает CGI-сценарий (скрипт, который обычно разрабатывается на языке Perl или C/C++ и который взаимодействует со специальным компонентом Web-сервера -Common Gateway Interface) или программы, написанные на основе таких серверных языков программирования, как PHP, ASP, JSP и др.

Значение пользовательских форм трудно переоценить – они являются особым средством HTML, дающим посетителю возможность не только пассивно просматривать информацию, но и быть задействованным в содержании Web-сайта. Такое свойство принято называть интерактивностью, которая на сегодняшний день встречается практически во всех электронных документах.

Диапазон функций, выполняемых HTML-формами, чрезвычайно широк, поэтому остановимся лишь на некоторых, наиболее распространенных направлениях.

### **Организация обратной связи**

Форма обратной связи – это неотъемлемый атрибут Web-сайта любой тематики и функциональной направленности. Размещая электронный документ во Всемирной сети, мы уже подразумеваем реакцию пользовательской аудитории – похвалу и одобрение или же несогласие и критику.

В качестве средства обратной связи между пользователем и автором электронного документа могут выступать следующие.

1. *Почтовая форма.* Посетителя просят указать свои персональные данные и изложить суть обращения. Текст обращения обрабатывается программой и высылается автору сайта по электронной почте. При этом другие посетители сайта не могут получить доступ к содержанию отправленного сообщения;

2. *Гостевая книга.* Электронная книга жалоб и предложений, в которой любой желающий может оставить свой отзыв, поделиться впечатлениями о сайте, изложить критические замечания и указать на ошибки автора. В этом случае сообщения одного посетителя видимы для остальных пользователей гостевой книги;

3. *Интернет-форум.* Более мощный инструмент стимулирования оценки пользовательской аудитории, нежели почтовая форма или гостевая книга. В Интернет - форумах происходит многосторонний диалог, причем не только между человеком, задающим вопрос, и автором Web-сайта, но и между всеми пользователями данного Интернет-ресурса. На вопрос одного пользователя может ответить другой, третий подскажет то, чего не упомянул второй, наконец, автор сайта подытожит высказывания всех пользователей и даст свой развернутый ответ. Часто в рамках Интернет-форума работают дополнительные функциональные средства общения: личные сообщения, скрытые разделы форума для привилегированных пользователей, опросы общественного мнения и пр.;

4. *Чат.* Средство чата (от англ. "to chat" – разговаривать, болтать) подразумевает единовременное общение многих пользователей в режиме реального времени. В отличие от гостевой книги или форума, где ответ на поставленный вопрос может быть выслан и через час, и через день, общение в чате происходит между двумя (или более) собеседниками так, будто они сидят в одной комнате. Собеседники могут находиться в разных городах, государствах и временных зонах, однако, присутствуя в чате, все пользователи способны общаться одновременно.

### ***Структура пользовательских форм***

Структура любой пользовательской формы описывается тегом-контейнером <FORM>, внутри которого могут располагаться теги и параметры составляющих элементов HTML-формы. В пределах тега <FORM> не может содержаться другая пользовательская форма. Закрывающий тег обязателен для указания. Тег <FORM> может содержать четыре основных параметра: ACTION, METHOD, NAME и ENCTYPE. Рассмотрим каждый из них.

**<FORM> </FORM>**

Этот элемент необходим для построения сложных форм. Все формы начинаются тегом <FORM> и завершаются тегом </FORM>. После заполнения формы и подтверждения ввода со стороны пользователя введенная информация пересылается на сервер и обрабатывается при помощи CGI-программы, связанной с формой.

Параметр **ACTION** является единственным обязательным параметром тега <FORM> и предназначен для указания пути на Web-сервере к программе обработчику данных пользовательской формы, например:

<FORM ACTION = "http://www.название.домен/имя программы">

Параметр **METHOD** применяется для указания протокола, используемого для пересылки данных на сервер. Протокол GET выбран по умолчанию, но в большинстве случаев он не удовлетворяет разработчиков, поэтому чаще используется протокол POST.

При использовании типа GET данные пользовательской формы пересылаются в составе адреса запроса браузера: после имени программы-обработчика ставится знак вопроса (?), обозначающий вывод запроса браузера к переменным HTML-формы, а также последовательность переменных и их свойств из самой формы. Последовательность переменных формы разделяется символом амперсанда "&".

Параметр **NAME** присваивает HTML-форме уникальное имя, которое используется в программе-обработчике для идентификации пользовательских данных, например:

<FORM NAME="mail">

<INPUT>

Этот элемент позволяет создавать различные элементы управления в форме: текстовую строку; поле ввода пароля; поле выбора локального файла для загрузки на Web-сервер; опцию выбора (флажки); опцию переключения; кнопку отправления пользовательских данных; графический вариант кнопки отправления пользовательских данных; кнопку сброса введенных пользовательских данных.

