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

Кафедра программной инженерии

Методические указания
к лабораторным работам по дисциплине
«Информатика и основы программирования»

для студентов направления подготовки 45.03.03
«Фундаментальная и прикладная лингвистика»

Курск 2017

УДК 681.3

Составитель Е.И.Аникина

Рецензент
Кандидат технических наук, доцент кафедры программной инженерии *Н.Н. Бочанова*

Методические указания к лабораторным работам по дисциплине «Информатика и основы программирования» для студентов направления подготовки 45.03.03 «Фундаментальная и прикладная лингвистика»/Юго-Зап. гос. ун-т; сост. Е.И. Аникина. Курск, 2017. 74 с.

Содержит задания к лабораторным работам, теоретические сведения и примеры решения задач по темам курса, связанным с технологией работы с текстовыми процессорами, базами данных и разработкой приложений для компьютерной обработки текстов.

Предназначено для студентов направления подготовки бакалавров 45.03.03 «Фундаментальная и прикладная лингвистика».

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.
Усл. печ. л. . Уч.-изд. л. . Тираж 100 экз. Заказ .
Бесплатно.

Юго-Западный государственный университет
305040, Курск, ул.50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ТЕХНОЛОГИЯ РАБОТЫ С ТЕКСТОВОЙ ИНФОРМАЦИЕЙ

Лабораторная работа №1

СОЗДАНИЕ ДОКУМЕНТА средствами LibreOffice Writer

Цель работы: ознакомить с основными элементами рабочего окна Writer, продемонстрировать возможности начальной настройки текстового процессора, научить создавать текстовый документ в среде Writer и выпол-нять установку его начальных параметров.

Задание и технология работы

1. Выполните начальную настройку текстового процессора Writer.

1.1. Загрузите текстовый процессор Writer, используя **Пуск / Програм-мы / LibreOffice**.

1.2. Загрузите справочную систему (пункт меню **Справка**). Ознакомь-тесь со структурой справочной системы. Ее разделы вы можете использовать в случае затруднений при выполнении лабораторных работ.

1.3. Ознакомьтесь с элементами окна Writer (рис. 1) и выполните на-стройки, удобные для набора текста.

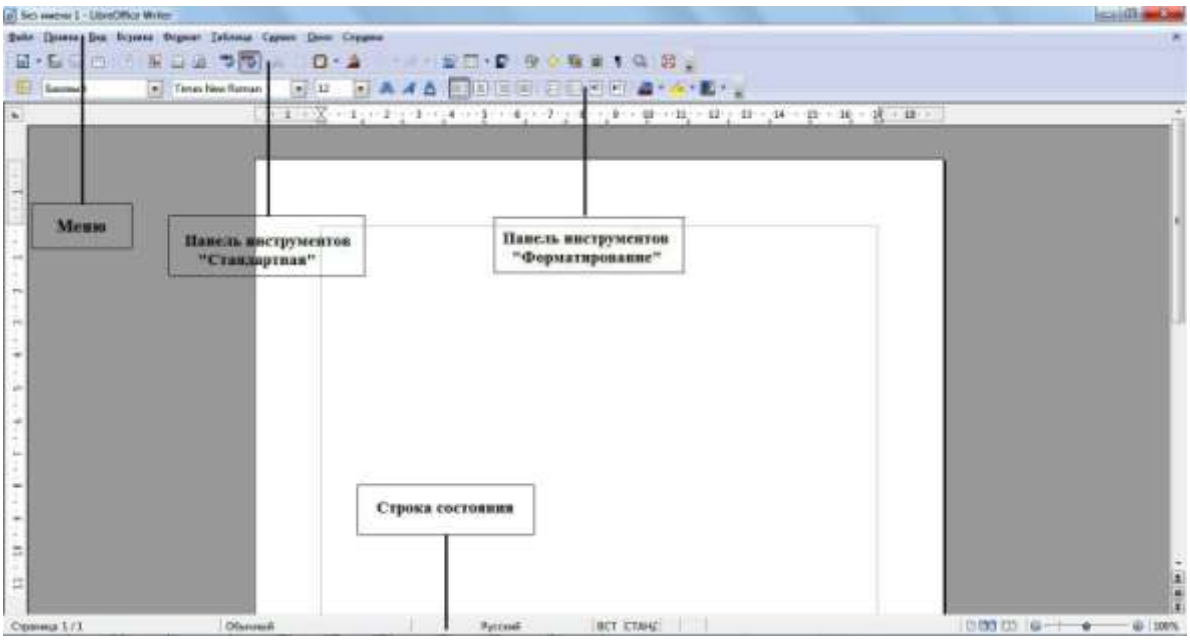


Рис. 1. Рабочее окно LibreOffice Writer

2.

С помощью пункта меню **Вид** установите режим отображения **Разметка печати**, убедитесь в наличии горизонтальной и вертикальной линеек (при необходимости подключите их, используя команду **Вид / Линейка**). Опробуйте другие режимы отображения.

Проверьте наличие двух панелей – **Стандартной** и **Форматирования** (**Вид / Панели инструментов**). Если установлены другие панели, отключите их (**Вид / Панели инструментов**). Установите масштаб изображения 100 % (**Вид / Масштаб / 100 % / ОК** или поле **Масштаб** с раскрывающимся списком в стандартной панели инструментов).

1.4. Установите параметры работы текстового процессора. Для этого выполните команду **Сервис / Параметры** и в дереве каталога, расположенном с левой стороны, войдите в раздел **LibreOffice Writer**:

– в разделе **Общие** установите единицы измерения – сантиметры;

– в разделе **Вид** проверьте наличие флажков у настроек **Изображения и внедренные объекты**, **Таблицы**, **Рисунки и элементы управления**, **Линейка**, **Горизонтальная и вертикальная полосы прокрутки**;

– в разделе **Настройки языка** / **Лингвистика** включите команды **Автоматически проверять орфографию** и **Автоматически проверять грамматику**;

– в разделе **Загрузка / сохранение** в пункте **Общие** выберите **Автосохранение каждые 10 минут.** Просмотрите установки в других разделах. Для реализации сделанных установок щелкните по кнопке **ОК.**

2. Установите параметры страницы.

Параметры страницы настраиваются с помощи меню **Формат / Страница.** В открывшемся окне выберите вкладку **Страница**, где могут быть установлены размер листа и его ориентация, поля и другие параметры. Задайте нужные параметры: размер бумаги А4, ориентация – книжная, верхнее и ниж-нее поля – по 2 см, левое и правое поля – по 1,5 см.

Обратите внимание, что поля документа также можно менять с помощью линеек страниц (рис. 2), на которых поля отображаются серыми областями. Для изменения полей с помощью линейки:

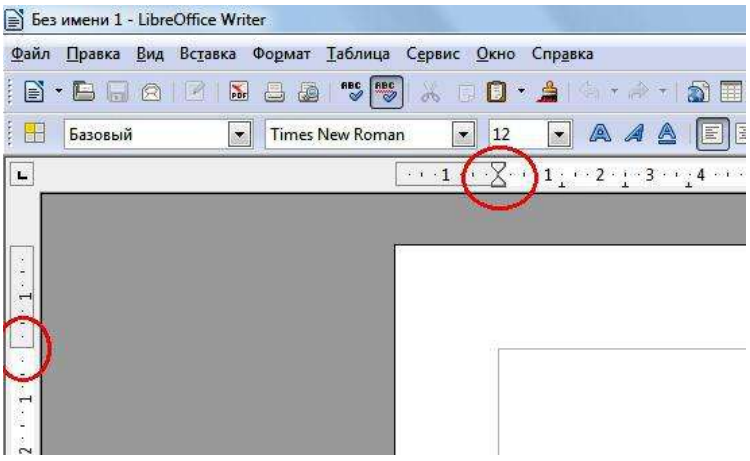


Рис. 2. Границы полей документа

– поместите курсор мыши на линию раздела серой и белой областей. Курсор изменит свое изображение на двойную стрелку;

– удерживайте нажатой левую кнопку мышки и перетаскивайте границу поля до нужного размера.

3. Создайте документ.

3.1. Ввод текста начните с вставки текущей даты. Для этого выполните команду **Вставка / Поля / Дата**. Выровняйте дату по правому краю с помощью команды **Формат / Выровнять / По правому краю** или с помощью значка  на панели инструментов «Форматирование».

3.2. Создайте **Автотекст**, содержащий ваши данные (факультет, специальность, номер группы, фамилия, имя). Так как эти данные будут использованы в следующих работах, то включение их в список функции **Автотекст** позволит быстро вставить данные в документ.

Для создания и вставки автотекста:

– на следующем абзаце после вставленной даты введите ваши данные (факультет, специальность, номер группы, фамилия, имя). Выделите набранный текст и скопируйте (команда **Правка / Копировать** или значок  на панели инструментов «Форматирование» или клавиши **<Ctrl> +<C>**);

– выберите команду **Правка / Автотекст** (рис. 3);

– выберите категорию, в которой требуется хранить автотекст (**Мой Автотекст**);

– установите курсор в пустое окошко около раздела **Автотекст** и клавишами **<Ctrl> +<V>** вставьте ваши данные. В разделе **Сокращение** можно задать имя. Это позволит использовать сокращение в качестве подсказки при вводе. Установите галочку у параметра «Предлагать замену при наборе»;

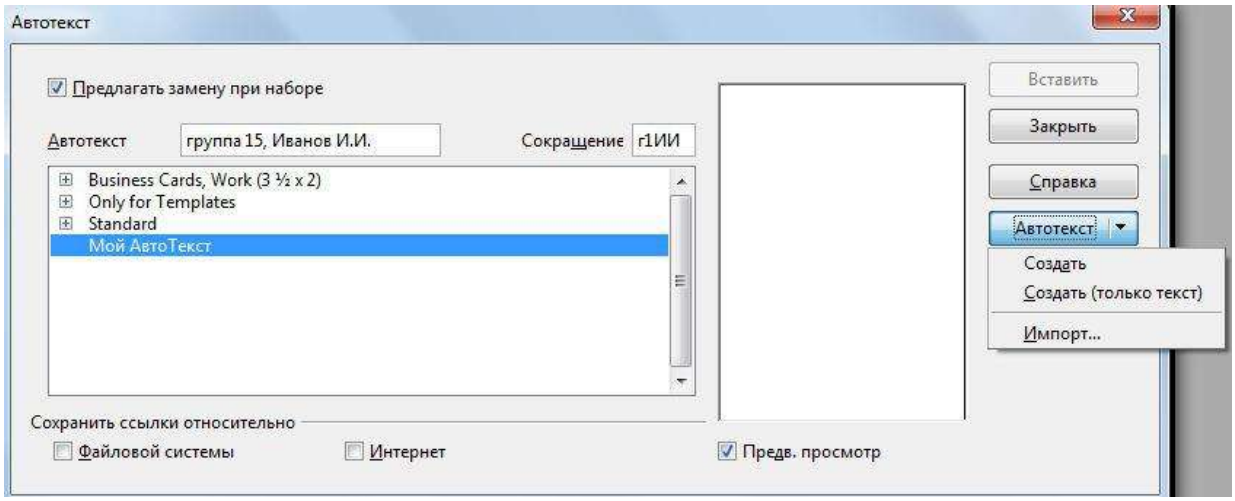


Рис. 3. Создание автотекста

– нажмите кнопку **Автотекст** и выберите команду **Создать;**

– нажмите кнопку **Закрыть;**

– теперь ваши данные остались в памяти функции **Автотекст** и вы сможете вставить их в любой документ в любое время;

– проверьте работу **Автотекста**. Удалите ваши данные и заново вставьте их командой **Правка / Автотекст / Мой Автотекст / Вставить.**

3.3. Ознакомьтесь с работой функции **Автозамены.**

По умолчанию LibreOffice автоматически исправляет многие часто встречающиеся ошибки и применяет форматирование при вводе текста. Посмотреть полный перечень замен можно, открыв окно автозамены командой **Формат / Автозамена / Параметры автозамены**.

Создайте свою замену для слов, часто встречающихся при наборе, например, LibreOffice (рис. 4):

– в разделе **Заменять** введите сокращение, например, две буквы – *ло*.

В разделе **На** введите полное название – *LibreOffice*;

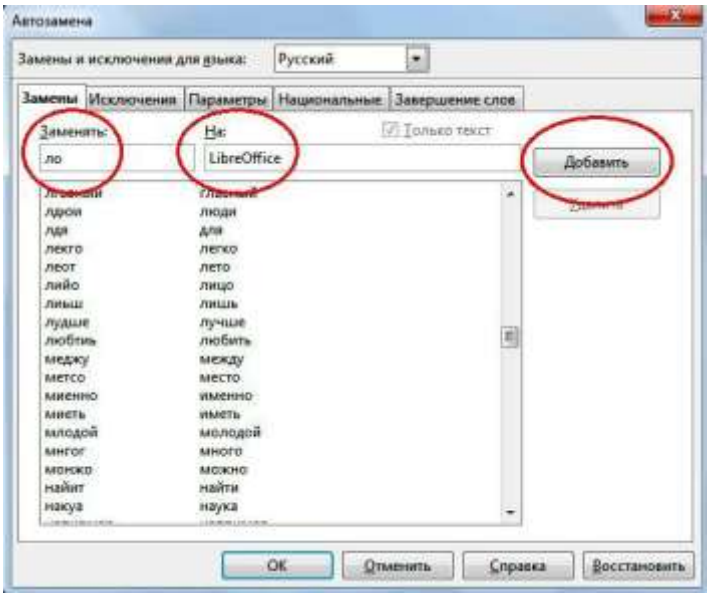


Рис. 4. Создание автозамены

– теперь при наборе текста вместо слова *LibreOffice* вы можете печатать две буквы *ло*, которые автоматически заменятся на *LibreOffice*;

– проверьте работу функции, напечатав на листе буквы *ло* и удостоверившись, что они заменились на *LibreOffice*;

– аналогичным образом создайте автозамену для слова *Writer*.

Обратите внимание, что LibreOffice сохраняет слова, часто используемые в текущем сеансе. При вводе первых трех букв такого слова LibreOffice автоматически подставляет остальную часть слова. Если в памяти автозамены имеется несколько слов, соответствующих трем введенным буквам, нажмите **<Ctrl> +<Tab>** для прокрутки доступных слов. Для прокрутки в обратном направлении нажмите клавиши **<Ctrl> + <Shift> +<Tab>**. Если вы подтверждаете подстановку слова, то предлагаемый вариант завершения сло-

ва принимается нажатием клавиши <**Enter**>. Чтобы отклонить вариант завершения слова, продолжайте печатать с любой другой клавиши.

3.4. Введите приведенный ниже текст.

Дата: (текущая)

Данные о студенте: (факультет, специальность, номер группы, фамилия, имя)

Writer – текстовый процессор LibreOffice

Writer – текстовый процессор для всех видов документов. Writer со-держит все необходимые функции современного полнофункционального текстового редактора и инструмента публикаций.

Он достаточно прост для создания быстрых заметок и достаточно мощен, чтобы создать целые книги с содержанием, диаграммами, индекса-ми и т.д.

«Мастера» берут на себя всю работу по созданию стандартных доку-ментов, таких, как письма, факсы, повестки дня, протоколы, а также могут выполнить более сложные задачи, например, создание множества докумен-тов из одного шаблона и источника данных для рассылки разным адресатам.

«Стили и форматирование» предоставляют каждому пользователю всю мощь таблиц стилей.

Поиск опечаток «на лету» и словарь автозамены проверят орфографию. Если вам необходимо использовать различные языки в документе, Writer способен и на это.

Сокращение усилий и времени на ввод текста достигается благодаря функции автозавершения, которая предлагает варианты быстрого ввода и по-зволяет завершить не до конца набранные слова и фразы (наиболее часто встречающиеся).

Текстовые структуры и линковка позволяют заниматься компьютерной верстkой информационных бюллетеней, листовок и т.д. Сделайте ваши длин-ные и сложные документы более полезными, добавив оглавление, сноски, библиографические ссылки, иллюстрации, таблицы или другие объекты.

Привлекательные примечания позволяют отображать заметки на полях документов. Благодаря этому их значительно проще читать. Кроме того, при-мечания от разных пользователей отображаются разными цветами и содер-жат дату и время редактирования.

Выкладывайте ваши документы в свободный доступ в Интернет, пред-варительно экспортировав их с помощью Writer в HTML или в формат MediaWiki для публикации в вики. Опубликуйте документ в формате Portable Document Format (.pdf), чтобы гарантировать, что ваш читатель увидит имен-но то, что вы написали. Функция экспорта в

PDF в LibreOffice содержит ог-ромный набор параметров форматирования и безопасности.

Сохраняйте документы в формате OpenDocument – новом междуна-родном стандарте для офисных документов. Этот формат, основанный на XML, не «привязывает» вас к Writer. Вы сможете открывать документ в этом формате с помощью любого программного обеспечения, совместимого

с OpenDocument.

Writer, конечно, сможет открыть ваши старые документы Microsoft Word или сохранить работу в формате Microsoft Word. Writer может также открывать .docx файлы, созданные в Microsoft Office 2007 или Microsoft Office 2008 для Mac OS X.

4. Проверьте набранный текст на наличие ошибок.

Проверка орфографии начинается с текущего положения курсора или

с начала выделенного текста. Для проверки орфографии:

– щелкните внутри документа или выделите текст, который необходимо проверить;

– выберите команду **Сервис / Правописание и грамматика;**

– при обнаружении возможной орфографической ошибки откроется диалоговое окно **Проверка орфографии**, в котором LibreOffice будут предложены варианты для исправления. Чтобы принять вариант исправления, щелкните его, а затем нажмите кнопку **Заменить**. Либо проигнорируйте исправление кнопкой **Пропустить**. Чтобы добавить неизвестное слово в пользовательский словарь, нажмите кнопку **Добавить.**

5. Сохраните документ в личной папке под именем **«Задание 1».**

Контрольные вопросы

1. Как запустить текстовый процессор Writer?
2. Каким образом осуществляется работа со справочной системой текстового процессора?
3. Какие особенности можно выделить у различных режимов отображения документов?
4. Каким образом можно отобразить или скрыть панели документов?
5. Как установить параметры работы текстового процессора?
6. Как установить поля документа?
7. Как автоматически вставить в текст документа дату и время?
8. Как создать автотекст?
9. Для чего нужна автозамена и как ее создать?
10. Как проверить правописание в тексте?
11. Как исправить ошибки правописания?

12. Как выполняется сохранение документа?
13. Как сохранить документ под другим именем?
14. Что нужно сделать для выхода из программы?
15. Как открыть существующий документ?

Лабораторная работа №2

РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ДОКУМЕНТА

2. Редактирование и форматирование документа

Цель работы: сформировать навык по выполнению основных приемов редактирования и форматирования текста (выделение, копирование, перемещение и вставка текста, поиск и замена фрагментов текста, изменение формата символов и абзаца и др.)

Теоретические сведения

Редактирование документа осуществляется как в процессе ввода текста, так и после него. Редактирование документа (от лат. redactus – «приведенный в порядок») – это внесение изменений в содержимое документа. Кроме того, к редактированию относятся выявление и устранение ошибок в тексте, проверка правописания. В процессе редактирования текста можно выделить различные уровни: редактирование символов, слов, строк и фрагментов текста.