Элемент не имеет конечного тега, так как все параметры задаются при помощи атрибутов.

Атрибут **type** позволяет указывать один из перечисленных элементов формы, каждому из которых соответствует конкретное значение.

**type = "text"** – это элемент текстовой строки.

Параметр	Описание
size	Определение максимального количества символов, вводимых в текстовой строке без перемещения курсора.
maxlength	Определение максимального количества символов, допустимых для ввода в текстовой строке. По умолчанию количество вводимых символов не ограничено.
name	Присвоение идентификационного имени для программы-обработчика.
value	Указание значения текстовой строки (при просмотре в браузере выводится в виде обычного текста в самой строке).

### Тема 5: Защита информации в сети

#### План:

1. Основные аспекты безопасности.
2. Методы защиты информации: криптография, электронная подпись, аутентификация, сертификация Web-узлов.

**Цель:** рассмотреть теоретические аспекты безопасности и методы защиты информации.

#### Задачи:

- определить сущность теоретических аспектов;
- охарактеризовать методы защиты информации;
- обозначить проблемы.

#### Ключевые вопросы:

##### Основные понятия в защите информации

Угроза безопасности информации - совокупность условий и факторов, создающих потенциальную или реально существующую опасность, связанную с утечкой информации, как несанкционированными, так и непреднамеренными воздействиями на нее.

Основными формами нарушения (угрозами), которые возможны при наличии уязвимостей, применительно к основным свойствам информации, являются:

- разглашение (конфиденциальной) информации – несанкционированное доведение защищаемой информации до потребителей, не имеющих права доступа к этой информации;
- несанкционированный доступ (НСД) – получение защищаемой информации заинтересованным субъектом с нарушением установленных правовыми документами или собственником, владельцем информации прав или правил доступа к защищаемой информации;
- изменение, искажение (модификация), копирование, уничтожение и др. в результате непреднамеренных или несанкционированных воздействий;
- блокирование доступа к ЗИ в результате непреднамеренных или несанкционированных воздействий;
- уничтожение, или сбой функционирования носителя ЗИ в результате непреднамеренных или несанкционированных воздействий;

##### Сбои оборудования:

- сбои кабельной системы перебои электропитания; сбои дисковых систем; сбои систем архивации данных; сбои работы серверов, рабочих станций, сетевых карт и т.д.
- потери информации из-за некорректной работы ПО;

- потери или изменение данных при ошибках ПО; потери при заражении системы компьютерными вирусами;
- технические средства получения информации:
- перехват электрического излучения;
- включение в электрическую цепь;
- жучки;
- акустический перехват;
- хищение информационных носителей;
- программные ловушки;
- внедрение вирусов.

Программные средства - программы, специально предназначенные для выполнения функций, связанных с защитой информации.

#### Абстрактные модели защиты информации

Одной из первых моделей была опубликованная в 1977 модель Биба (Biba). Согласно ей все субъекты и объекты предварительно разделяются по нескольким уровням доступа, а затем на их взаимодействия накладываются следующие ограничения: 1) субъект не может вызывать на исполнение субъекты с более низким уровнем доступа; 2) субъект не может модифицировать объекты с более высоким уровнем доступа. Как видим, эта модель очень напоминает ограничения, введенные в защищенном режиме микропроцессоров Intel 80386+ относительно уровней привилегий.

Модель Гогена-Мезигера (Goguen-Meseguer), представленная ими в 1982 году, основана на теории автоматов. Согласно ей система может при каждом действии переходить из одного разрешенного состояния только в несколько других. Субъекты и объекты в данной модели защиты разбиваются на группы – домены, и переход системы из одного состояния в другое выполняется только в соответствии с так называемой таблицей разрешений, в которой указано какие операции может выполнять субъект, скажем, из домена С над объектом из домена D. В данной модели при переходе системы из одного разрешенного состояния в другое используются транзакции, что обеспечивает общую целостность системы.

Сазерлендская (от англ. Sutherland) модель защиты, опубликованная в 1986 году, делает акцент на взаимодействии субъектов и потоков информации. Так же как и в предыдущей модели, здесь используется машина состояний со множеством разрешенных комбинаций состояний и некоторым набором начальных позиций. В данной модели исследуется поведение множественных композиций функций перехода из одного состояния в другое.

Важную роль в теории защиты информации играет модель защиты Кларка-Вильсона (Clark-Wilson), опубликованная в 1987 году и модифицированная в 1989. Основана данная модель на повсеместном использовании транзакций и тщательном оформлении прав доступа субъектов к объектам. Но в данной модели впервые исследована защищенность третьей стороны в данной проблеме – стороны, поддерживающей всю систему безопасности. Эту роль в информационных системах обычно играет программа-супервизор. Кроме того, в модели Кларка-Вильсона транзакции впервые были построены по методу верификации, то есть идентификация субъекта производилась не только перед выполнением команды от него, но и повторно после выполнения. Это позволило снять проблему подмены автора в момент между его идентификацией и собственно командой. Модель Кларка-Вильсона считается одной из самых совершенных в отношении поддержания целостности информационных систем.