В Writer имеются различные средства для выполнения редактирования текста документа. Используется технология «перетащи и отпусти» (Drag and Drop); применяются команды **Выделить, Вырезать, Копировать, Вставить че-рез буфер обмена, Найти и заменить, Проверка правописания** и другие. Отработать их вы сможете в процессе выполнения лабораторной работы. Следует помнить, что любое действие с текстом – перемещение, копирование, проверка ошибок – начинается с его выделения. И только после этого фрагмент можно переместить, скопировать, удалить, выровнять или переформатировать. В основ-ном средства редактирования текста или группы команд для работы с текстом размещены в пункте меню **Правка** или на панели инструментов **Стандартная.**

Процесс формирования внешнего вида документа в целом или его фрагментов в любой программной среде называют форматированием (от слова «форма»). Различные способы и инструменты форматирования, которые предоставляет текстовый процессор Writer, позволяют получить профессионально оформленный текст. С помощью команд, расположенных в пункте меню **Формат**, Writer позволяет провести форматирование символов, абзацев и страниц.

Ход работы

1. Запустите текстовый процессор Writer.
2. Откройте документ **«Задание 1»**, созданный на предыдущем занятии.
3. Установите режим разметки печати (**Вид / Разметка печати**).
4. Отработайте приемы выделения фрагментов текста: слова, предложения, строки, нескольких строк, большого фрагмента.

Существуют различные способы выделения фрагментов текста – с помощью клавиатурных клавиш или манипулятора «мышь».

Для выделения фрагмента текста с помощью клавиш необходимо установить курсор в начало выделения и применить одновременное нажатие клавиши **<Shift>** и клавиш перемещения курсора для выделения фрагмента текста.

Выделение фрагмента текста с помощью мыши производится на уровне:

- отдельных символов, слов, строк текста – установить указатель мыши в начало выделения и, держа нажатой левую кнопку, протащить мышь до конца выделяемого фрагмента;
- прямоугольного фрагмента – установить указатель мыши в начало выделения, при нажатой клавише **<Alt>** и левой кнопке протаскивать мышь как по горизонтали, так и по вертикали;
- отдельного слова – установить указатель мыши на слово и сделать двойной щелчок левой кнопкой мыши;
- отдельного абзаца – установить курсор в произвольное место абзаца

и сделать тройной щелчок левой кнопкой мыши;

- одной строки – сделать одинарный щелчок левой кнопкой мыши слева от строки текста;
- группы строк текста – сделать одинарный щелчок левой кнопкой мыши слева от начала текста и вертикально протаскивать мышь до конца фрагмента;
- объекта (рисунка, формулы, диаграммы) – установить курсор на объекте и сделать одинарный щелчок левой клавишей мыши.

Выделение текста всего документа выполняется с помощью команд **Правка, Выделить все** или одновременным нажатием клавиш **<Ctrl> + <A>.**

В текстовом процессоре Writer внизу рабочего окна на строке состояния действует отображение текущего режима выделения.

5. Отработайте приемы копирования, перемещения и удаления текста.

Выделенный фрагмент может быть перенесен или скопирован через буфер обмена как в любое место активного документа, так и в другой документ. Используется любой из предложенных методов:

– кнопки **Вырезать**  , **Копировать**  , **Вставить**  ;

– соответствующие команды контекстного меню, вызываемого щелчком правой кнопки мыши или меню **Правка;**

– сочетания клавиш: **<Ctrl> + <X>** – вырезать, **<Ctrl> + <C>** – скопировать в буфер, **<Ctrl> + <V>** – вставить из буфера.

При переносе фрагмента на небольшое расстояние более удобен другой способ: поместите указатель мыши в выделенную область (вид указателя – стрелка) и перетащите фрагмент левой кнопкой мыши в нужное место (указатель изменит вид – теперь это будет стрелка с прямоугольником внизу). Подобным образом фрагмент можно скопировать: те же действия производятся при нажатой клавише **<Ctrl>.** Выделенный фрагмент может быть удален нажатием клавиши удаления **<Delete>**.

Используя описанные технологии, отредактируйте документ следующим образом:

– выделите первый абзац, скопируйте его и вставьте в конец текста, отмените последнее действие клавишей  ;

– выделите абзац *«Мастера…»*, переместите его на абзац ниже;

– Выделите абзац *«Привлекательные примечания…»*, вырежьте его и вставьте после абзаца *«Поиск опечаток…»*.

6. Просмотрите отредактированный текст. При работе с текстом иногда полезно видеть, где нажата клавиша **<Enter>**, сколько раз нажата клавиша пробела и т.п., т.е. видеть специальные непечатаемые символы. Для этого используется кнопка **Непечатаемые символы** в стандартной панели инструментов (¶). Нажмите эту кнопку и просмотрите обозначения и расстановку таких символов. Отключите или оставьте включенной кнопку **Непечатаемые символы** для удобства дальнейшей работы.

7. Отформатируйте заголовок текста.

Выделите заголовок. Выполните команду **Формат / Символы**. В открыв-шемся окне выполните следующие действия:

– оформите заголовок в соответствии с параметрами: шрифт Arial, кегль 16, полужирный, цвет синий, эффект «Тень», интервал, разреженный на 4 пт.;

– просмотрите все возможные настройки, отраженные на вкладках этого окна: **Эффекты шрифта, Положение, Гиперссылка, Фон**. Пронаблюдайте, как они меняют написание текста;

– выровняйте заголовок по центру страницы.

8. Отформатируйте основной текст.

Выделите основной текст. Командой **Формат / Символы** оформите его в соответствии с параметрами: шрифт Arial, кегль 14. Установите параметры абзаца (**Формат / Абзац**): первая строка – отступ 1,5 см, выравнивание – по ширине, междустрочный интервал – полуторный.

9. Найдите в тексте англоязычные слова. Оформите их *курсивом*.

10. Выделите только первый абзац и в диалоговом окне **Формат / Абзац** на вкладке **Буквица** добавьте параметр **Добавить буквицу.**

11. Для точной настройки параметров абзаца можно использовать меню **Формат / Абзац.** Просмотрите все возможные настройки, отраженные на вкладках этого окна.

Перейдите на вкладку **Отступы и интервалы**. Найдите на вкладке настройку выравнивания текста, его стиль (основной текст); отступы, установленные для указанного абзаца:

– слева – отступ всего абзаца влево от поля;

– справа – отступ всего абзаца вправо от поля;

– перед абзацем – отступ всего абзаца вниз от предыдущего абзаца;

– после абзаца – отступ всего абзаца вверх от последующего абзаца.

Выделите абзац *«Выкладывайте ваши документы в свободный доступ в Интернет…»* и установите отступ слева 1 см, отступ справа 1 см, интервал перед абзацем 1 см, интервал после абзаца 1 см. Посмотрите, как изменился текст. Отмените последние действия.

12. Выполните команду **Правка / Найти и заменить**. Используя средства автоматического поиска и замены, найдите все упоминания в тексте Microsoft и замените их на MS.

13. Освойте разбиение текста по колонкам.

Выделите весь текст, кроме первого абзаца. Выполните команду **Формат / Колонки** и выберите разбиение на две колонки; расстояние между колонками 0,5 см.

14. Выполните предварительный просмотр документа (**Файл / Предварительный просмотр страницы**).

15. Сохраните документ в личной папке под именем **«Задание 2».**

Контрольные вопросы

1. Раскройте понятия редактирования и форматирования текста.
2. Какие существуют способы выделения фрагментов текста?
3. Как провести выделение фрагмента текста с помощью клавиш?
4. Как можно выделить отдельные символы, слова, строки текста?
5. Каким образом выделить прямоугольный фрагмент текста?
6. Как выделить отдельное слово (абзац)?
7. Как провести переключение режимов выделения в Writer?
8. Как образом можно копировать, перемещать и удалять текст?
9. Как включить режим «Непечатаемые символы» и чем он удобен для работы?
10. Как изменить формат символа (абзаца)?
11. Как провести автоматический поиск и замену?
12. Какой командой можно оформить абзац с помощью буквицы?
13. Какой командой производится разбиение текста по колонкам?

2. ТЕХНОЛОГИЯ РАБОТЫ С БАЗАМИ ДАННЫХ

Лабораторная работа №3

СОЗДАНИЕ БАЗЫ ДАННЫХ

ЗАДАНИЕ

Задание 1

Получите у преподавателя вариант задания. Создайте средствами СУБД ACCESS базу данных, соответствующую вашему варианту задания.

1.1 Создайте структуры трех таблиц из вашего варианта задания. ***Данные в таблицы пока не вводите!***

1.2. Создайте схему данных.

Задание 2

Введите в каждую из таблиц вашей базы данных по 5 записей (используйте режим таблицы).

Задание 3

3.1. Проверьте обеспечение сущностной целостности данных. Попробуйте ввести повторяющиеся данные в ключевое поле любой из ваших таблиц. Как на это реагирует система?

3.2. Попробуйте ввести текстовые данные в числовое поле. Как на это реагирует система?

3.3. Проверьте обеспечение ссылочной целостности данных. Попробуйте удалить запись из таблицы связи. Как на это реагирует система?

ТЕХНОЛОГИЯ РАБОТЫ

Технологию создания базы данных продемонстрируем на примере создания базы данных **Заказ товаров по каталогу,** данные в которой будут храниться в трех таблицах : КЛИЕНТЫ, ТОВАРЫ и ЗАКАЗЫ.

Таблица Клиенты

Код клиента	Фамилия Имя Отчество	Адрес	Телефон

Таблица Товары

Код товара	Наименование	Размер	Цвет	Цена

Таблица Заказы

Код клиента	Код товара	Количество единиц товара	Дата приема заказа	Дата выполнения заказа

Создание структуры базы данных

Работа с новой базой данных начинается с создания структуры базы данных. Для каждой таблицы задается ***структура записи*** (имена полей и типы данных в каждом поле). Структура записи – это «заготовка» будущей таблицы, её каркас. Для обеспечения целостности данных между таблицами задаются связи путем построения Схемы данных. После этого в таблицы вводятся данные.

Запускаем на выполнение ACCESS 2007 и видим на экране страницу **Приступая к работе с Microsoft Office Access.**

В разделе Новая пустая база данных выбираем команду Новая база данных.

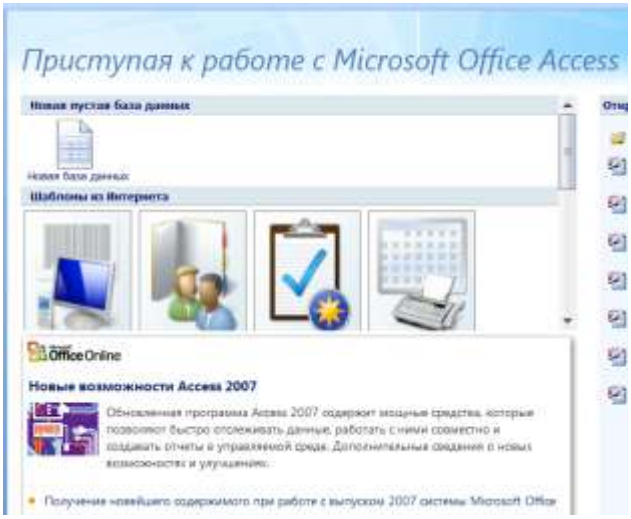


Рис.2. Страница Приступая к работе с Microsoft Office Access

В области Новая база данных справа от страницы Приступая к работе с Microsoft Office Access в поле Имя файла вводим имя файла.

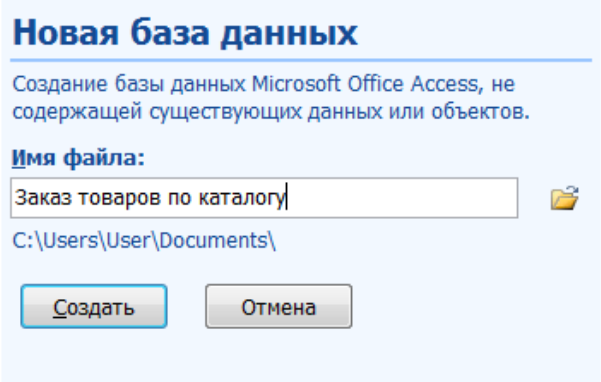


Рис.3. Ввод имени файла базы данных

Чтобы сохранить файл в своей папке, отличной от используемой по умолчанию, нажимаем кнопку Открыть (рядом с полем Имя файла), переходим к нужной папке и нажимаем кнопку ОК, а затем - кнопку Создать.

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и открывает эту таблицу в режиме таблицы.

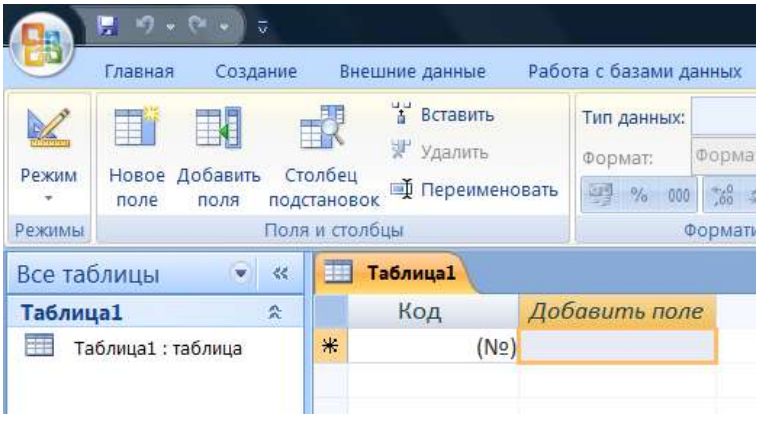


Рис.4. Таблица1 в режиме таблицы

Таблицы являются основой базы данных. В них хранятся данные. Начнем с создания таблицы КЛИЕНТЫ. Для этого

перейдем в режим Конструктора таблиц, нажав на кнопку 

Программа предлагает сохранить создаваемую таблицу

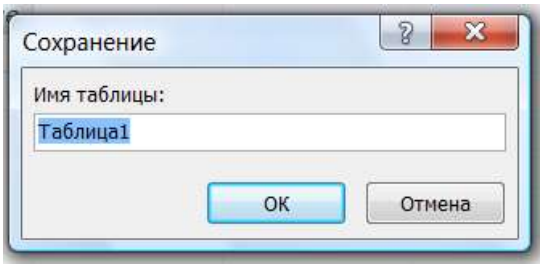


Рис.5. Сохранение таблицы

Введем в поле Имя таблицы вместо предлагаемого системой имени Таблица1 имя ***Клиенты***.

На экране появится бланк для ввода данных о структуре таблицы.

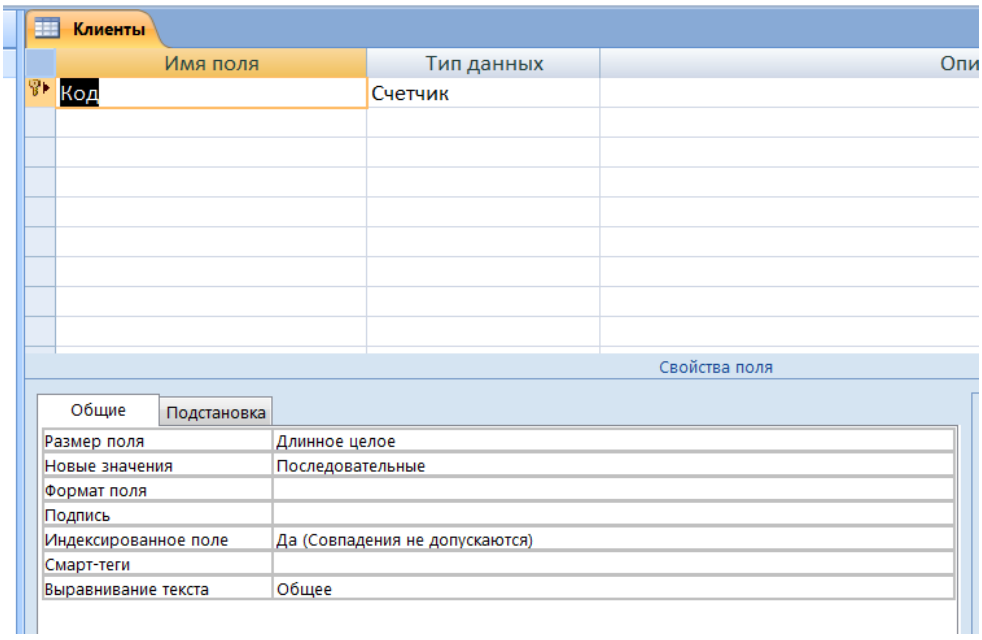


Рис.6. Бланк для ввода данных о структуре таблицы.

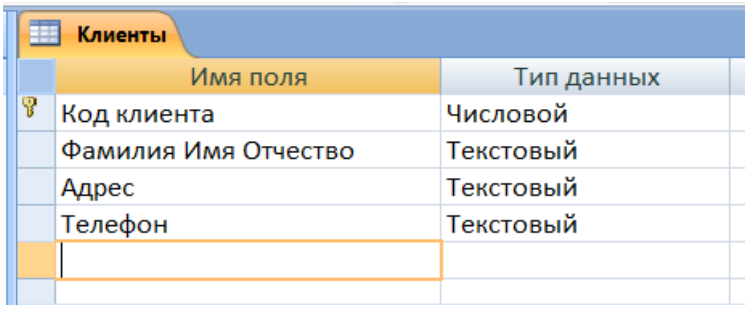
В системе ACCESS столбцы таблицы называются ***полями***, а строки – ***записями***. В таблице КЛИЕНТЫ 4 столбца, поэтому каждая запись данной таблицы будет состоять из 4-х полей.

Вместо предлагаемого программой имени первого поля **Код** вводим имя первого поля "Код клиента". Переходим в колонку Тип данных и, щелкнув на кнопке раскрытия списка типов полей, выбираем тип ***Числовой***. Код клиента был введен в таблицу Каждый клиент имеет уникальный код в базе данных. Программа автоматически делает это поле ключевым, о чем говорит изображение ключика.

Ключевое поле содержит значения, которые не повторяются, поэтому могут использоваться для однозначной идентификации записей о клиентах.

Переходим к следующей строке бланка конструктора и вводим имя поля Фамилия Имя Отчество. Программа предлагает текстовый тип данных. Это то, что нам нужно.

Остается ввести в следующие строки бланка имена полей таблицы **Адрес** и **Телефон**, для которых также подойдет текстовый тип данных.



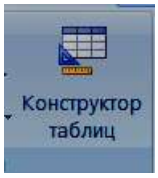
.

Рис.7. Структура записи таблицы КЛИЕНТЫ

Мы ввели данные обо всех полях таблицы, поэтому заканчиваем создание таблицы, закрывая окно конструктора таблиц.

В появившемся диалоговом окне нажимаем кнопку «Да», чтобы сохранить созданную структуру таблицы Клиенты

Переходим к созданию таблицы Товары. Для этого надо выбрать на ленте вкладку Создание, а затем выбрать команду Конструктор таблиц



В появившийся бланк вводим сведения об именах полей и типах данных таблицы Товары:

Таблица1

Имя поля	Тип данных
Код товара	Числовой
Наименование товара	Текстовый
Размер	Числовой
Цвет	Текстовый
Цена	Денежный

Рис.8. Структура записи таблицы Товары

Поле Код товара надо назначить ***ключевым*** для данной таблицы. Для этого надо щелкнуть мышью слева от имени поля Код товара, а затем выбрать на ленте команду

Изображение ключика появится рядом с именем поля Код товара.