#### Особенности применения криптографических методов

Проблемой защиты информации путем ее преобразования занимается *криптология* (*kryptos* - тайный, *logos* - наука). Криптология разделяется на два направления - *криптографию* и *криптоанализ*. Цели этих направлений прямо противоположны.

*Криптография* занимается поиском и исследованием математических методов преобразования информации.

Сфера интересов *криптоанализа* - исследование возможности расшифровывания информации без знания ключей.

В этой книге основное внимание будет уделено криптографическим методам.

Современная криптография включает в себя четыре крупных раздела:

- 1) Симметричные криптосистемы.
- 2) Криптосистемы с открытым ключом.
- 3) Системы электронной подписи.
- 4) Управление ключами.

Основные направления использования криптографических методов - передача конфиденциальной информации по каналам связи (например, электронная почта), установление подлинности передаваемых сообщений, хранение информации (документов, баз данных) на носителях в зашифрованном виде.

В качестве информации, подлежащей шифрованию и дешифрованию, будут рассматриваться *тексты*, построенные на некотором *алфавите*. Под этими терминами понимается следующее.

*Алфавит* - конечное множество используемых для кодирования информации знаков.

*Текст* - упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС можно привести следующие:

- алфавит  $Z_{33}$  - 32 буквы русского алфавита и пробел;
- алфавит  $Z_{256}$  - символы, входящие в стандартные коды ASCII и КОИ8;
- бинарный алфавит -  $Z_2 = \{0,1\}$ ;
- восьмеричный алфавит или шестнадцатеричный алфавит.

*Шифрование* - преобразовательный процесс: *исходный текст*, который носит также название *открытого текста*, заменяется *шифрованным текстом*.

*Дешифрование* - обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный.

*Ключ* - информация, необходимая для беспрепятственного шифрования и дешифрования текстов.

*Криптографическая система* представляет собой семейство  $T$  преобразований открытого текста. Члены этого семейства индексируются, или обозначаются символом  $k$ ; параметр  $k$  является *ключом*. Пространство ключей  $K$  - это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита.

## **Тема 6:** Решение задач описательной статистики средствами Excel

### **План:**

1. Применение стандартных функций MS Excel для решения задач описательной статистики.
2. Ввод исходных данных.
3. Вычисления размаха (вариации), оценки среднего, среднеквадратичного отклонения и дисперсии, асимметрии и эксцесса.
4. Построение таблицы частот и гистограммы.
5. Применение пакета анализа для решения задач описательной статистики.
6. Сглаживание данных.
7. Описательная статистика, построение гистограмм

**Цель:** изучить принципы обработки статистических данных и способов представления полученных результатов в статистических таблицах и на графиках.

### **Задачи:**

1. Сформулировать методы заполнения и формирования таблиц на основе использования встроженных формул в Microsoft Excel;
2. Описать основные приемы вычисления показателей вариации и группировки данных;
3. Сформулировать алгоритм построения графиков.

### **Ключевые слова:**

Прогнозирование социальных процессов

Прогнозирование (от греческого – предвидение, предсказание) – способ предвидения, используя ту или иную степень вероятности результата, направленности протекания социальных процессов в течение некоторого промежутка времени.

Разработка прогноза – научное исследование, которое представляет собой развитие какого-либо социального явления.

Цель прогнозирования – эффективное воздействие на явление в нужном направлении, а не просто предвидение будущего явления.

Виды социального прогнозирования: социально-экономические прогнозы, социологическое прогнозирование, прогнозирование социально-психологических процессов, юридическое прогнозирование.

Прогнозы можно классифицировать, как средне-, кратко-, долгосрочные, качественные и количественные.

Качественные – проводятся на базе субъективных оценок экспертов. Количественные – определяются с помощью статистических данных и математических методов расчета.

#### Метод скользящей средней

Трендовые значения в методе скользящей средней определяются с помощью прямых расчетов, а прогнозное значение рассчитывается как среднее арифметическое.

Экстраполирование тренда (метод аналитического выравнивания)

Анализом данных в целях прогнозирования тренда называется экстраполяция. Долгосрочное направление развития временного ряда есть тренд. Временной ряд – последовательность результатов наблюдений признака упорядоченных по времени до текущего момента. Полученные данные в будущем экстраполируются.

равномерное развитие (рис.1) – развитие с постоянным абсолютным приростом уровней временного ряда. Основная тенденция развития описывается линейным типом тренда  $y = a_0 + a_1t$ , где  $a_0 = \text{const}$ ,  $a_1$  - коэффициент, характеризующий скорость (темп) развития изучаемого процесса и направление его развития (при  $a_1 > 0$  уровни динамики равномерно возрастают, при  $a_1 < 0$  – равномерно снижаются).