Закрываем окно конструктора таблиц и сохраняем таблицу с именем Товары.

Нам осталось создать таблицу Заказы. Повторяем описанные выше действия, чтобы с помощью конструктора создать следующую структуру таблицы:

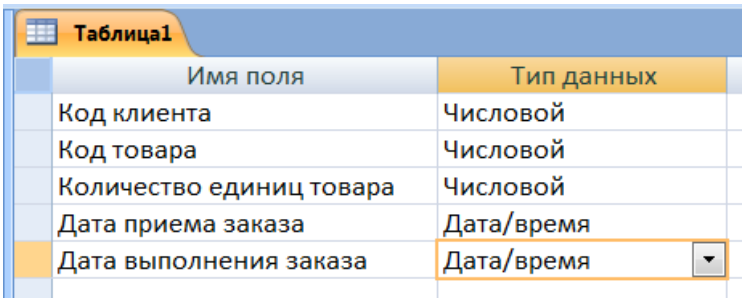
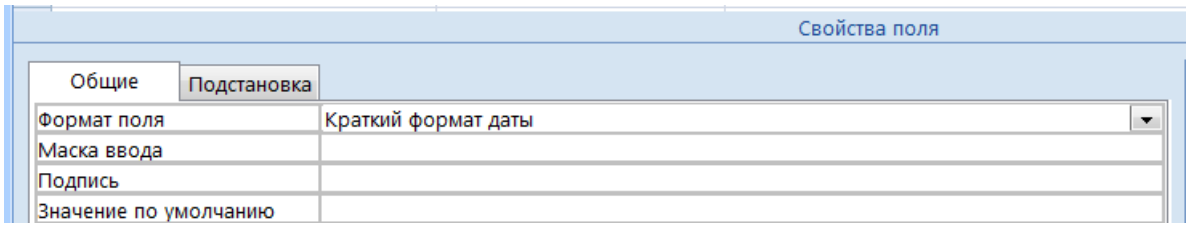


Рис.9. Структура записи таблицы Заказы

Для полей Дата приема заказа и Дата выполнения заказа в разделе Свойства поля надо в строке Формат выбрать Краткий формат даты

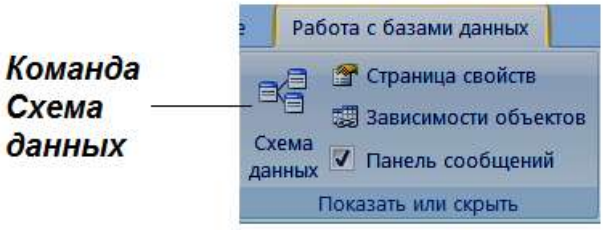


Закрываем окно конструктора и вводим в окне сохранения имя таблицы Заказы. У этой таблицы не будет ключевых полей, так как любой клиент может оформить множество заказов и заказать любые товары. Отказываемся от предложения системы создать ключевое поле и завершаем создание этой таблицы.

Создание связей между таблицами

Для обеспечения целостности данных (см. «ВВЕДЕНИЕ») надо определить связи между таблицами нашей базы данных.

На ленте находим вкладку Работа с базами данных и выбираем пункт Схема данных.



Открывается диалоговое окно **Добавить таблицу**.

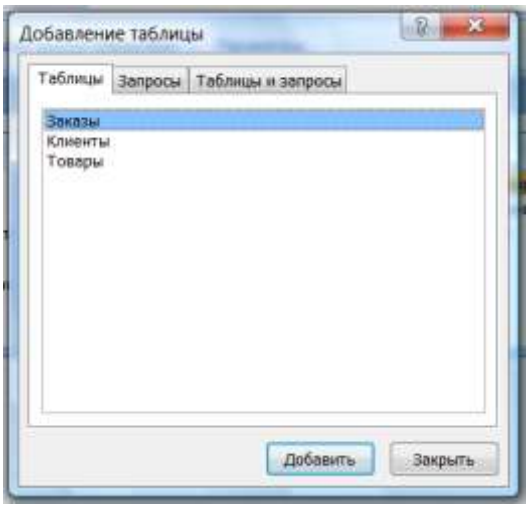
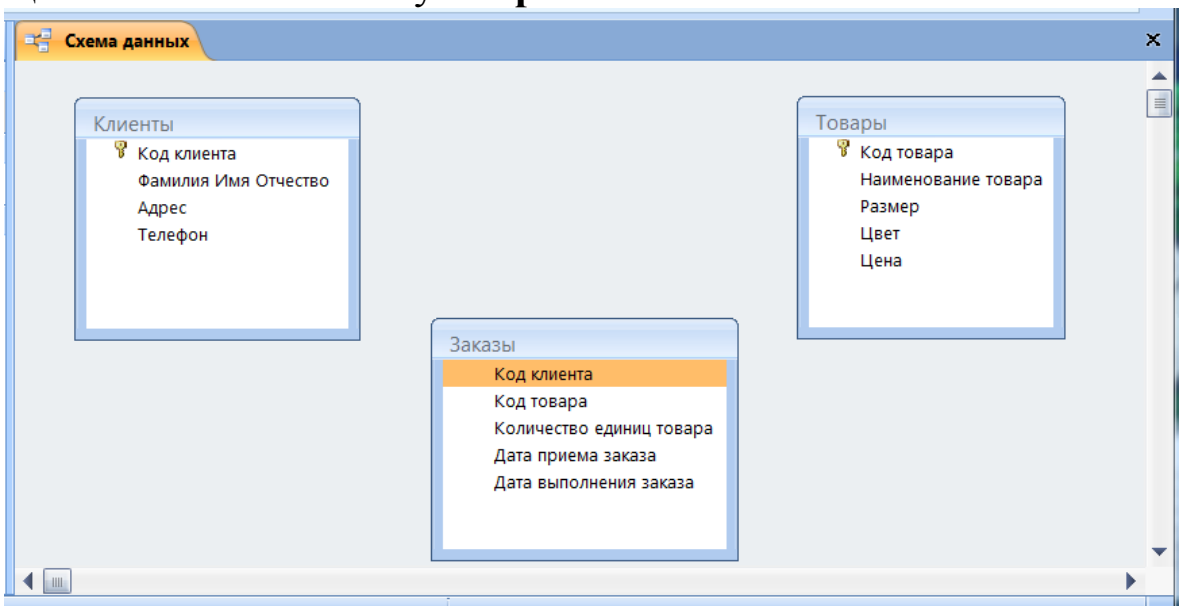


Рис.10. Добавление таблиц в схему данных

Выбираем первую по списку таблицу и нажимаем кнопку **Добавить**. Аналогично добавляем в схему данных 2 другие таблицы и нажимаем кнопку **Закрыть.**



Ри.11. Размещение таблиц в окне «Схема данных»

Ставим курсор на ключевое поле Код клиента в таблице **Клиенты** и перетаскиваем его мышкой на поле с таким же именем в таблице Заказы. В появившемся диалоговом окне Изменение связей отмечаем галочками ***Обеспечение***

целостности, ***Каскадное удаление*** и ***каскадное обновление*** и нажимаем кнопку Создать.

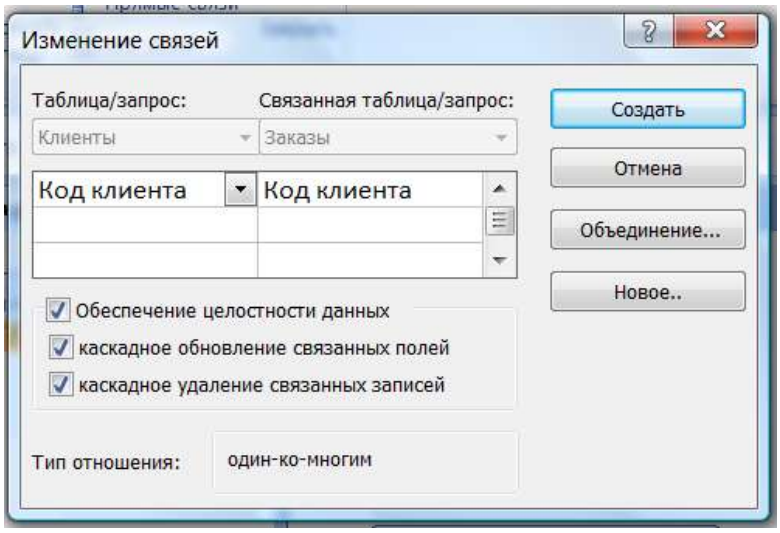


Рис.12. Окно Изменение связей

На схеме данных появляется графическое изображение связи типа «один-ко-многим» между таблицами Клиенты и Заказы.

Из таблицы Товары перетаскиваем ключевое поле Код товара на поле с таким же именем в таблице Заказы и подтверждаем условия обеспечения целостности.

В результате наших действий схема данных принимает следующий вид:

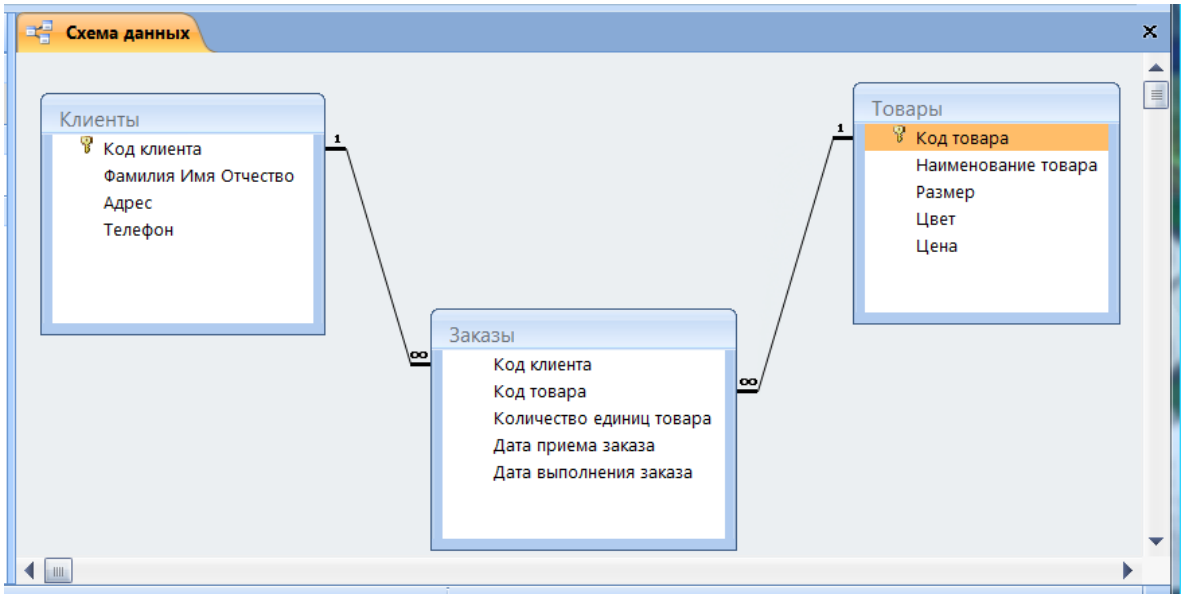


Рис.13. Схема данных

Закрываем окно Схема данных и подтверждаем ее сохранение.

Ввод данных в режиме таблицы

С данными в системе ACESS можно работать в двух основных режимах – в ***режиме таблицы*** и в ***режиме формы***. В данной работе познакомимся с работой в ***режиме таблицы***, который чем-то напоминает работу в электронных таблицах. Знакомство с формами запланировано в следующей лабораторной работе.

Мы задали связи между таблицами с автоматическим контролем целостности данных, поэтому в таблицу Заказы можно вводить только коды существующих в базе клиентов и товаров. Следовательно, сначала надо заполнить таблицы Клиенты и Товары, а затем – в таблицу Заказы.

Мы видим на экране слева имена созданных нами таблиц. Для того, чтобы получить возможность ввести в таблицу данные, надо выбрать таблицу и дважды щелкнуть мышью по значку  .

На экране появится пустая таблица, состоящая из одной записи. Новые записи будут появляться по мере заполнения бланка.

Откроем таблицу Клиенты и введем несколько записей

Клиенты

Код клиента	Фамилия Имя Отчество	Адрес	Телефон
101	Иванов Иван Иванович	г.Курчатов,ул.	4712-33-45-55
102	Васильева Василиса Петров	г.Железногорс	910-310-22-33
107	Антонова Анна Тимофеевна	г.Курск, ул. Ра	4712-59-92-09

Рис.14. Ввод записей в таблицу Клиенты

Обратите внимание на то, что система контролирует уникальность значений в ключевых полях и не допускает ввод повторяющихся значений. Система контролирует также и соответствие вводимых значений данных типам полей таблицы.

Заканчивается ввод данных закрытием окна таблицы и подтверждением сохранения данных.

Проверка обеспечение целостности данных
Сущностная целостность

Попробуем ввести повторяющиеся значения в ключевое поле таблицы. Система выдаст сообщение о недопустимости ввода повторяющихся значений в ключевое поле.

Попробуем ввести текстовые данные в числовое поле. Система выдаст сообщение о несоответствии типов вводимого значения и поля таблицы.

Ссылочная целостность

Попробуем удалить одну из записей в таблице о любом объекте, информация о котором содержится также и в таблице связи. Например, нам потребовалось удалить информацию о клиенте с кодом 110. Открываем таблицу КЛИЕНТЫ, выделяем первую запись, нажимаем правую кнопку мыши и в появившемся контекстном меню выбираем команду. **Удалить запись**. Система выдает сообщение о невозможности такого удаления

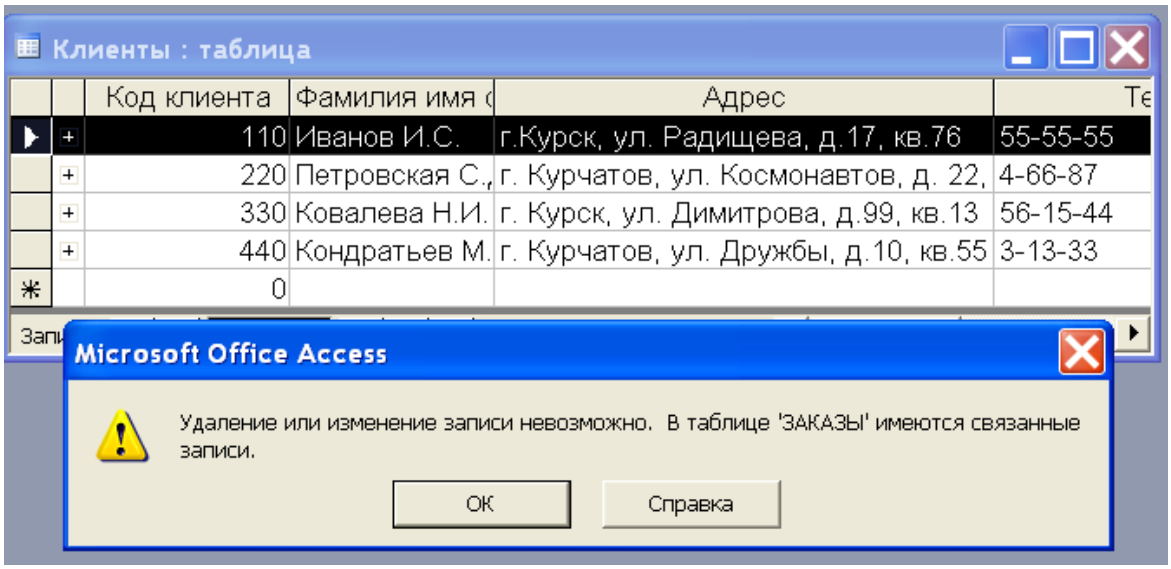


Рис.15. Сообщение о нарушение целостности при попытке удаления записи

Для того чтобы действительно удалить из базы информацию о данном клиенте, надо сначала удалить все связанные с ним записи из таблицы ЗАКАЗЫ.

Лабораторная работа №4

РАБОТА С ФОРМАМИ

Форма — это объект базы данных, который можно использовать для ввода, изменения или отображения данных из таблицы или запроса. Формы могут применяться для управления доступом к данным: с их помощью можно определять, какие поля или строки данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только нужные им поля, это облегчит для них использование базы данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы.

Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных. Рационально построенная форма ускоряет работу с базой данных, поскольку пользователям не требуется искать то, что им нужно. Внешне привлекательная форма делает работу с базой данных более приятной и эффективной, кроме того, она может помочь в предотвращении неверного ввода данных.

Задание

Создайте для каждой из таблиц вашей базы форму.
Примечание. Для таблицы, отображающей связь между данными об объектах (аналогично таблице ЗАКАЗЫ) необходимо создать ***форму с элементами управления типа «поле со списком».***
Введите в каждую из ваших таблиц по 5 записей в режиме формы.

ТЕХНОЛОГИЯ РАБОТЫ

Начнем с создания формы для работы с таблицей Клиенты. На ленте выбираем вкладку Создание. В группе Формы из

раскрывающегося списка Другие формы выбираем Мастер форм

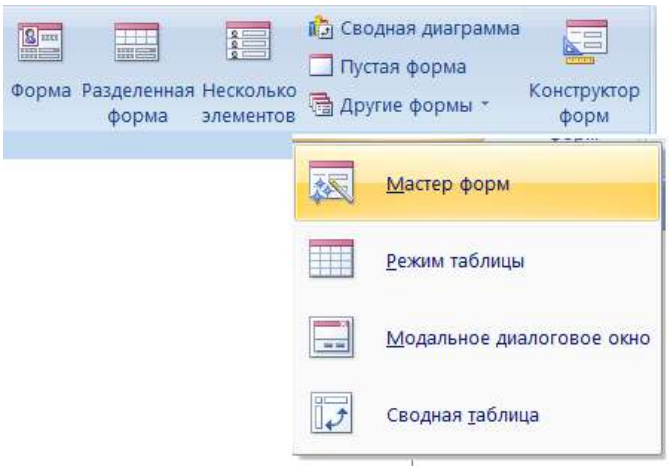


Рис.16. Вызов Мастера форм

В открывшемся окне Создание форм из списка Таблицы и запросы выбираем ***Таблица: Клиенты***

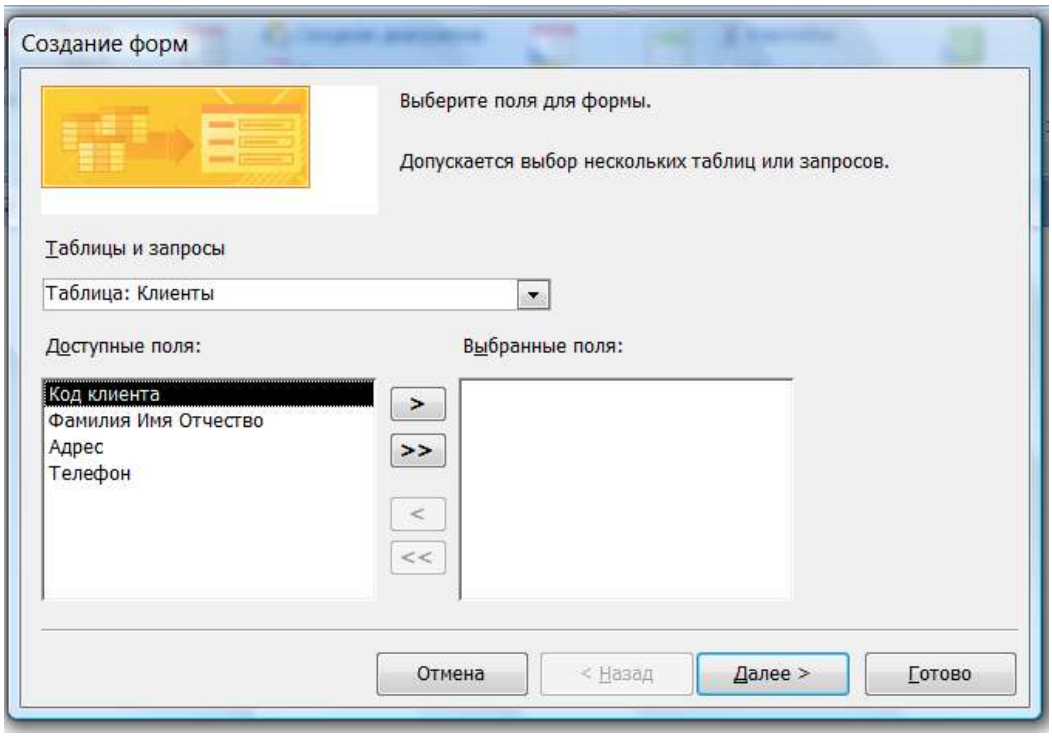


Рис.17. Выбор таблицы, для которой создается форма

Для того, чтобы поместить в форму все поля нашей таблицы, в этом же окне с помощью кнопки >> переносим все поля из списка Доступные поля в список Выбранные поля. Нажимаем кнопку Далее.

На следующем шаге выбираем внешний вид в один столбец и нажимаем кнопку Далее. Выбираем любой понравившийся стиль (внешний вид), например, Изящная и переходим к следующему шагу. Остается задать имя формы (согласимся с предлагаемым программой именем Клиенты) и нажать кнопку Готово. Перед нами открывается готовая для работы форма:

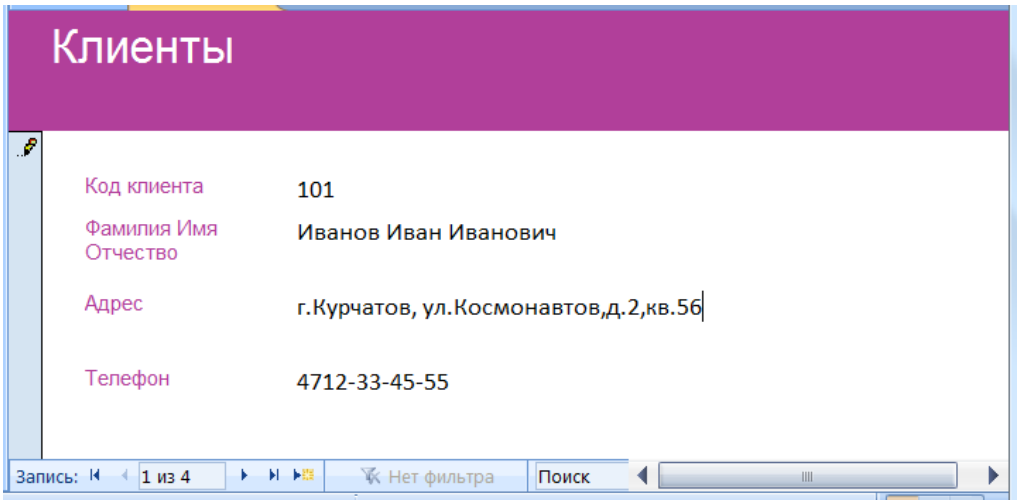


Рис.18. Форма для таблицы Клиенты

Аналогичным образом создаем форму для таблицы Товары.

Включение в форму элементов управления «поле со списком»

При вводе данных в формы обычно легче выбирать значение из списка, чем вводить его по памяти. Кроме того, список вариантов гарантирует, что в поле будут вводиться допустимые значения. Записи таблицы Заказы включают в себя коды клиентов и коды товаров, которые предварительно должны быть введены в соответствующие таблицы. Было бы удобно дать возможность работающему с базой данных человеку просто выбирать ранее введенные коды клиентов и товаров из раскрывающихся списков. Покажем, как это можно сделать.

Сначала надо создать для таблицы Заказы форму, аналогичную уже созданным нами для таблиц Клиенты и

Товары. Особенностью формы является то, что она пока будет включать не все поля. Не будем пока включать в список отображаемых полей поля Код клиента и Код товара.

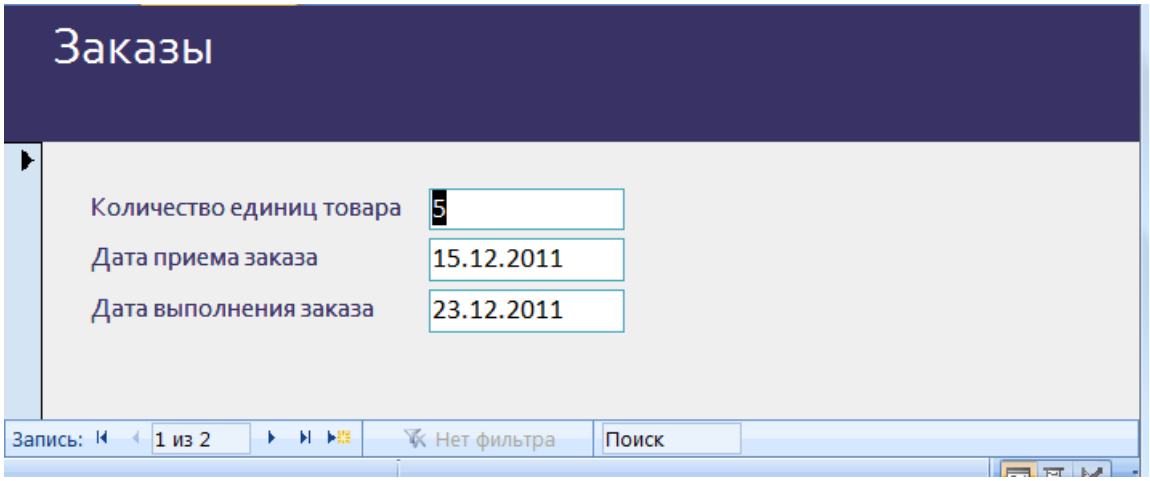


Рис.19. Предварительный вид формы

Добавим в созданную форму поле Код клиента в виде ***поля со списком***. Для этого открытую на экране форму Заказы надо закрыть.

Наводим курсор на значок формы Заказы в области переходов, нажимаем правую кнопку мыши и в появившемся контекстном меню выбираем пункт Конструктор.

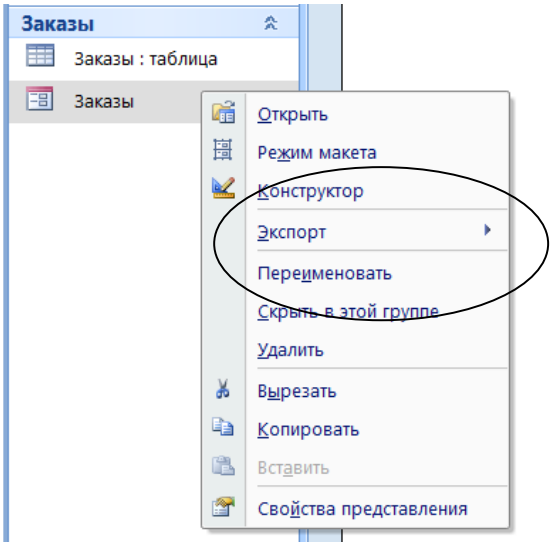


Рис.20. Вызов Конструктора форм

Открывается окно Конструктора форм с формой Заказы.

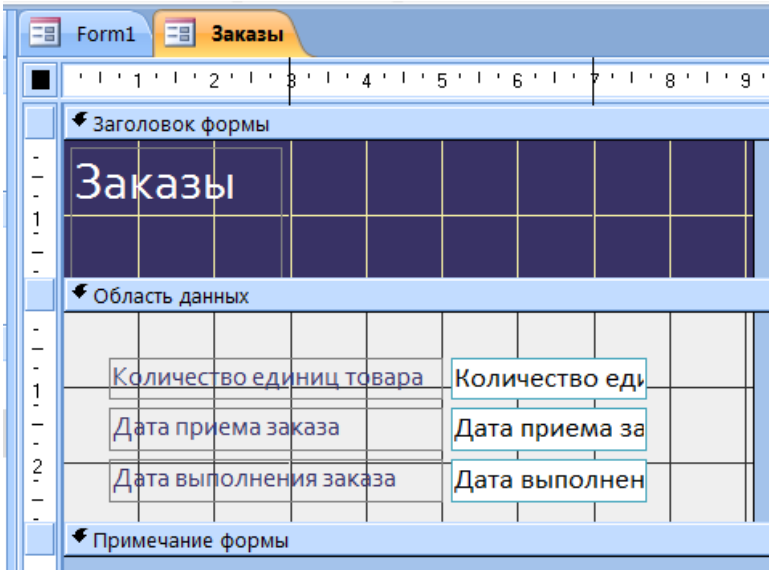


Рис.21. Окно Конструктора форм с формой Заказы

Надо приготовить на этой форме свободное место для размещения двух раскрывающихся списков. Для этого с помощью мыши надо несколько сдвинуть вниз все отображаемые поля.

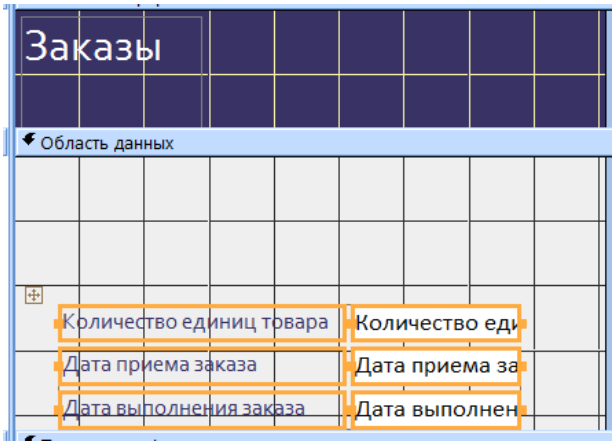


Рис.22. Подготовка места для раскрывающихся списков

На вкладке ленты Конструктор в группе Элементы управления находим элемент Поле со списком.

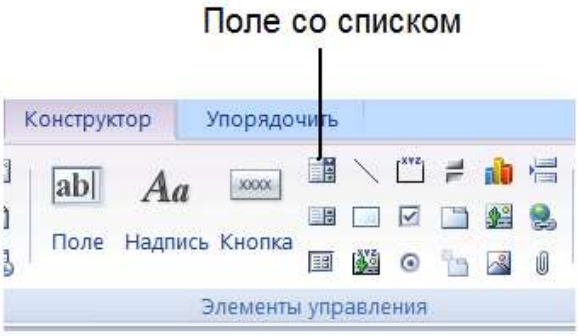


Рис.23. Элемент Поле со списком в группе Элементы управления

Щелкаем мышью по его изображению, а потом щелкаем по форме в месте, где мы хотим его расположить. Запускается Мастер создания поля со списком:

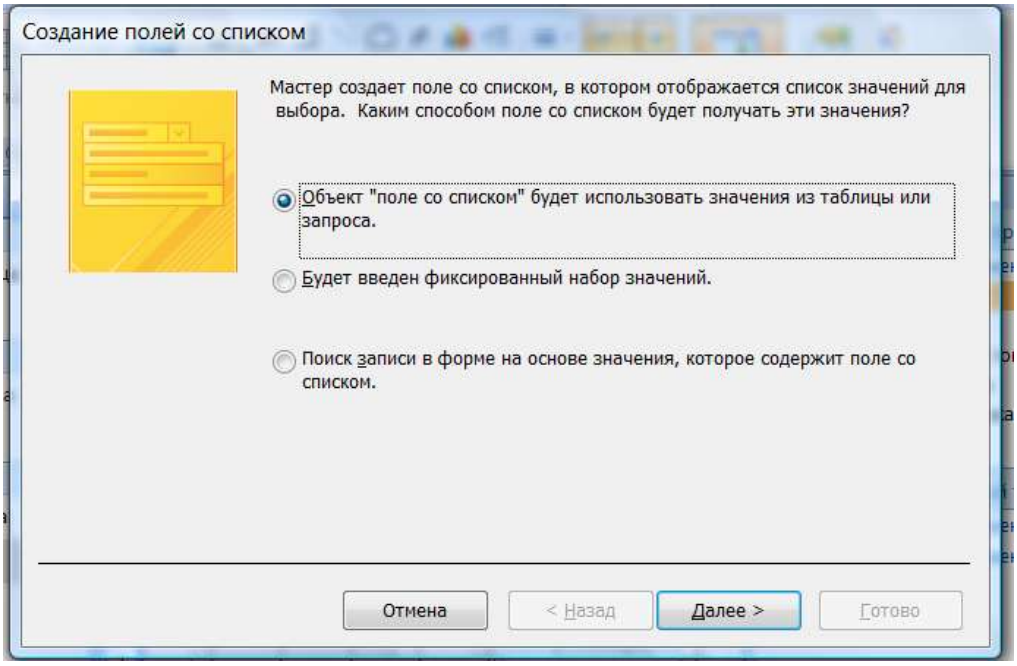


Рис.24. Окно Мастера создания поля со списком:

В нашем случае список будет получать значения из таблицы, поэтому соглашаемся с предложением программы, нажав кнопку Далее.

В следующем окне выбираем из списка таблиц таблицу Клиенты и нажимаем кнопку Далее.

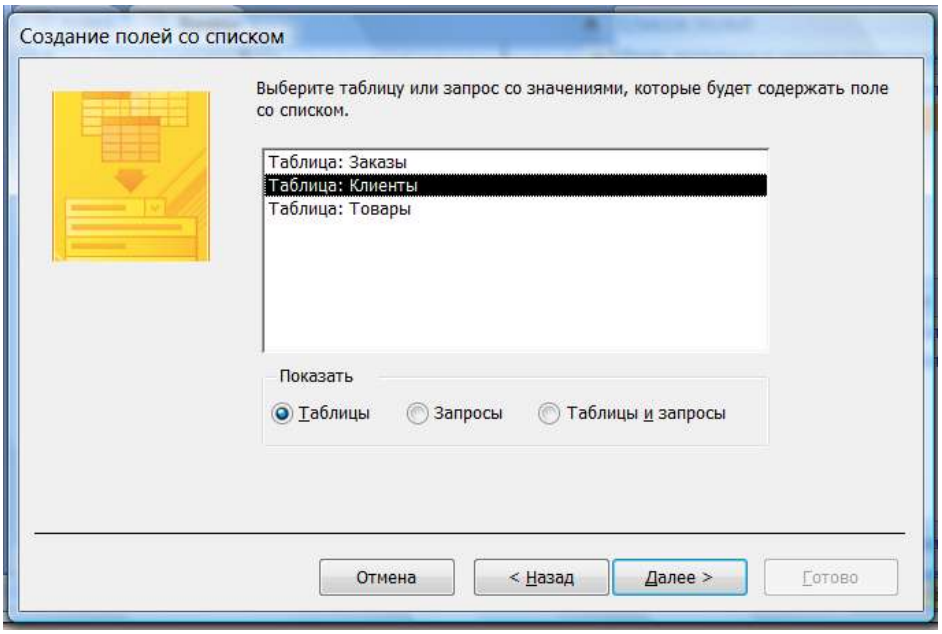


Рис.25. Выбор таблицы Клиенты

Из списка полей таблицы Клиенты выбираем поле Код клиента.

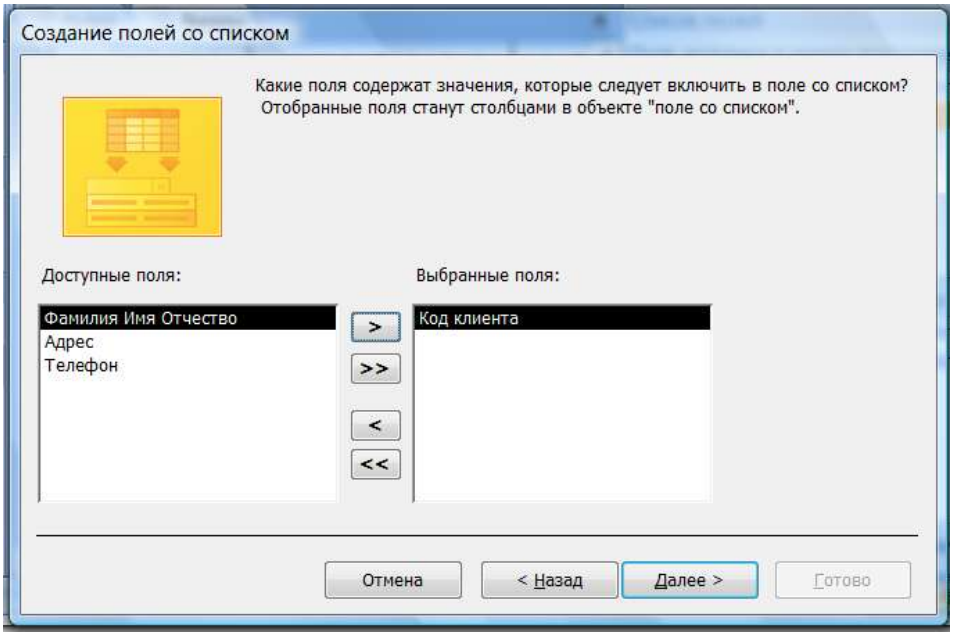


Рис.26. Выбор поля Код клиента

Значения кодов, удобнее выбирать из списка, если они упорядочены, например, по возрастанию. Надо выбрать из раскрывшегося в следующем окне списка поле Код клиента и порядок По возрастанию:

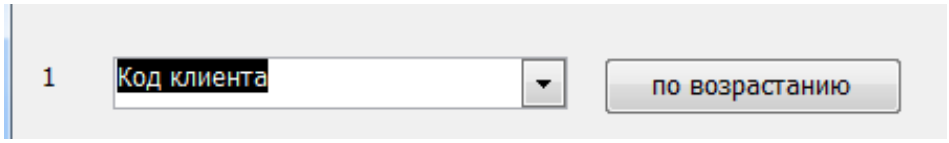


Рис.27.Задание порядка По возрастанию.