Основные типы развития явлений во времени временных рядов:

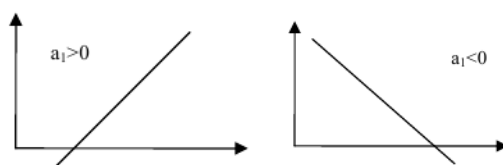


Рис.1. Равномерное развитие временного ряда

равноускоренное (равнозамедленное) развитие (рис.2) – развитие при постоянном увеличении (замедлении) темпа прироста уровней временного ряда. Основная тенденция развития описывается полиномом второй степени  $y = a_0 + a_1t + a_2t^2$ , где  $a_2$  – коэффициент, характеризующий постоянное изменение скорости (темпа) развития (при  $a_2 > 0$  происходит ускорение развития, при  $a_2 < 0$  – замедление развития).

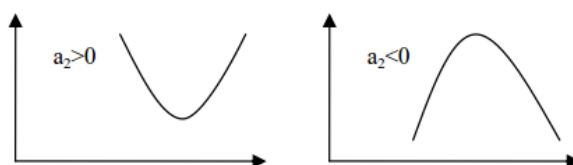


Рис.2. Равноускоренное развитие временного ряда

развитие с переменным ускорением (замедлением) (рис.3) – развитие при переменном увеличении (замедлении) темпа прироста уровней временного ряда. Основная тенденция описывается полиномом третьей степени:  $y = a_0 + a_1t + a_2t^2 + a_3t^3$ , где  $a_3$  – коэффициент, характеризующий изменение ускорения развития (при  $a_3 > 0$ , ускорение возрастает; при  $a_3 < 0$  – замедляется).



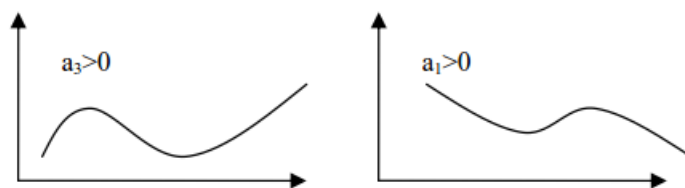


Рис.3. Развитие с переменным ускорением временного ряда

развитие с замедлением роста в конце периода (рис. 4) – развитие, при котором прирост в конечных уровнях временного ряда стремится к нулю. Основная тенденция описывается логарифмической функцией  $y = a_0 + a_1 \ln t$

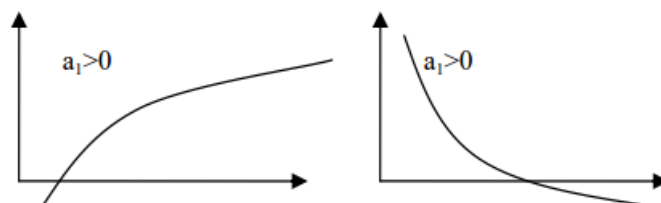


Рис.4. Развитие с замедлением роста в конце периода временного ряда

развитие по экспоненте – развитие, характеризующееся стабильным темпом роста (снижения). Основная тенденция описывается показательной функцией (в частном случае, экспоненциальной функцией  $y = a_0 a_1^t$ , где  $a_1$  – коэффициент, характеризующий интенсивность развития).

развитие по степенной функции – развитие с постоянным относительным приростом уровней временного ряда. Основная тенденция развития описывается степенной функцией  $y = a_0 t_1^a$ .

Использовать трендовые модели для среднесрочных и краткосрочных прогнозов можно, в том случае если:

в анализируемый период трендовая модель развивается эволюционно;

описываемый временным рядом процесс, обладает значительным количеством времени, для описания наступления изменения в поведении данного процесса;

на прогнозируемой оценке сильнее должно отражаться влияние более поздней информации, чем более ранней;

изучаемый прогнозируемый процесс, должен иметь достаточный период времени, для выявления закономерностей.

При выборе конкретной функции необходимо учесть, что функция должна максимально отражать временной ряд. После определения коэффициентов функции тренда, подставляя в функцию будущие значения  $t$ , можно рассчитать прогнозируемое значение  $y$ .

## 2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КОНСПЕКТИРОВАНИЮ ЛЕКЦИОННОГО МАТЕРИАЛА

Конспектирование – процесс мысленной переработки и письменной фиксации информации, в виде краткого изложения основного содержания, смысла какого-либо текста. Результат конспектирования – запись, позволяющая конспектирующему немедленно или через некоторый срок с нужной полнотой восстановить полученную информацию. Конспект в переводе с латыни означает «обзор». По существу его и составлять надо как обзор, содержащий основные мысли текста без подробностей и второстепенных деталей. Конспект носит индивидуализированный характер: он рассчитан на самого автора и поэтому может оказаться малопонятным для других.

Как конспектировать текст. Выделение главной мысли – одна из основ умственной культуры при работе с текстом. Во всяком научном тексте содержится информация двух видов: основная и вспомогательная. Основной является информация, имеющая наиболее существенное

значение для раскрытия содержания темы или вопроса. К ней относятся: определения научных понятий, формулировки законов, теоретических принципов и т.д. Назначение вспомогательной информации – помочь читателю лучше усвоить предлагаемый материал. К этому типу информации относятся разного рода комментарии. Как же следует поступать с информацией каждого из этих видов в процессе конспектирования? Основную – записывать как можно полнее, вспомогательную, как правило, опускать. Содержание конспектирования составляет переработка основной информации в целях ее обобщения и сокращения. Обобщить – значит представить ее в более общей, схематической форме, в виде тезисов, выводов, отдельных заголовков, изложения основных результатов и т.п. Выбор ключевых слов – это первый этап смыслового свертывания, смыслового сжатия материала. Важными требованиями к конспекту являются наглядность и обзримость записей и такое их расположение, которое давало бы возможность уяснить логические связи и иерархию понятий. В процессе чтения следует делать лишь предварительные заметки (тезисы), отмечая вкладышами наиболее важные положения, факты, и только по прочтению всей книги можно приступить к составлению ее конспекта. Наряду с текстом, цитируемым дословно, конспект содержит также соображения и мысли его составителя. Можно включить сюда факты, цифры, таблицы и схемы из конспектируемой книги. В конспекте желательно выделить подчеркиванием или условными значками наиболее характерные места текста, выводы и определения, следует также оставлять поля для дополнительных записей и заметок. Составление конспекта требует вдумчивой работы, затраты времени и усилий. Наконец, конспект включает и выписки. В него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из конспектируемой книги. Работа над конспектом только тогда полноценная и творческая, когда она не ограничена рамками текста изучаемого произведения.

Приступая к составлению конспекта, прежде всего, следует указать фамилию автора произведения, полное название работы, год и место издания. Если конспектируется статья, то указывается, где и когда она была напечатана. Полезно также отметить страницы изучаемого материала, чтобы можно было, руководствуясь записями, быстро отыскать в книге нужное место. Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты этого плана целесообразно записывать в тексте или на полях конспекта. При конспектировании (так же, как и при остальных видах записей) допускаются сокращения слов, но нужно соблюдать известную осторожность и меру. Случайные сокращения ведут к тому, что спустя некоторое время конспект становится непонятным. Недопустимы сокращения в наименованиях и фамилиях. Конспект ведется в тетради или на отдельных листках. Записи в тетради легче оформить, они занимают меньше места, их удобно брать с собой на лекции. Рекомендуется оставлять в тетрадях поля для последующей работы над конспектом, для дополнительных записей, замечаний и пунктов плана. Конспект в тетради имеет, однако, и недостаток: в нем мало места для пополнения новыми материалами, выводами и обобщениями. В этом отношении более удобен конспект на отдельных листках. Из него нетрудно извлечь отдельную понадобившуюся запись, его можно быстро пополнить листками с новыми выводами, обобщениями, фактическими данными. При подготовке выступлений, лекций и докладов легко подобрать листки из различных конспектов, свести их вместе.

#### **Памятка обучающемуся по конспектированию текста**

1. Внимательно прочитать текст. Уточнить в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.
2. Выделить главное, составьте план.
3. Кратко сформулировать основные положения текста, отметить аргументацию автора.
4. Законспектировать материал, четко следуя пунктам плана. При конспектировании стараться выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывать цитаты. Цитируя, учитывать лаконичность, значимость мысли. Конспект должен быть легко обзрим и легко читаем. Для этого надо выполнить правила оформления:  
заголовок пишется цветной пастой;

левая треть листа отводится под поле для отметок обучающегося, 2/3 справа предназначены для конспектирования;

подзаголовки пишутся темной пастой и подчеркиваются цветной;

в тексте конспекта высота строчных букв 2 мм (бумага в клетку, записи в каждой строке);

абзацы текста отделяются друг от друга пробельной строкой, чтобы облегчить чтение записей;

в каждом абзаце ключевое слово подчеркивается цветной пастой;

в конце изучаемой темы оставляется чистая страница для построения структурно-логической схемы или сжатой информации иного типа.

### **3 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ**

Важной составной частью учебного процесса в университете являются лабораторные занятия.

Задачей преподавателя при проведении лабораторных работ является грамотное и доступное разъяснение принципов и правил проведения работ, побуждение обучающихся к самостоятельной работе, определения места изучаемой дисциплины в дальнейшей профессиональной работе будущего выпускника.

Цель лабораторной работы – научить обучающихся самостоятельно производить необходимые действия для достижения желаемого результата.

Прежде чем приступить к выполнению лабораторной работы, обучающемуся необходимо ознакомиться с теоретическим материалом, соответствующим данной теме.