На следующем шаге Мастер предлагает установить ширину отображаемого поля данных

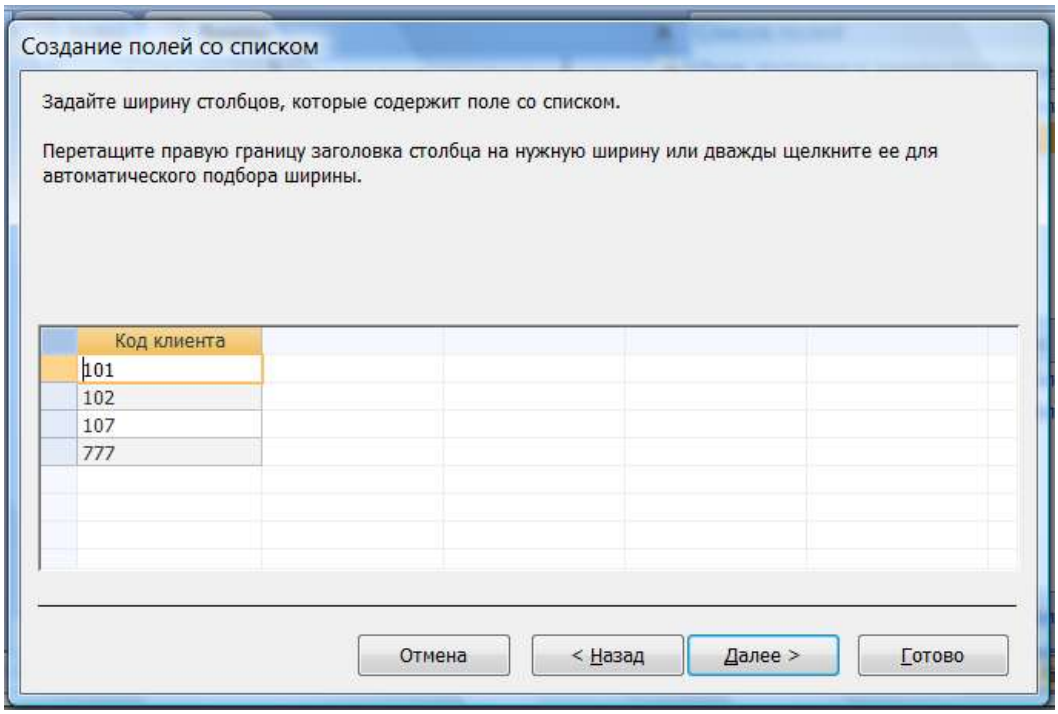


Рис.28. Установка ширины отображаемого поля данных

Выбираемые из списка значения должны сохраняться в поле Код клиента (см. рис.29).

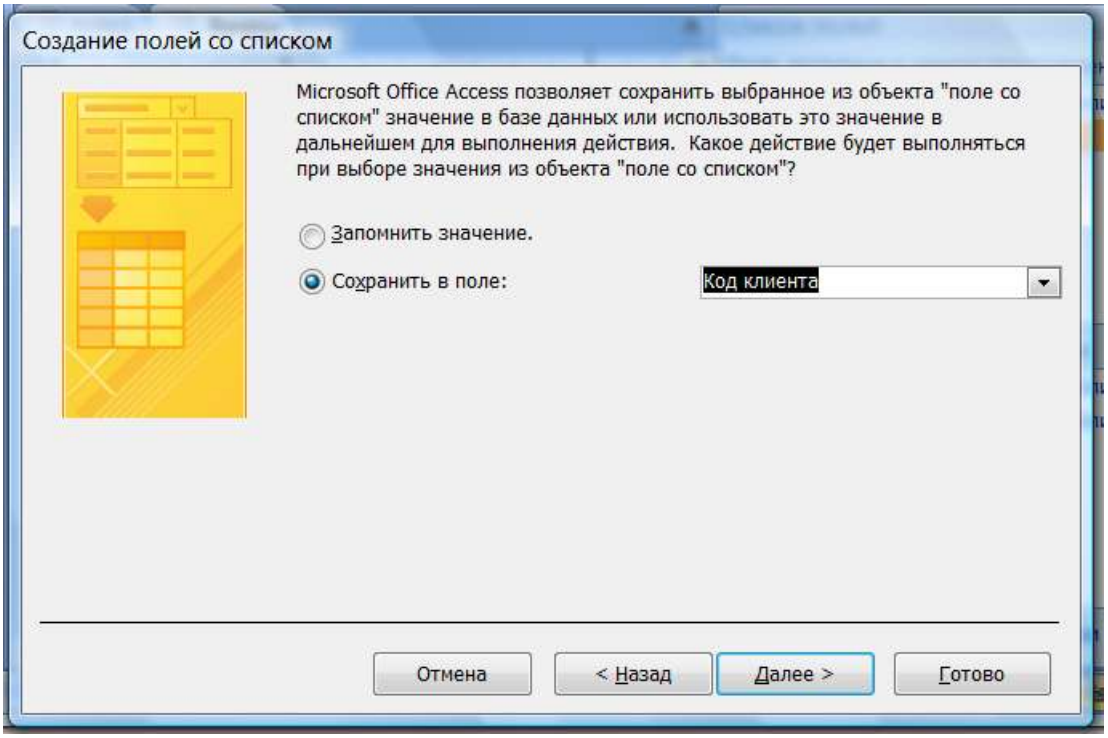


Рис.29. Задание имени поля для сохранения выбранного значения

Пользователю должно быть понятно, с каким полем он работает, поэтому в качестве подписи вводим имя поля Код клиента:

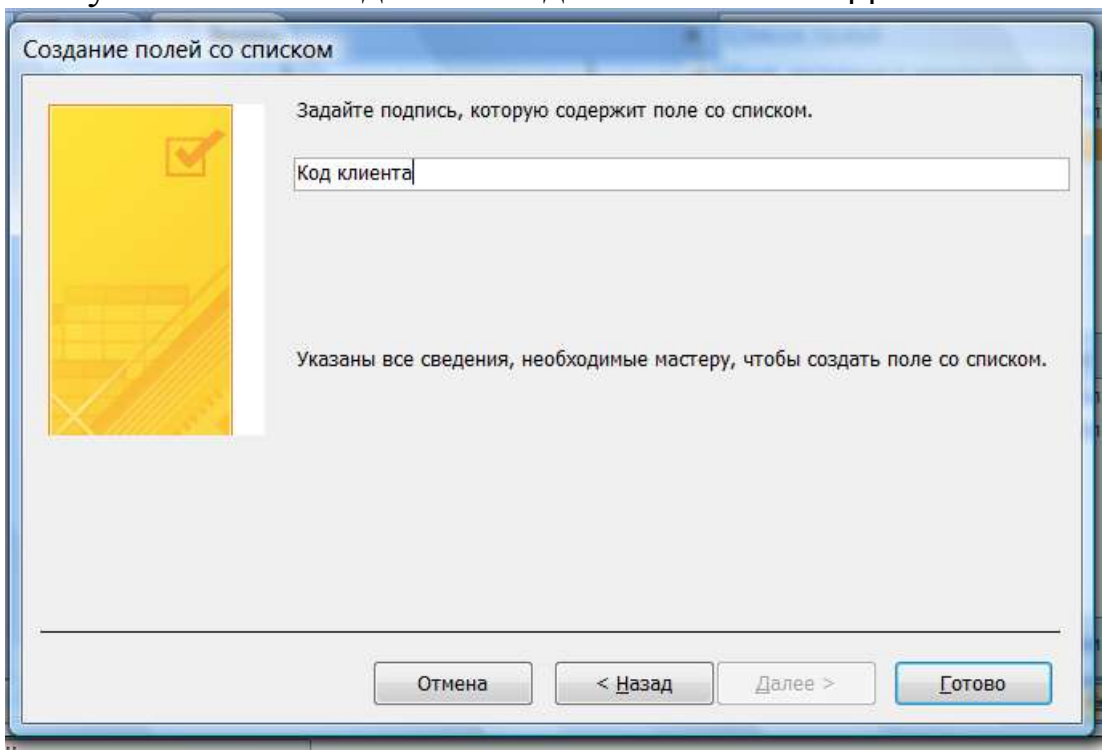


Рис.30. Ввод подписи для поля со списком

Нажимаем кнопку Готово. На форме в режиме конструктора отобразится добавленный нами элемент. Обычно требуется немного подвинуть вставленный список, чтобы была видна относящаяся к нему надпись.

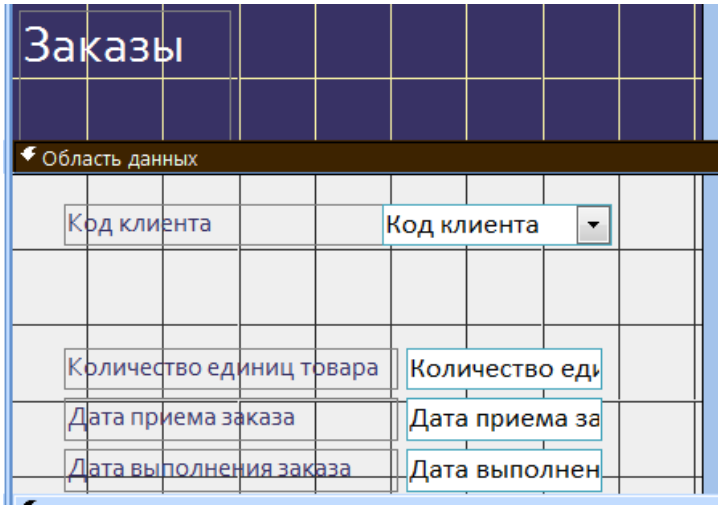


Рис.31. Готовая форма в режиме конструктора

По описанной технологии добавим на форму ещё одно поле со списком, связанное с полем Код товара из таблицы Товары.

Закрываем окно Конструктора. Двойным щелчком по значку формы Заказы открываем форму для работы с данными.

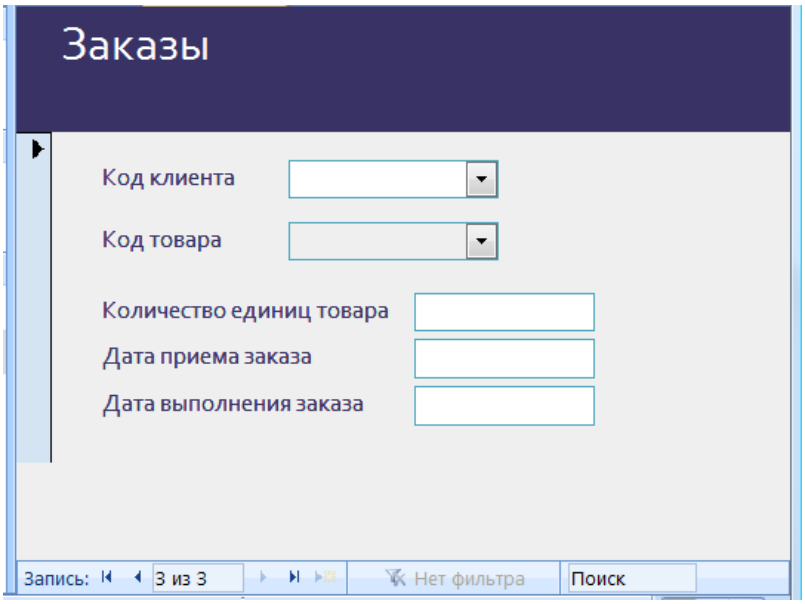


Рис. 32. Таблица Заказы в режиме формы

Теперь можно при добавлении новой записи о заказе выбирать из раскрывающихся списков Код товара и Код клиента. Кроме этого, для ввода дат можно воспользоваться встроенным календарем, который вызывается при щелчке мышью по полю типа Дата.

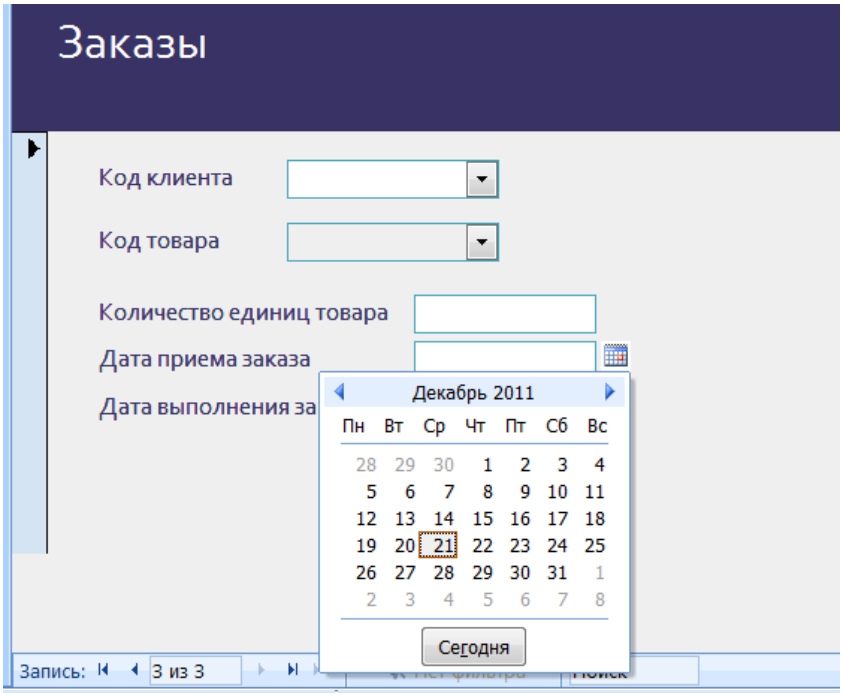


Рис.33. Ввод даты с помощью встроенного календаря

Лабораторная работа №5

РАБОТА С ЗАПРОСАМИ

Запросы являются основным рабочим инструментом базы данных и могут выполнять множество различных функций. Самая распространенная функция запросов — извлечение определенных данных из таблиц. Данные, которые необходимо просмотреть, как правило, находятся в нескольких таблицах; запросы позволяют представить их в одной таблице. Кроме того, поскольку обычно не требуется просматривать все записи сразу, с помощью запросов можно, задав ряд условий, «отфильтровать» только нужные записи. Часто запросы служат источником записей для форм и отчетов.

Некоторые запросы предусматривают возможность обновления: это означает, что данные в основных таблицах можно изменять через таблицу запроса. Изменения фактически вносятся не только в таблицу запросов, но и в соответствующие таблицы базы данных.

Существует ***два основных вида запросов***: запросы на выборку и запросы на изменение.

Запрос на выборку просто извлекает данные и дает возможность пользоваться ими. Результаты такого запроса можно просмотреть на экране, распечатать или скопировать в буфер обмена. Кроме того, их можно использовать в качестве источника записей для формы или отчета.

Запрос на изменение выполняет действия с данными. Запросы на изменение можно использовать для создания новых таблиц, добавления данных в существующие таблицы, обновления или удаления данных.

Задание

Спроектировать и реализовать 9 запросов к своей базе данных. ***Запросы по сложности должны быть аналогичными тем, которые приведены в качестве примеров в данных методических указаниях. Организовать поиск во всех таблицах вашей базы данных.***

Оформить в тетради отчет по данной лабораторной работе. Отчет должен содержать для каждого запроса 1) формулировку запроса на естественном языке, 2) зарисованный с экрана заполненный бланк запроса.

Технология работы

Создание запроса на выборку на основе одной таблицы

Пример 1 .Клиента фирмы интересует, какие товары 48-го размера можно заказать. Тогда запрос на выборку можно сформулировать так **«Вывести наименования товаров 48-го размера».**

Построения запроса с помощью конструктора

На вкладке Создание в группе Другие выбираем Конструктор запросов.

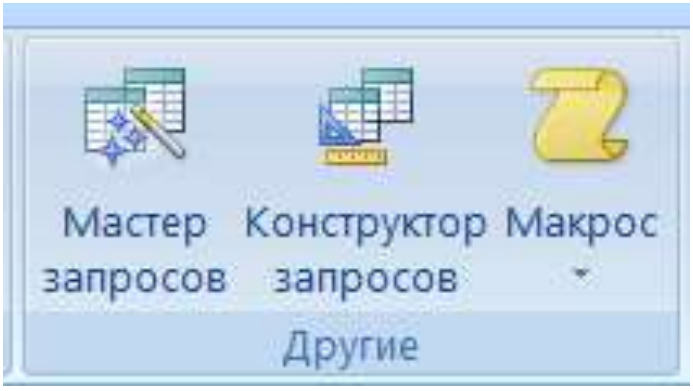


Рис. 34. Вызов Конструктора запросов

На экране появляется диалоговое окно Добавление таблицы. Информация о товарах хранится в таблице Товары, поэтому выбираем таблицу и нажимаем кнопку Добавить, а затем – кнопку Закрыть.

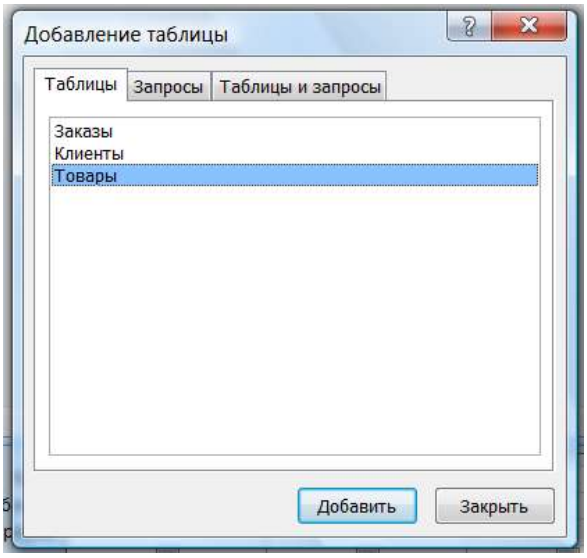


Рис.35. Добавление таблицы в запрос

Открывается окно конструктора запроса. Нижняя часть окна -бланк запроса – содержит описание запроса в табличной форме.

В первой колонке бланка запроса в строке Поле из раскрывающегося списка выбираем поле Размер. Нам надо выбрать только те записи, в которых значение поля Размер равно **48**. Для этого надо задать соответствующее условие отбора.

Условие отбора — это правило, определяющее, какие записи требуется включить в результаты запроса.

В строке Условие отбора вводим значение ***48***. В таблице ТОВАРЫ содержатся разные данные о товарах. Нас интересуют только наименования товаров, поэтому во второй колонке в строке Поле из раскрывающего списка выбираем поле Наименование

В результате наших действий бланк запроса принимает вид, показанный на рис.36.

.

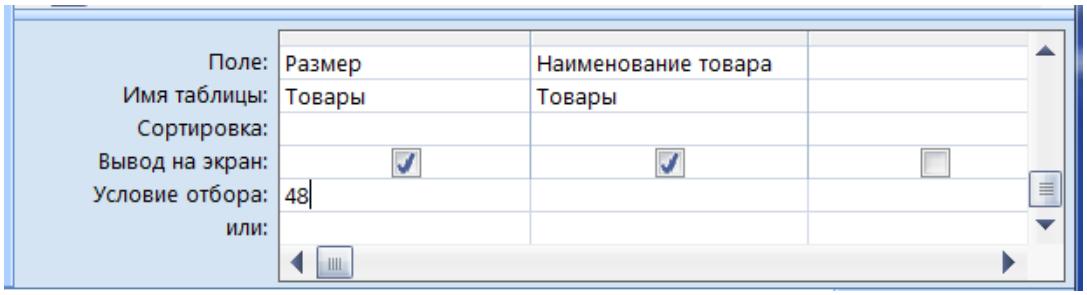


Рис.36. Бланк запроса для примера 1

Просмотр результатов выполнения запроса

Просмотреть результаты выборки данных можно, выбрав пункт Выполнить (изображение большого красного восклицательного знака) в группе Результаты.