Выполнение лабораторной работы целесообразно разделить на несколько этапов:

- формулировка и обоснование цели работы;
- определение теоретического аппарата, применительно к данной теме;
- выполнение заданий;
- анализ результата;
- выводы.

Индивидуальные задания для лабораторных работ представлены конкретно-практическими и творческими задачами.

На первой ступени изучения темы выполняются конкретно-практические задачи, при решении которых формируется минимальный набор умений. Преподаватель опосредованно руководит познавательной деятельностью обучающихся, консультирует и подробно разбирает со обучающимися возникшие затруднения в ходе решения задачи, обращает внимание группы на возможные ошибки.

Вторая ступень изучения темы дифференцируется в зависимости от степени усвоения его обязательного уровня. Обучающиеся, усвоив содержание типовых методов и приемов решения задач, приступают к решению творческих задач. Если уровень знаний и умений, демонстрируемых обучающимся при контрольном обследовании, не соответствует установленным требованиям, обучающийся вновь возвращается к стандартным упражнениям, но под более пристальным наблюдением преподавателя.

После изучения отдельной темы курса дисциплины, каждый обучающийся получает определенное количество баллов по результатам выполнения лабораторных работ. Максимальное количество баллов за каждую лабораторную работу указано в рабочей программе по дисциплине «Информатика».

Выполнение лабораторных работ актуально и значимо для текущей и промежуточной аттестации.

#### **Тема 1. Управление списками в Microsoft Excel**

**Цель:** изучить приёмы консолидации данных, создания сводных таблиц и подведения итогов при помощи табличного процессора Microsoft Excel.

**Задачи:**

познакомить со способами подведения промежуточных итогов в базах данных Microsoft Excel;

2) уметь применять средства Microsoft Excel для сортировки и фильтрации информации в базе данных;

3) научить составлять итоговые таблицы, связывая данные из других источников.

### **Основные вопросы:**

Какие возможности предоставляет программа Microsoft Excel для работы с базами данных?

2. Что такое сортировка? Как отсортировать список по двум и более ключам?

3. Что такое фильтр?

4. В чем отличие сортировки списка от фильтрации списка?

5. Что собой представляет расширенный фильтр?

6. Как сформировать критерий для расширенного фильтра?

7. Для чего используются сводные таблицы?

8. Что может использоваться в качестве диапазона данных для построения сводной таблицы?

9. Для чего используется поле Страница в макете сводной таблицы?

10. Для чего используется поле Строка в макете сводной таблицы?

11. Для чего используется поле Данные в макете сводной таблицы?

12. Как изменить функцию для расчета данных в сводной таблице?

13. Какие функции используются в Excel для расчета данных в сводной таблице?

14. Как можно провести группировку данных в сводной таблице?

15. Консолидация данных по расположению.

16. Консолидация данных по категории.

## **Тема 2: База данных Microsoft Access**

**Цели:** сформировать у студентов основные понятия реляционных БД и показать возможности средств автоматизации проектирования БД.

### **Задачи:**

- знать основные положения теории баз данных, технологии организации БД, содержание этапов проектирования БД и особенности реляционной модели;
- владеть методами и средствами представления данных и знаний о предметной области.

### **Основные вопросы:**

1. Что называется базой данных (БД)? Как представляются БД?

2. Что такое система управления базой данных (СУБД)?

3. Что представляет собой СУБД Access? В чем его отличие от табличного процессора Excel?

4. Какие БД называются реляционными?

5. Каковы основные элементы БД?

7. Что определяют поля таблицы?

8. Каковы основные свойства полей?

9. Что такое первичный ключ?

10. В каких случаях используется составной ключ?

11. Какие этапы включает в себя технология разработки СУБД?

12. Для чего предназначены таблицы БД?

13. Каково назначение схемы данных?

14. Перечислите типы связей между таблицами. Охарактеризуйте их.

15. Что такое модель данных?

16. Иерархическая модель данных. Привести пример.

17. Сетевая модель данных. Привести пример.

18. Реляционная модель данных. Привести пример.

19. Объекты БД.

20. Какие этапы включает в себя технология разработки СУБД?

21. Каково назначение полей "Тип данных" и "Описание"?
22. Как установить связь между таблицами?
23. Между какими полями таблиц устанавливается связь?
24. В каких режимах можно работать в Access.
25. Основные разделы отчета.
26. Виды отчетов.
27. Виды запросов в Access.

### **Тема 3: Поиск информации в Интернет.**

**Цель:** научить пользоваться информационно-поисковыми системами, разыскивать необходимую информацию в сети Internet и работать с электронной почтой.

**Задачи:**

- владеть навыками работы в локальных и глобальных компьютерных сетях, использовать в профессиональной деятельности сетевые средства поиска и обмена информацией;
- изучить основные навыки простого и расширенного поиска информации;
- научиться работать с поисковыми системами.