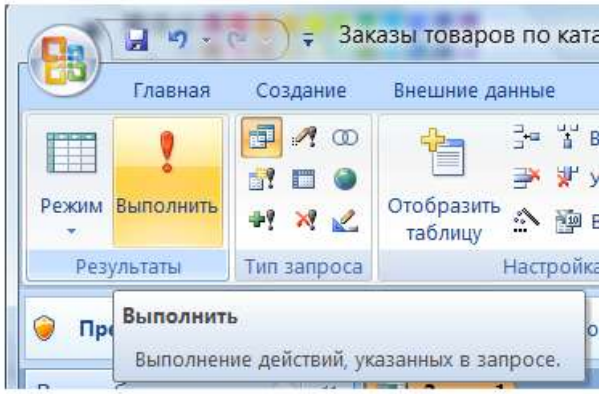


Рис.37. Команда Выполнить запрос

Если в таблице Товары есть интересующие нас данные, то они будут выведены в виде таблицы, а если нет – результатом выполнения запроса будет пустая таблица.

Сохранение запроса

Запрос можно использовать многократно, поэтому лучше его сохранить. Закрываем окно с результатами выборки и отвечаем утвердительно на вопрос о сохранении запроса. В поле «имя» надо заменить предлагаемое системой имя Запрос1 на любое другое, например, Товары 48-го размера.

Пример 2. ***Вывести наименования товаров, цены на которые не превышают 1000 руб.***

Интересующая нас информация содержится в той же таблице Товары, но теперь условие отбора надо связать с полем Цена. Следуя описанной в первом примере технологии, создаем с помощью конструктора бланк запроса следующего вида.

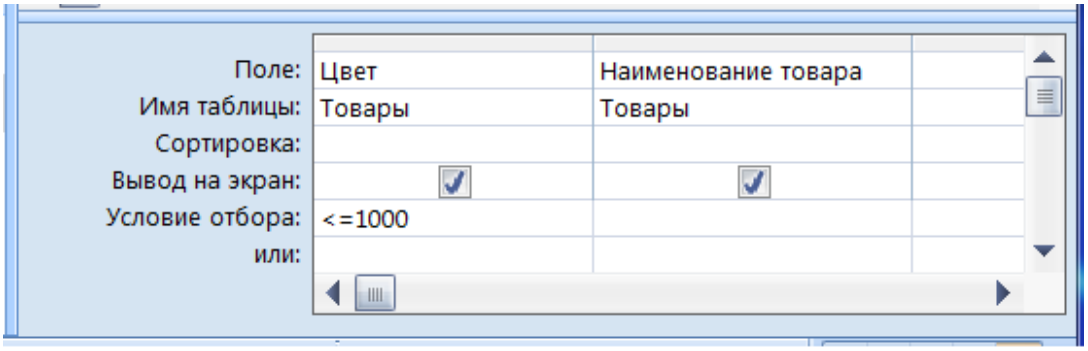


Рис.38. Бланк запроса для примера 2

В первом примере было использовано простейшее условие отбора – совпадение значения в указанном поле числового типа с заданной константой. Во втором примере использовано условие с использованием операции сравнения. Программа Access позволяет задавать разнообразные условия отбора с использованием символов операций сравнения **<, <= , > , >= .**

Примеры записи возможных условий отбора для полей числового и денежного типа приведены в Таблице 1.

Таблица 1

Примеры записи условий отбора для полей типа Числовой и Денежный

Записи	Условие	Результат запроса
Точно соответствуют определенному значению, например 1500	1500	Записи, в которых цена единицы товара составляет 1500 рублей.
Не соответствуют значению, например, 1000	Not 1000	Записи, в которых цена единицы товара не равна 1000 рублей.
Содержат значение, которое меньше заданного, например 2000	< 2000 <= 2000	Записи, в которых указана цена единицы меньше 2000 рублей (<2000). Второе выражение (<=2000) позволяет отобразить записи, в которых цена единицы меньше либо равна 2000.
Содержат значение, которое больше заданного, например 990,99	>990,99 >=990,99	Записи, в которых указана цена единицы больше 990,99 рублей (>990,99). Второе выражение позволяет отобразить записи, в которых цена единицы больше либо равна 990,99.
Содержат значение, которое входит в определенный диапазон	>490,99 and <990,99 -или- Between 490,99 and 990,99	Записи, в которых указана цена единицы в диапазоне между 490,99 и 990,99 рублей (сами эти значения не включаются в результаты).

Пример 3. ***Вывести коды и адреса клиентов, фамилии которых, начинаются на букву «А».***

Нужная нам информация содержится в таблице Клиенты, а условие отбора необходимо связать с полем Фамилия. В первых двух запросах выборка данных производилась в полях числового типа. Теперь нам надо организовать выборку из поля

текстового типа. Примеры записи условий для текстовых полей приведены в таблице 2. Бланк запроса для третьего примера показан на рисунке 39.

Таблица 2

Примеры условий отбора для полей типа Текстовый

Записи	Условие	Результат запроса
Точно соответствуют определенному	"Иванов"	Записи, в которых поле «Фамилия» содержит значение «Иванов».
Начинаются с заданной строки символов	Like С*	Записи, в которых фамилия клиента начинается с буквы «С», например, Семёнов, Савин и т.д. Используемая в выражении звездочка (*) является подстановочным знаком и заменяет любое количество любых символов. .
Заканчиваются заданной строкой,	Like "*ина"	Записи, в которых фамилии заканчиваются на «ина», например «Савина» или «Кирилина».
Входят в определенный диапазон,	Like "[А-Г]*"	Записи, в которых фамилия начинается с одной из букв от «А» до «Г».

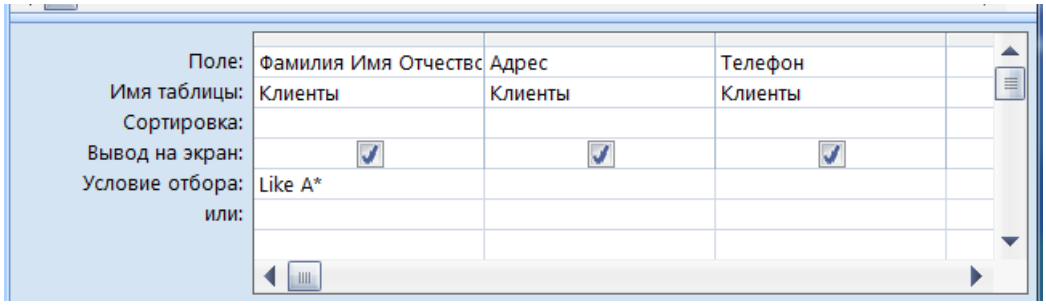


Рис.39. Бланк запроса для примера 3

Пример 4. ***Вывести коды товаров, заказы на которые должны быть выполнены в марте 2012 года.***

Условие отбора для поля Дата выполнения заказа включает диапазон значений дат от 01.03.07 до 31.03.07. Диапазон значений задается с помощью оператора BETWEEN (между). Признаком константы типа дата является разделитель " # ". Бланк запроса имеет следующий вид:

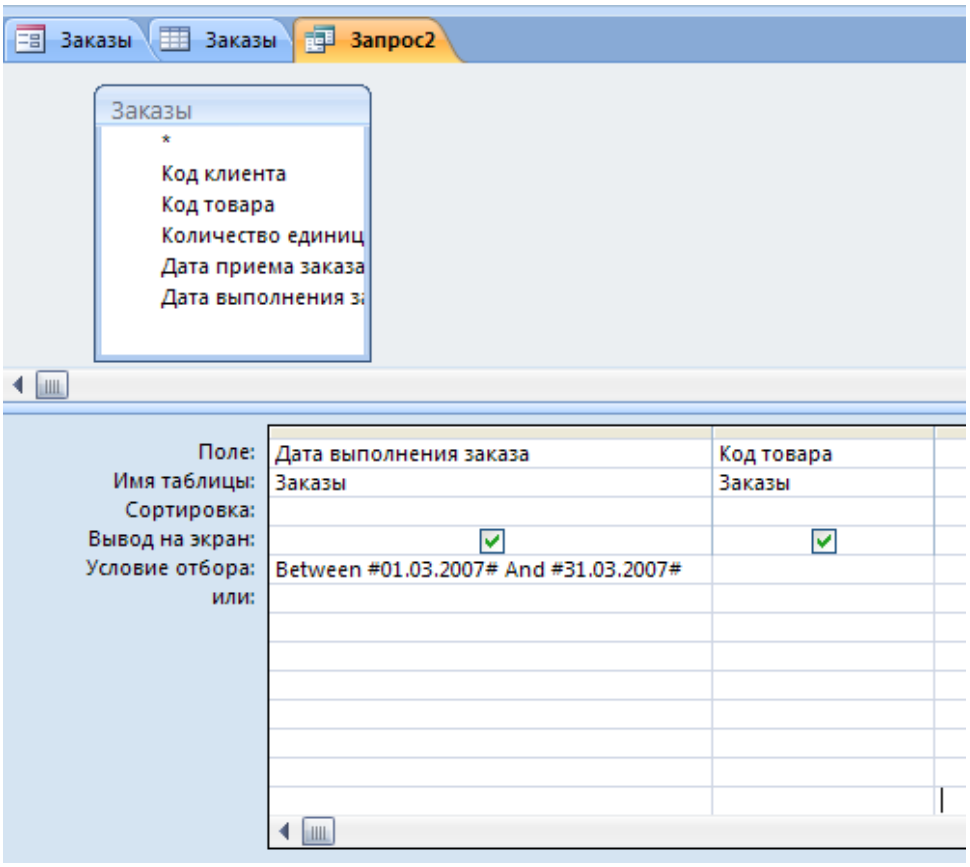


Рис.40. Бланк запроса для примера 4

Таблица 3

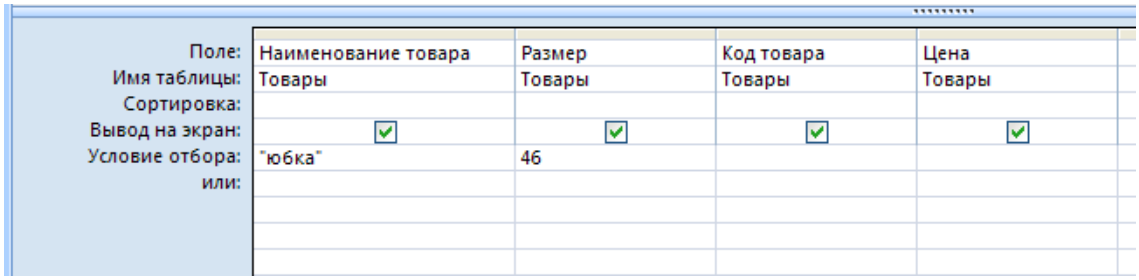
Примеры условий для полей типа «Дата/Время»

Записи	Условие	Результат запроса
Точно соответствуют определенному значению, например 02.02.2006	#02.02.2006#	Значения даты должны быть окружены знаками #, чтобы Access мог отличить значения даты от текстовых строк.
Содержат значения, которые предшествуют определенной дате, например 02.02.2006	< #02.02.2006#	Записи об операциях, совершенных до 2 февраля 2006 г.
Содержат значения, которые следуют за определенной датой, например 02.02.2006	> #02.02.2006#	Записи об операциях, совершенных после 2 февраля 2006 г.
Содержат значения, которые входят в определенный диапазон дат	>#02.02.2006# and <#04.02.2006# ИЛИ Between #02.02.2006# and #04.02.2006#	Записи об операциях, совершенных между 2 и 4 февраля 2006 г.
Содержат текущую дату	Date()	Записи об операциях, совершенных на текущую дату.
Содержат вчерашнюю дату	Date()-1	Записи об операциях, совершенных за один день до текущей даты.
Содержат прошедшую дату	< Date()	Записи об операциях, совершенных до наступления текущей даты.

Поиск по значениям двух полей

Пример 5. ***Вывести коды и цены юбок 46-го размера.***

В данном примере условия должны быть заданы для двух полей: Наименование товара и Размер, а выводиться на экран должны коды и цены соответствующих товаров. Бланк запроса показан на рисунке 41.



Поле:	Наименование товара	Размер	Код товара	Цена
Имя таблицы:	Товары	Товары	Товары	Товары
Сортировка:				
Вывод на экран:	☑	☑	☑	☑
Условие отбора:	"юбка"	46		
или:				

Рис.41. Бланк запроса для примера 5.

Условия, заданные в соседних колонках бланка запроса, объединяются с использованием логической операции "И", то есть результатом выборки будут те записи, для которых все эти условия выполняются одновременно.

Поиск по двум таблицам

Пример 6. ***Вывести коды и наименования товаров, которые заказал клиент с кодом 100.***

Сведения о кодах товаров, которые заказал клиент с определенным кодом, содержатся в таблице ЗАКАЗЫ, а наименования товаров находятся в таблице ТОВАРЫ. Следовательно, для ответа на вопрос надо организовать поиск по двум таблицам.

На вкладке Создание в группе Другие выбираем Конструктор запросов.

В окне Добавление таблицы добавляем 2 таблицы ЗАКАЗЫ и ТОВАРЫ. Между этими таблицами нами была установлена связь, которая отображается в окне бланка запроса. Заполняем бланк запроса так, как показано на . рис.42:

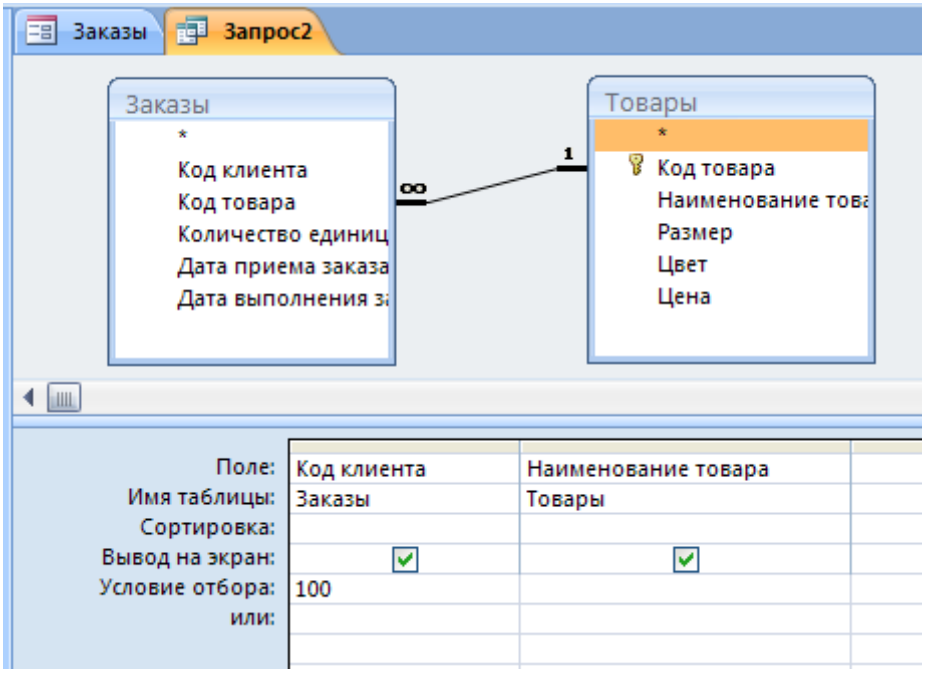


Рис.42. Бланк запроса для примера 6

Запрос с параметром

Пример 7. ***Вывести список наименований товаров, заказанных клиентом, номер которого вводится с клавиатуры.***

Запрос, в котором в условии отбора используется не константа, а введенное с клавиатуры значение, реализуется с помощью запроса с параметром. Запрос с параметром отличается от обычного запроса тем, что в поле УСЛОВИЕ ОТБОРА записывается не константа, по которой идет поиск по данному полю, а приглашение пользователю ***ввести*** значение для поиска. Это значит, что с помощью одного и того же запроса можно получать сведения о разных клиентах, вводя с клавиатуры соответствующие коды. Приглашение вводится в бланк запроса в виде текста, заключенного в квадратные скобки.

Таблицы включаются в запрос так же, как и в предыдущем примере. Бланк с приглашением для ввода показан на рисунке 43.

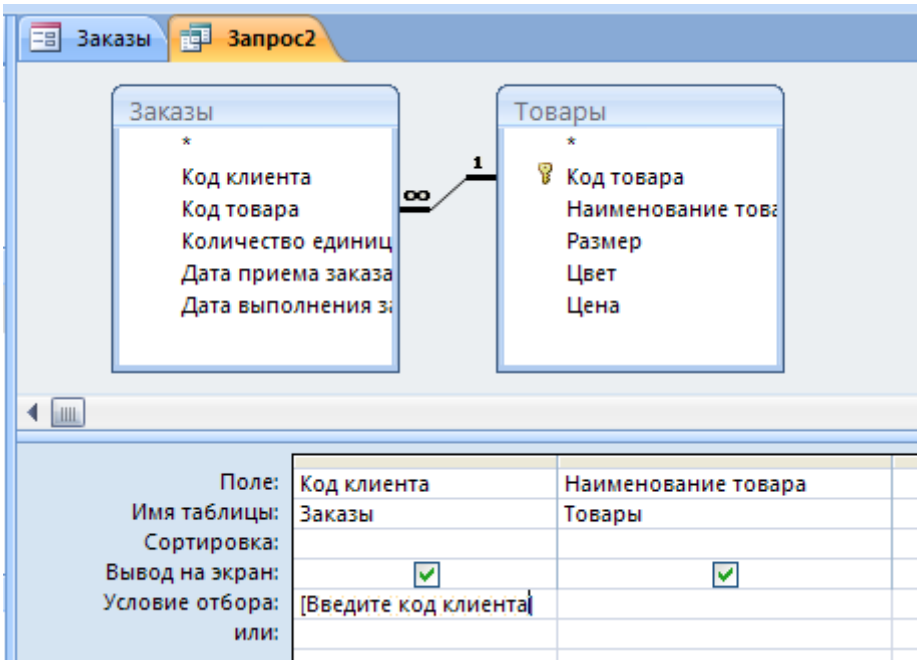


Рис.43. Бланк запроса для примера 7.

Запрос с вычислениями

Пример 8*. **Определить стоимость каждого заказа.***

В таблице ЗАКАЗЫ есть сведения о количестве единиц заказанных товаров, но нет информации о стоимости заказа. Стоимость заказа можно вычислить, умножив число единиц товара на его цену. Вычисления можно произвести с помощью запроса с вычислением. Результат вычисления помещается в вычисляемое поле, которое будет существовать только в запросе (в базовой таблице никаких изменений не происходит).

На вкладке Создание в группе Другие выбираем Конструктор запросов.

В окне Добавление таблицы добавляем 2 таблицы ЗАКАЗЫ и ТОВАРЫ. Между этими таблицами нами была установлена связь, которая отображается в окне бланка запроса.

В первых 3-х колонках бланка запроса устанавливаем имена полей Код клиента, Код товара, Количество единиц товара.

В четвертую колонку банка вводим формулу

Стоимость заказа: [Количество единиц товара]*[Цена].

Получаем бланк как на рисунке 44.

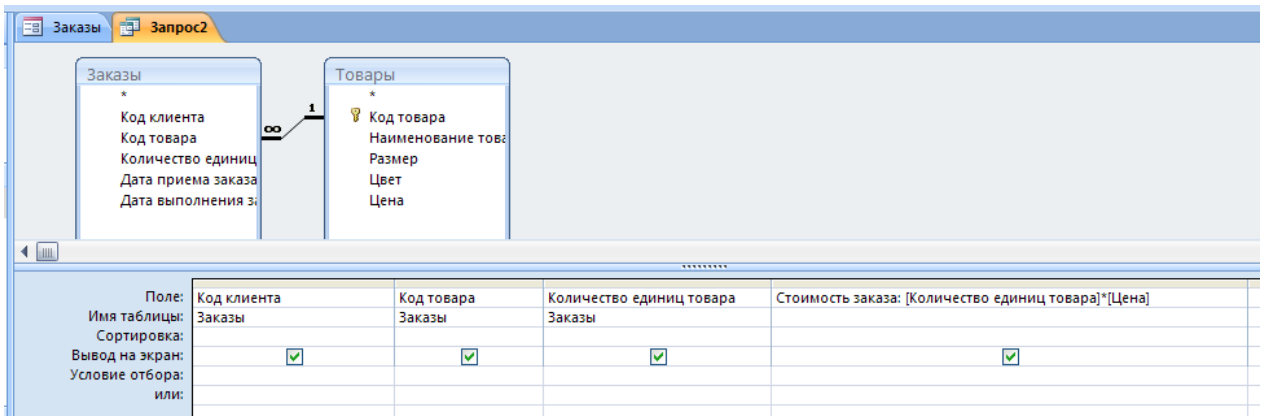


Рис.44. Бланк запроса для примера 8

Итоговый запрос

Пример 9. ***Определить суммарную стоимость заказов клиента с кодом 110.***

Перед нами стоит задача: найти суммарную стоимость заказов конкретного клиента. Сведения о стоимости заказов содержатся не в базовой таблице, а в ***запросе*** Стоимость заказов. Поэтому входим в режим создания запроса с помощью конструктора и в окне «Добавление таблицы» выбираем вкладку Запросы. Из появившегося списка запросов выбираем Стоимость заказов и закрываем данное окно.

В первой колонке бланка запроса в строке Поле из списка полей выбираем Код клиента.

В строке Условие отбора для данного поля вводим код клиента 110.

Во второй колонке бланка запроса в строке Поле из списка полей выбираем Стоимость заказа.

Просуммировать значения в поле Стоимость заказа можно путем добавления в запрос ***строки итогов***. Для этого сначала надо выполнить запрос (нажав кнопку с изображением красного восклицательного знака). Запрос становится доступным в режиме таблицы.

На вкладке Начальная страница в группе Записи выберите команду Итоги.

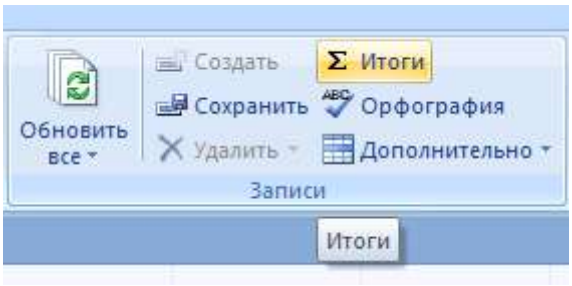


Рис.45. Команда Итоги.

В таблице появится новая строка Итог.

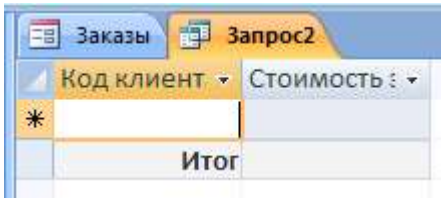


Рис.46. Добавление строки Итог в бланк запроса

В строке Итог щелкните ячейку в поле, по которому необходимо провести суммирование, и выберите в списке функцию Сумма.

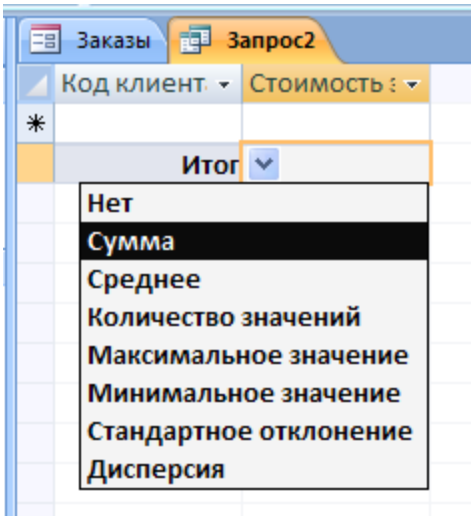


Рис.47. Выбор функции Сумма

3. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Лабораторная работа 6

РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММ ЛИНЕЙНОЙ СТРУКТУРЫ

ЗАДАНИЕ

1. Внимательно прочитайте «Краткие теоретические сведения».
2. Запишите в тетрадь ответы на следующие вопросы:
   1) Что такое алгоритм?
   2) Как можно представить алгоритм?
   3) Перечислите основные структуры алгоритмов.
   4) Как схематически изображается структура алгоритмов Последовательность?
   5) Что такое линейная программа?
2. Выполните на компьютере Пример1.
3. Запишите и зарисуйте в тетради условие задачи, структуру данных, алгоритм, эскиз формы и текст программы из Примера 1

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

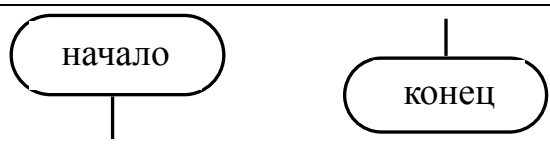
Основные структуры алгоритмов

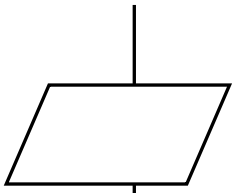
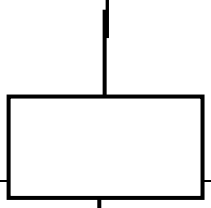
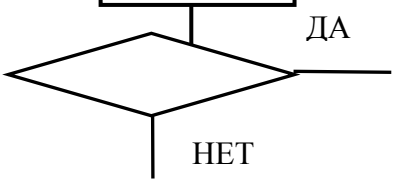
Алгоритм-это план решения задачи, в котором четко определена последовательность действий, которая приводит к заданному результату.

Алгоритм любой сложности можно представить как комбинацию ***трёх основных структур алгоритмов:*** последовательность, разветвление и цикл.

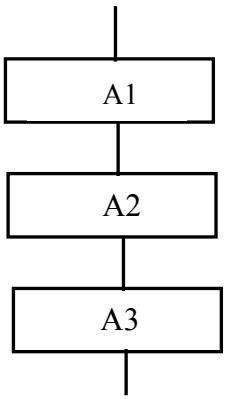
Существуют 2 способа представления алгоритмов: словесное описание и графическое представление в виде блок-схемы.

Основные обозначения на блок-схемах алгоритмов.

	начало и завершение алгоритма

	Ввод и вывод данных
	Обработка данных
	Проверка условия

Структура алгоритмов ПОСЛЕДОВАТЕЛЬНОСТЬ



В данной структуре блоки выполняются последовательно, один за другим. Блоки располагаются вдоль одной вертикальной линии, поэтому алгоритм, состоящий из структур типа последовательность, называются ***линейным.***

Программа, реализующая линейный

Пример 1: **Составить программу для вычисления суммы двух целых чисел, значения которых вводятся с клавиатуры.**

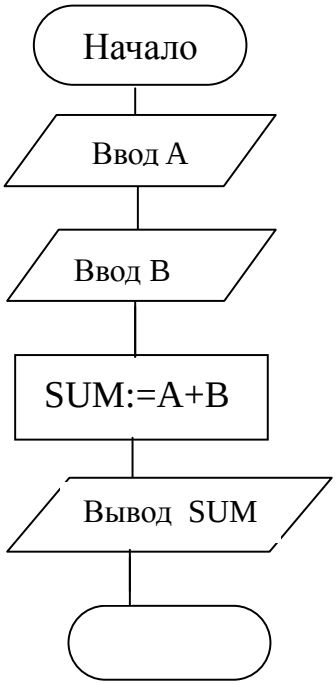
1. Определим структуру данных для этой задачи. ***Структура данных*** – описание состава входной и выходной информации.

Структура данных

Вход	Выход
A,B-числовые переменные целого	SUM - числовая переменная целого

типа	типа

2. Блок-схема алгоритма



3. Проектирование формы

Форма – это прообраз окна будущей программы. Компоненты, размещаемые на форме, реализуют взаимодействие программы с пользователем .при вводе и выводе информации (программный интерфейс).

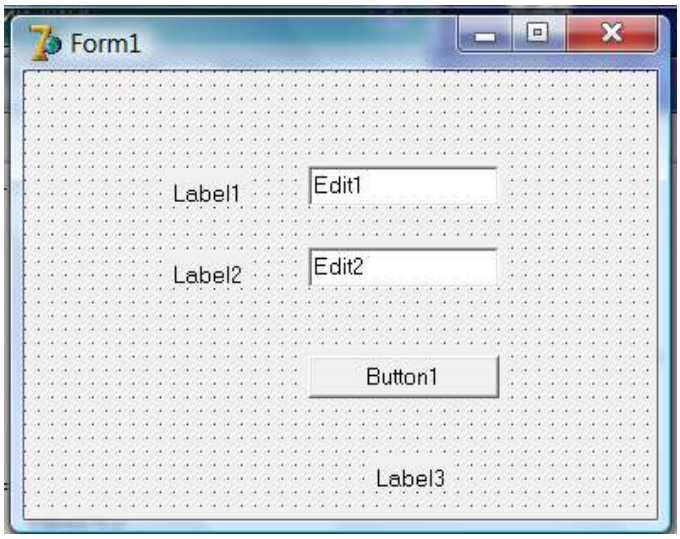
Вввод данных с клавиатуры.

Для ввода данных с клавиатуры в Delphi применяется компонент **Edit** (окно редактирования). В нашем примере надо ввести значения двух переменных A и B, следовательно, на форме должно быть 2 компонента **Edit1 и Edit2.**

Вывод данных

Для вывода результатов и поясняющих надписей используется компонент **(**метка) .Свойство компонента **Label,**отвечающие за содержание надписи, называется **Caption** (заголовок). В данном примере нам потребуется разместить на форме 3 компонента **Label** – один для вывода результата и 2 – для вывода поясняющих надписей.

4. Размещение компонентов на форме



5. Задание свойств компонентов с помощью Object Inspector

Имя компонента	***Свойство компонента***	***Значение свойства***
Label1	Caption	A=
Label2	Caption	B=
Label3	Caption	Пустая строка
Edit1	Text	Пустая строка
Edit2	Text	Пустая строка
Button1	Caption	Суммировать

6. Ввод текста процедуры для обработки события нажатия командной кнопки Button1.

Выполните двойной щелчок мышкой по изображению на форме командной кнопки Button1. Вы увидите открывшееся окно редактора кода с заготовкой для обработчика события нажатия клавиши (Click):

```
procedure TForm1.Button1Click(Sender: TObject);
begin

end;
```

Установите курсор в начало строки со словом ***begin*** и нажмите клавишу ***Enter***. В образовавшуюся пустую строку введите объявление переменных

```
Var     {раздел объявления переменных}

  A,B,SUM:integer
```

После слова **begin** введите текст**:**

```
  A:=Strtoint(Edit1.TEXT);  {Ввод значения переменной A}
  B:=Strtoint(Edit2.TEXT);  { Ввод значения переменной B}
  SUM:=A+B;
  Label3.Caption:=inttostr(SUM);{ Вывод значения переменной SUM}
```

В итоге должен получиться следующий текст процедуры:

```
procedure TForm2.Button1Click(Sender: TObject);
var
  A,B,SUM:integer;
begin
  A:=Strtoint(Edit1.TEXT);{Ввод значения переменной A}
```

```
  B:=Strtoint(Edit2.TEXT);{ Ввод значения переменной B}
  SUM:=A+B;
  Label3.Caption:=inttostr(SUM);{ Вывод значения переменной SUM}
end;
```

Пояснения к тексту программы

Объявление переменных

Все переменные, которые используются в программе, должны быть объявлены в разделе объявления переменных. В данном примере все переменные – целочисленные (integer).

Как происходит запись введённого с клавиатуры значения в переменную?

Пользователь вводит с клавиатуры число (например, значение A) в компонент **Edit1.** Введённое значение становится значением свойства **Text** компонента **Edit1. С**войство **Text** компонента **Edit1** содержит данные **в** виде строки символов (тип **string**) .

Так как для решения задачи требуется выполнить сложение **чисел** ,то необходимо ***преобразовать*** введённое значение из строки символов в целое число. Для этого используется функция

StrtoInt

(String to Integer conversion)

Вывод на экран результата вычислений с использованием компонента Label.

Чтобы вывести на экран вычисленное значение переменной **Sum**, надо присвоить это значение свойству **Caption** компонентf **Label3.** Результат получается в виде значения целого числа (значение переменной Sum) . Свойство **Caption** содержит строку символов , поэтому с помощью функции **Inttostr** происходит преобразование целого числа в строку:

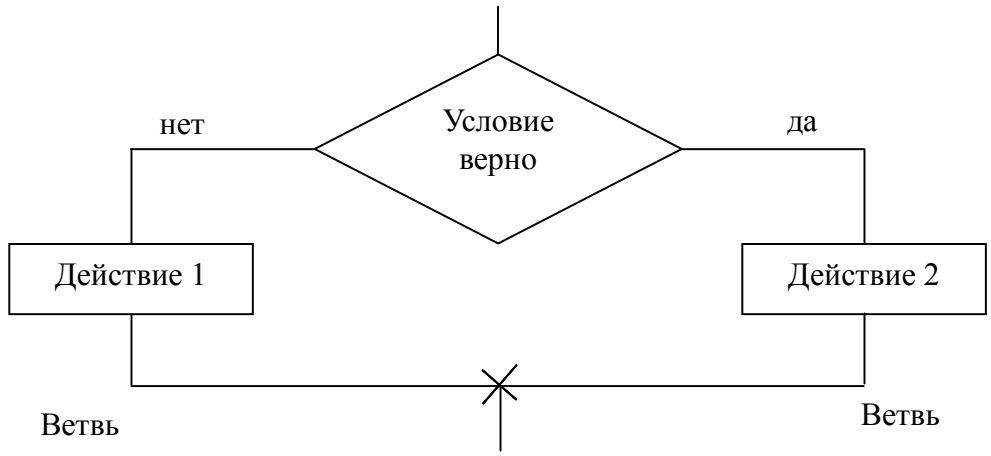
IttoStr

(Integer to String conversion)

ЛАБОРАТОРНАЯ РАБОТА №7
РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММ С РАЗВЕТВЛЕНИЯМИ

Краткие теоретические сведения

Разветвление используется для проверки условия и выбора дальнейших действий в программе и имеет следующую структуру:



Команда, с помощью которой проверяется условие и организуется разветвление в программе, называется командой условного перехода.

if условие ***then*** оператор 1 ***else*** оператор 2

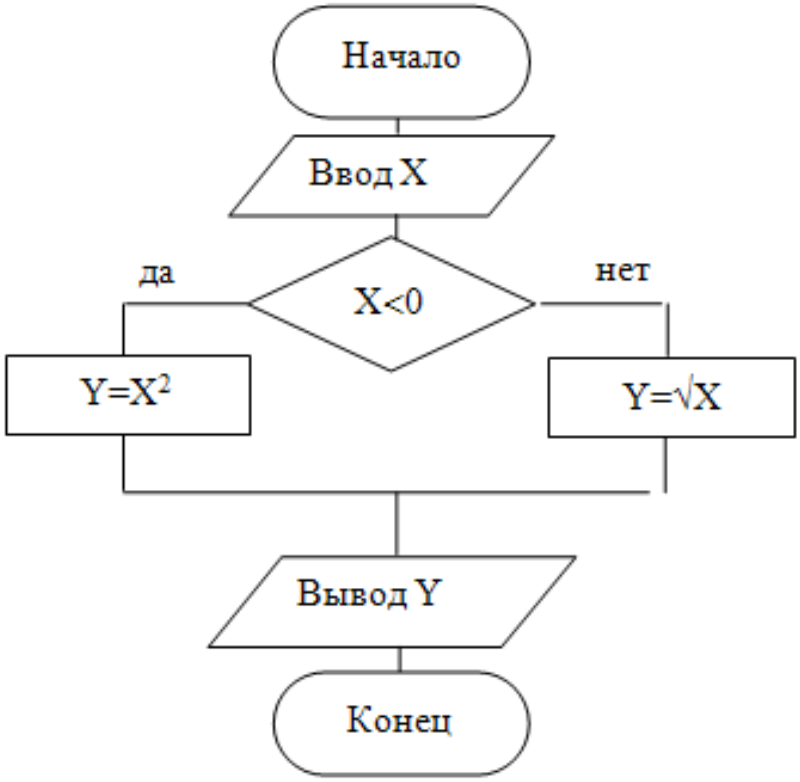
Пример 1*:*

Дано значение X. Вычислить значение Y

$$Y = \begin{cases} x^2 & \text{, если } x<0 \\ \sqrt{x} & \text{, в остальных случаях} \end{cases}$$

.

Блок-схема алгоритма



Текст процедуры обработчика события нажатия командной кнопки для примера 1

```
Procedure TForm1.Button1Click(Sender:
TObject);

var
x:integer;
y:real;
begin
x:=strtoint(Edit1.Text);
if x<0 then y:=SQR(x) else y:=SQRT(x);
Label3.Caption:=floattostr(y);
end;
```

ЗАДАНИЕ 1

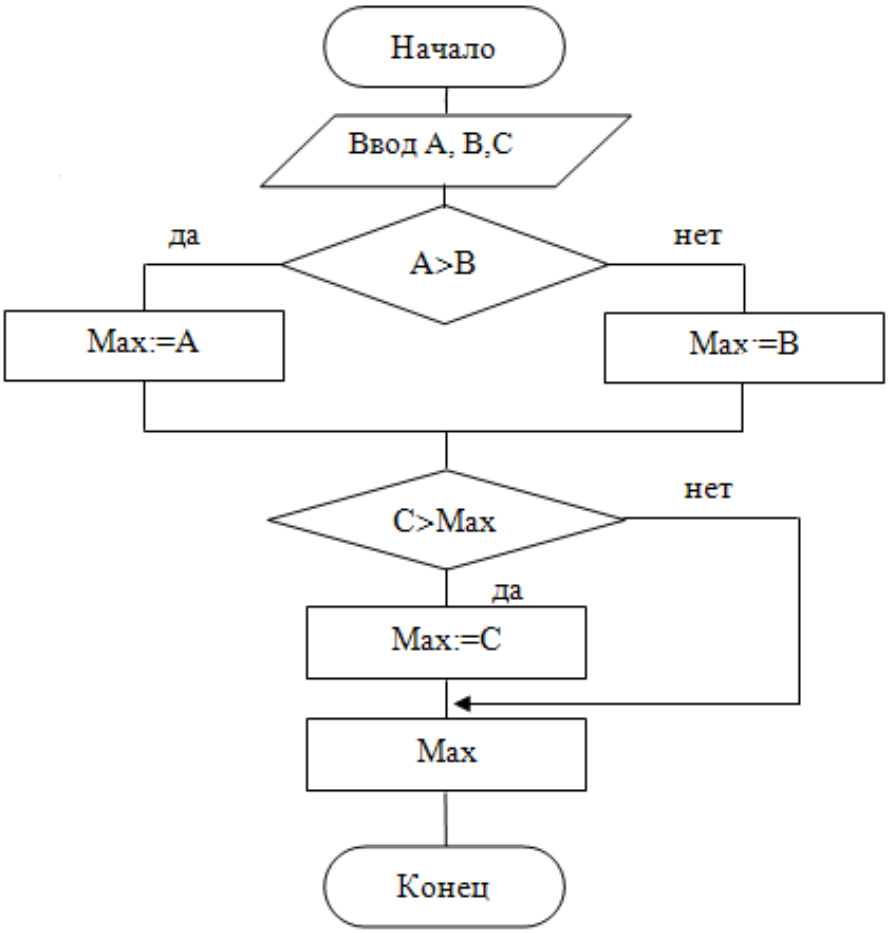
1. Разработать структуру данных для примера 1.
2. Спроектировать форму для примера 1.