**Основные вопросы**

1. Назвать средства поиска информации в Интернет.
2. Перечислить поисковые системы. Как их классифицировать?
3. Назвать адреса наиболее популярных русскоязычных поисковых систем.
4. Каковы правила составления запроса при поиске информации в информационно-поисковой системе aport.ru?
5. Какие типы графических файлов используются в Интернет?
6. Как выполнить перевод текста с одного языка на другой?
7. Какие языки поддерживает электронный переводчик www.translate.ru?

### **Тема 4: Создание Web-страницы с помощью языка HTML**

**Цель:** овладеть основными приемами создания WEB - страниц на языке HTML.

**Задачи:**

- познакомиться с основными возможностями языка HTML для создания Web-страниц;
- выявить особенности использования языка гипертекстовой разметки HTML.

**Основные вопросы:**

1. Сохранение Web-страниц. Особенности сохранения страниц, содержащих фреймы.
2. HTML. Интерпретация тегов HTML. Основные структурные элементы HTML.
3. Заголовки Что представляет собой язык HTML?
4. Что значит тэг? Какие бывают тэги? Порядок записи элементов тэгов.
5. Какие обязательные элементы входят в структуру HTML-документа?
6. Как определить цвет фона и цвет текста Web-страницы?
7. Как определить уровень заголовков Web-страницы средствами языка HTML?
8. Как увеличить (уменьшить) размер текста, оформить текст полужирным, курсивным, подчеркнутым начертанием
9. Как вставить рисунок в Web-документ в GIF-формате?
10. Как создать маркированный (нумерованный) список на Web-странице?
11. Что такое гипертекст, гиперссылка? Какие тэги используются для задания перехода по гиперссылке в языке HTML?
12. Какие атрибуты используются для изменения цвета гиперссылок?
13. Как создать таблицу на Web-странице?
14. Какие атрибуты используются для горизонтального и вертикального выравнивания ячеек в таблице, для определения высоты и ширины ячеек, их цвета?
15. Что такое фрейм? Какие тэги используются для описания фреймов.

### **Тема 5:** Решение задач описательной статистики средствами Excel

**Цель:** изучение принципов обработки статистических данных средствами Microsoft Excel и способов представления результатов в статистических таблицах и на графиках.

**Задачи:**

- закрепление навыков формирования и заполнения таблиц с использованием встроенных формул;
- овладение методикой группировки данных и вычисления основных показателей вариации;
- приобретение навыков построения графиков.

**Основные вопросы:**

1. Что называется прогнозированием.
2. Раскрыть принцип скользящего среднего.
3. Различия между методом скользящей средней и экстраполированием тренда.
4. Перечислить основные типы развития явлений во времени.
5. Равномерное развитие.
6. Равноускоренное развитие.
7. Развитие с переменным ускорением.
8. Развитие с замедлением роста в конце периода.
9. Развитие по экспоненте.
10. Развитие по степенной функции.
11. Условия создания трендовой модели для краткосрочных и среднесрочных прогнозов.
12. Показатели вариации.
13. Применение стандартных функций Microsoft Excel для решения задач описательной статистики.
14. Что такое тренд? Какими моделями он может быть описан?
15. Какими способами можно выделить тренд?

Содержание заданий отражено в учебно-методических пособиях:

Информационные технологии: учеб.-метод. пособие/ Н.А. Чалкина, Е.А. Абросимова, О.А. Лебедь, А.М. Попова; АмГУ, ФМиМ. – Благовещенск: Изд-во Амур. гос. ун-та, 2016. – 124 с.  
[http://irbis.amursu.ru/DigitalLibrary/AmurSU\\_Edition/7434.pdf](http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/7434.pdf)

### **4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ**

Практическая работа - это такая познавательная учебная деятельность, когда последовательность мышления студента, его умственные и практические операции и действия зависят и определяются самим студентом. Практическая работа - это метод, который очень помогает выяснить способности студентов. Работая практически, студент должен постепенно овладеть такими общими приёмами практической работы как ясное представление цели работы её выполнение, проверка, исправление ошибок. Выполнение практических работ студентами влияет на формирование и развитие информационных компетенций. Студенты овладевают способами работы с информацией:

- поиск в каталогах, поисковых системах, иерархических структурах;
- извлечение информации с различных носителей;
- систематизация, анализ и отбор информации (разные виды сортировки, фильтры, запросы, структурирование файловой системы, проектирование баз данных и т.д.);
- технические навыки сохранения, удаления, копирования информации и т.п. –
- преобразование информации (из графической – в текстовую, из аналоговой – в цифровую и т.п.)

Основными задачами практических работ являются: формирование умений подбирать материалы по их назначению, условиям эксплуатации, применять их при выполнении работ.

Содержание практической работы составляют:

- номер и тема практической работы;

- цель практической работы;
- рекомендации для выполнения практической работы;
- перечень используемых материалов, инструментов, оборудования;
- порядок выполнения работы;
- вывод о проделанной работе.

Перед тем как приступить к выполнению практической работы, студент должен усвоить краткие теоретические сведения по теме, методику выполнения работы, а также способы представления полученных данных.

### **Тема 1: Электронные таблицы Microsoft Excel**

**Цель:** изучение возможностей и практическое применение информационных технологий по созданию электронных таблиц, выполнению базовых математических операций, автоматический ввод данных и работы с таблицами.

**Задачи:**

- ознакомление с использованием формул и функций, выполнение вычислений;
- выработка практических навыков у студентов по использованию технологий конструирования электронных таблиц, оформления результатов работы, анализа данных в Excel.

**Основные вопросы:**

1. Структура рабочей книги. Структура окна Excel, окна рабочей книги.
2. Определения понятий: «рабочая книга», «рабочий лист», «диапазон ячеек», «ячейка».
3. Перечислить основные типы функций, используемых в Excel.
4. Составные элементы формул. Правила записи формул.
5. Категории функций MS Excel.
6. Вызов мастера функции. Вставка функции в формулу с использованием мастера функций.
7. Абсолютная, относительная и смешанная адресация.
8. Создание диаграмм (встроенных и на отдельных листах).
9. Типы диаграмм Excel. Основные компоненты диаграмм. Редактирование отдельных элементов.
10. Особенности применения функций работы с текстами и датами.

### **Тема 2: База данных Microsoft Access**

**Цель:** сформировать у студентов основные понятия реляционных БД и показать возможности средств автоматизации проектирования БД.

**Задачи:**

- знать основные положения теории баз данных, технологии организации БД, содержание этапов проектирования БД и особенности реляционной модели;
- владеть методами и средствами представления данных и знаний о предметной области.

**Основные вопросы:**

1. В каких режимах можно работать в Access.
2. Основные разделы отчета. Виды отчетов.
3. Виды запросов в Access.
4. Каково назначение полей "Тип данных" и "Описание"?
5. Как установить связь между таблицами?
6. Между какими полями таблиц устанавливается связь?
7. Каково назначение схемы данных?
8. Перечислите типы связей между таблицами. Охарактеризовать их.
9. Что такое первичный ключ?
10. В каких случаях используется составной ключ?

Содержание заданий отражено в учебно-методических пособиях:

Информационные технологии: учеб.-метод. пособие/ Н.А. Чалкина, Е.А. Абросимова, О.А. Лебедь, А.М. Попова; АмГУ, ФМиМ. – Благовещенск: Изд-во Амур. гос. ун-та, 2016. – 124 с.  
[http://irbis.amursu.ru/DigitalLibrary/AmurSU\\_Edition/7434.pdf](http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/7434.pdf)

## **5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

В процессе выполнения самостоятельной работы обучающиеся получают:

*практические умения и навыки :*

умение оперировать данными на информационном рынке;  
умения работать с информацией (кодировать, представлять, измерять);  
умения обрабатывать информацию средствами информатики.

*учебные умения:*

использовать различные информационные источники;  
расспрашивать, описывать, сравнивать, исследовать, анализировать оценивать;  
проводить самостоятельный поиск необходимой информации;

*специальные учебные умения:*

осуществлять эффективный и быстрый поиск нужной информации;  
организовывать работу на компьютере;  
выбирать оптимальное программное обеспечение для работы с информацией;  
излагать информацию средствами информатики.

Самостоятельные работы выполняются индивидуально на домашнем компьютере или в компьютерном классе в свободное от занятий время.

Обучающийся обязан:

перед выполнением самостоятельной работы, повторить теоретический материал, пройденный на аудиторных занятиях;  
выполнить работу согласно заданию;  
по каждой самостоятельной работе представить преподавателю отчет в виде результирующего файла на внешнем носителе;  
ответить на поставленные вопросы.

При выполнении самостоятельных работ обучающийся должен сам принять решение об оптимальном использовании возможностей программного обеспечения. Если по ходу выполнения самостоятельной работы у обучающихся возникают вопросы и затруднения, он может консультироваться у преподавателя.



## Оглавление

<b>ВВЕДЕНИЕ</b> .....	<b>3</b>
<b>1 КРАТКОЕ ИЗЛОЖЕНИЕ ЛЕКЦИОННОГО МАТЕРИАЛА</b> .....	<b>4</b>
<b>2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КОНСПЕКТИРОВАНИЮ ЛЕКЦИОННОГО МАТЕРИАЛА</b> .....	<b>25</b>
<b>3 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ</b> .....	<b>27</b>
<b>4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ</b> .....	<b>30</b>
<b>5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ</b> .....	<b>32</b>

**Ангелина Михайловна Попова,**

*старший преподаватель, каф. общей математики и информатики АмГУ,*

**Ольга Анатольевна Лебедь,**

*старший преподаватель, каф. общей математики и информатики АмГУ*