3. Реализовать программу на компьютере**.**

Пример 2.

Даны три числа. Найти наибольшее среди них.

Для решения этой задачи используется стандартный алгоритм поиска максимума из трех чисел:



Запись на Delphi основной части алгоритма (без организации ввода и вывода данных и результата)

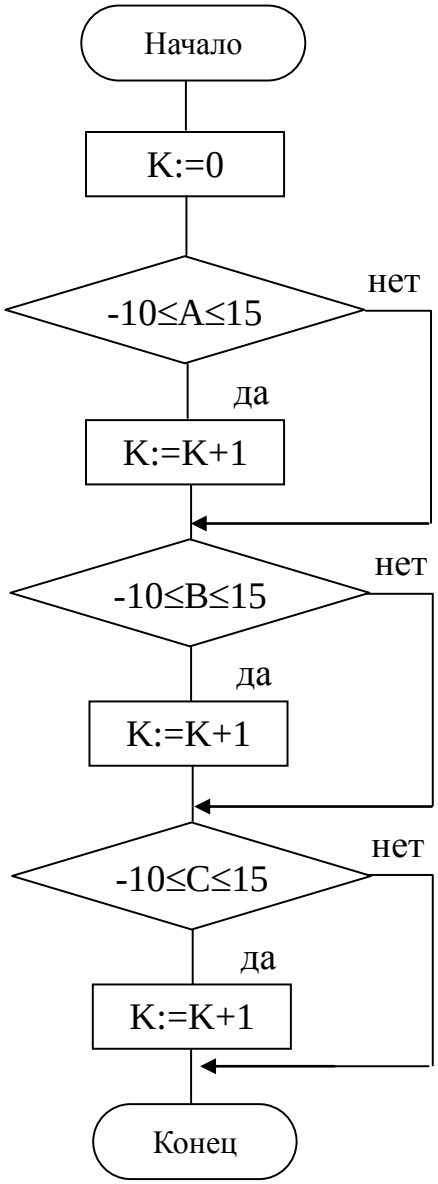
```
if A > B then max:=A else max:=B;
If C > max then max:=C;
```

ЗАДАНИЕ 2

Разработать структуру данных для примера 2.
Спроектировать форму для примера 2.
Реализовать программу на компьютере.

Пример 3

Даны 3 числа. Определить, сколько из этих чисел принадлежат интервалу [-10; 15].



Задача решается с использованием изображенного на рисунке стандартного алгоритма подсчета количества значений, удовлетворяющих заданному условию.

Запись на Delphi основной части алгоритма (без организации ввода и вывода данных и результата)

```
If (A>=-10) and (A<=15) then K:=K+1;
If (B>=-10) and (B<=15) then K:= K+1;
If (C>=-10) and (C<=15) then K:= K+1;
```

ЗАДАНИЕ 3

1. Разработать структуру данных для примера 3.
2. Спроектировать форму для примера 3.
3. Реализовать программу на компьютере.

ЛАБОРАТОРНАЯ РАБОТА №8

ИЗУЧЕНИЕ СТАНДАРТНЫХ АЛГОРИТМОВ ДЛЯ РАБОТЫ С МАССИВАМИ

Большинство задач, связанных с обработкой массивов, решаются с использованием трех основных стандартных алгоритмов:

1. Алгоритм постепенного накопления суммы элементов массива.
2. Алгоритм подсчёта количества элементов, удовлетворяющих заданному условию.
3. Алгоритм поиска минимального ил максимального элементов массива.

1. Алгоритм постепенного накопления суммы элементов массива

Общая формулировка задачи.
Дан числовой массив . Вычислить сумму элементов массива.

Пример

На метеостанции ведется ежедневное измерение количества выпавших осадков. Определить суммарное количество осадков, выпавших в течение марта.

Решение

В данной задаче исходные данные – это значения 31 числа (кол-во осадков за каждый из 31-го дня марта). В подобных случаях для представления исходных данных используются не отдельные переменные, а массивы. ***Массив*** – это множество соседних ячеек памяти, имеющих общее имя.

Пусть в нашей задаче исходные данные будут представляться массивом целых чисел с именем A. Тогда количеству осадков за 1-й, 2-й, .., i-тый день марта будут соответствовать элементы массива A[1], A[2], …,A[i]. Нам надо вычислить сумму элементов массива A. Для этого воспользуемся стандартным алгоритмом постепенного

накопления суммы элементов массива, блок-схема которого показана на рис.1.

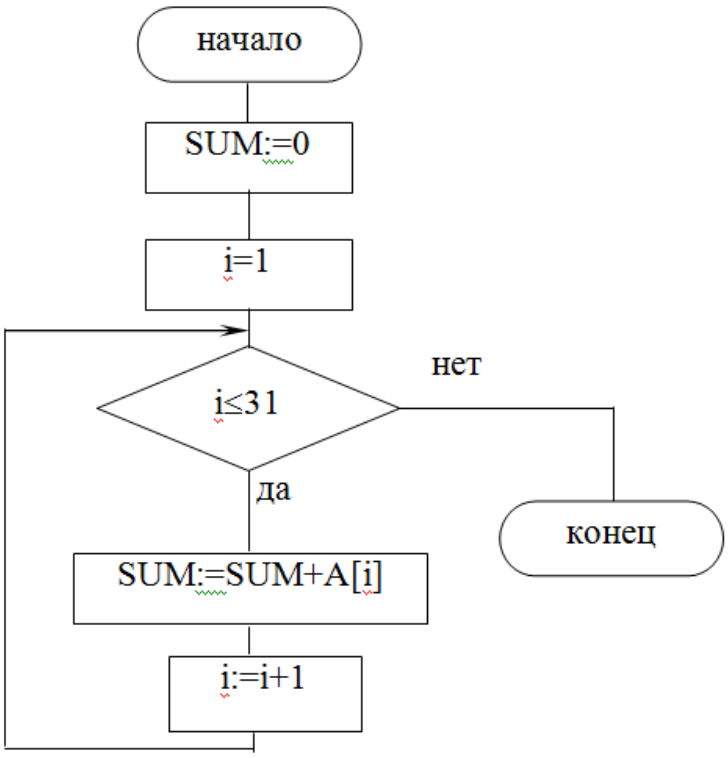


Рис.1. Алгоритм постепенного накопления суммы элементов массива

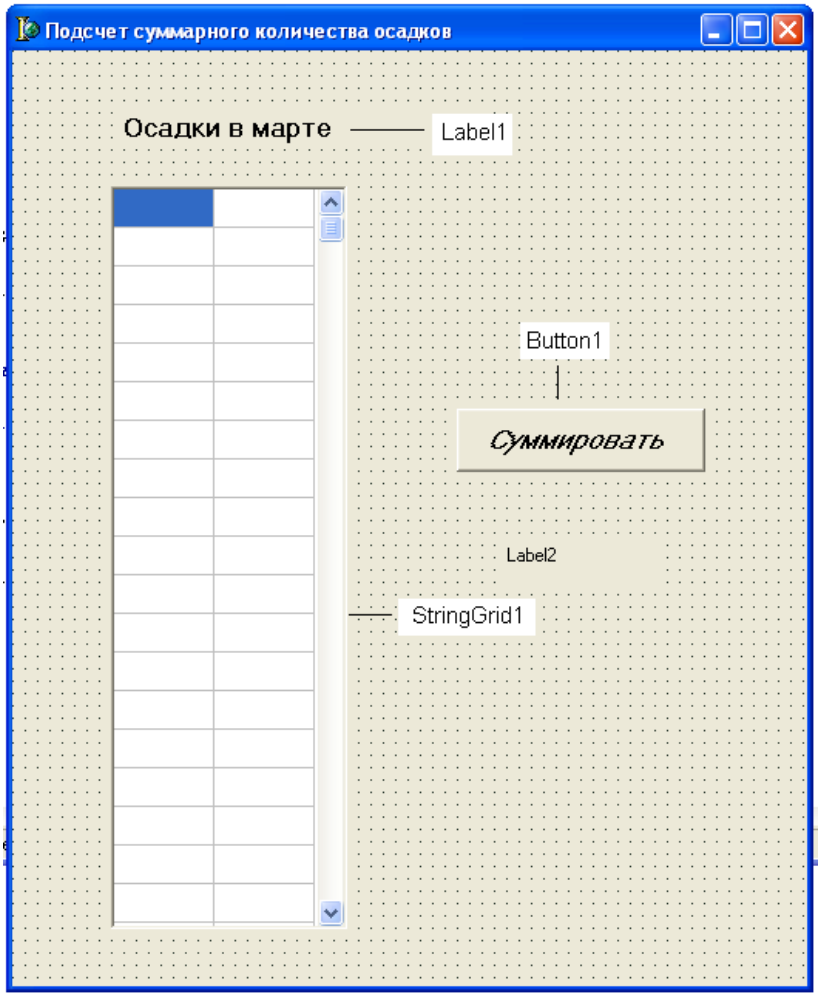
ПРОЕКТИРОВАНИЕ ФОРМЫ

В данной задаче с точки зрения человека исходные данные естественным образом представляются в виде таблицы:

День месяца	*Кол-во выпавших осадков (мм)*
1	10
2	0
….	…
31	7

Для работы с данными в табличной форме в Delphi используется специальный компонент StringGrid .

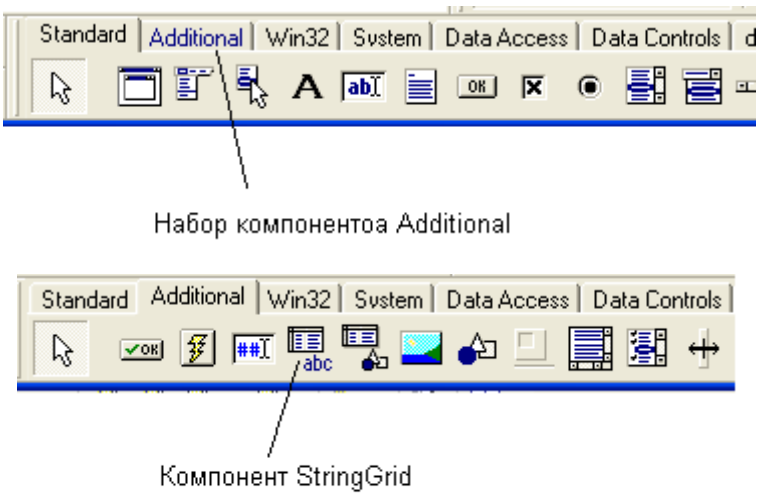
На рис.2 показан эскиз формы для решения задачи. Создайте такую форму, следуя описанной ниже технологии.



Технология работы с компонентом StringGrid

1. Как поместить компонент StringGrid на форму

Чтобы поместить на форме компонент StringGrid (Сетка), надо выбрать в палитре компонентов вкладку Additional (дополнительные).



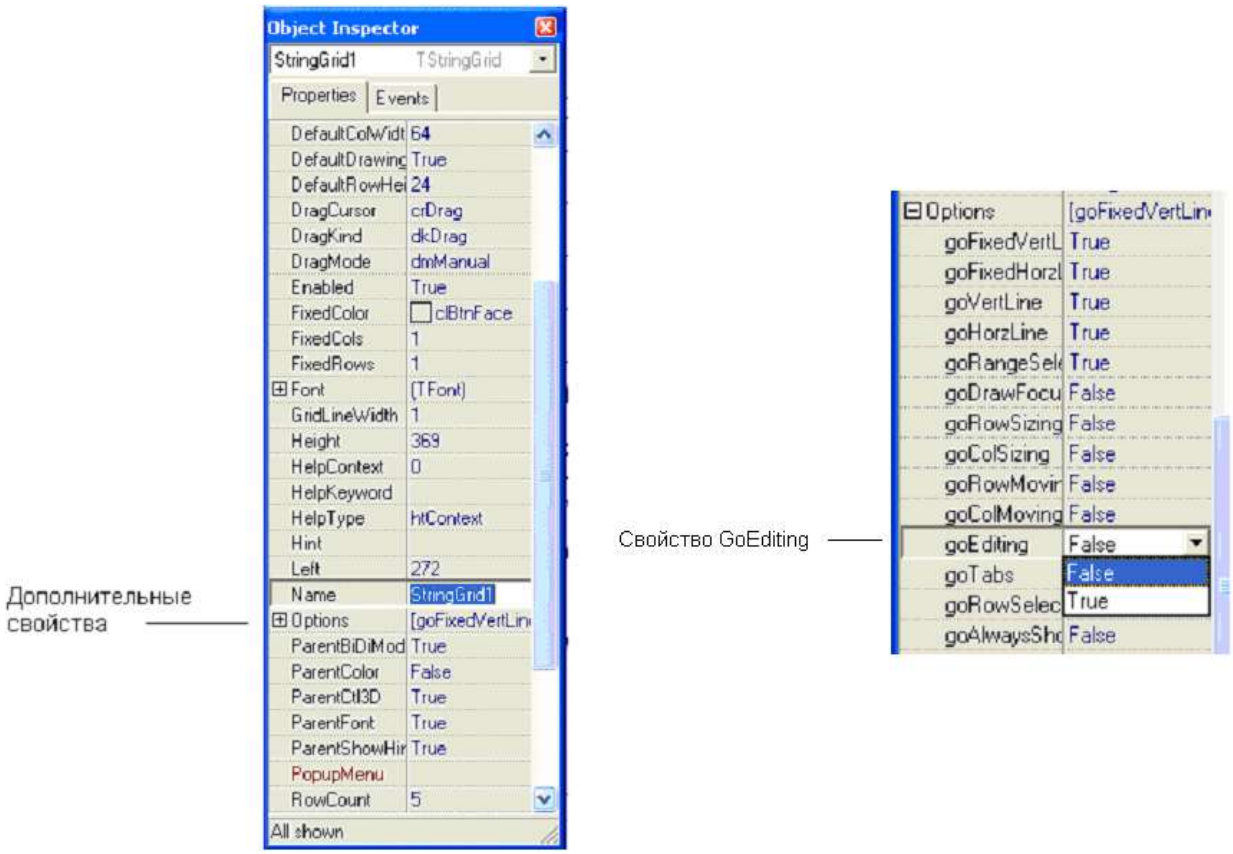
2. Как задать число строк и столбцов StringGrid

Сделать компонент активным, щелкнув по нему мышью, и установить следующие значения перечисленных свойств:

	Свойство (Property)	**Значение свойства**
Число столбцов	ColCount	2
	RowCount	32
Число фиксированных столбцов	FixedCols	0
Число фиксированных строк	FixedRows	0

3. Как разрешить ввод данных в StringGrid

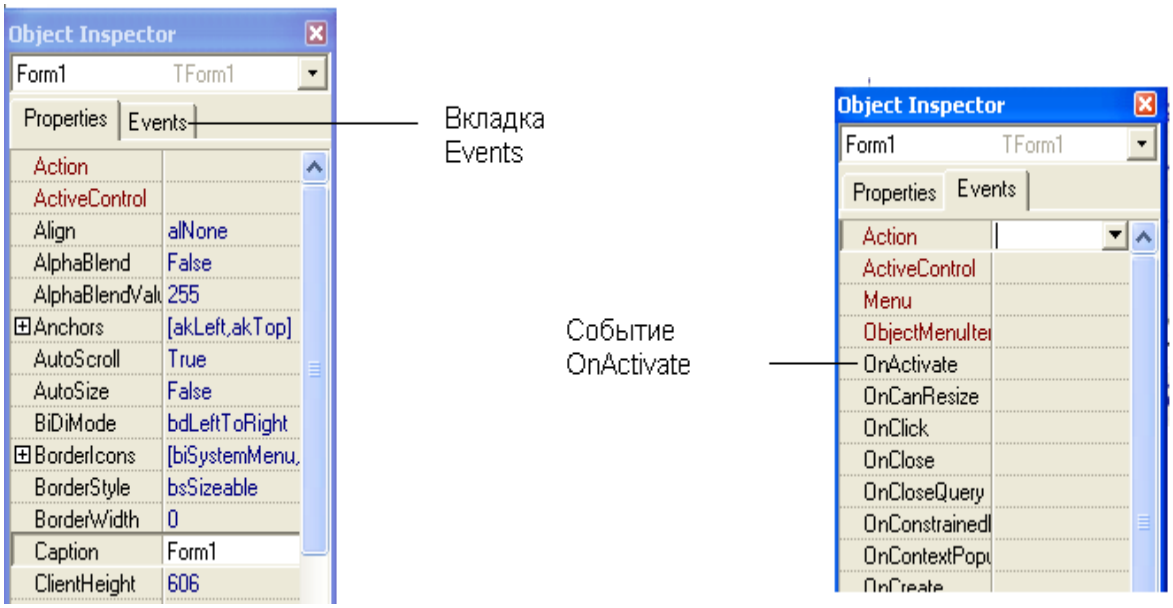
Чтобы сделать таблицу доступной для ввода в нее данных с клавиатуры, надо разрешить ее редактирование. Для этого в списке свойств надо найти строку ***Options*** и нажать на знак «+» слева от этого слова. Раскроется список дополнительных свойств компонента. В этом списке найти свойство ***GoEditing*** и установить его значение в ***True.***



4. Автоматический вывод заголовков столбцов таблицы и автоматическая нумерация строк

При запуске программы на выполнение в окне программы появится изображение таблицы. Пользователю должно быть понятно, какие данные надо вводить с клавиатуры. Для этого у столбцов таблицы должны быть заголовки (см. табл. Выше). Числа дней месяца – это целые числа от 1 до 31. Пользователя надо освободить от их ввода и пронумеровать строки таблицы автоматически. Запрограммируем вывод заголовка таблицы и нумерацию строк таблицы в процедуре *Form (активизация формы)*, которая автоматически выполняется. При запуске программы на выполнение происходит событие формы *onActivate.*

Сделать активной форму, щелкнув в любом месте форме, где нет компонентов. В окне Object Inspector щелчком перейти на вкладку **Events** (События).



Дважды щелкнуть в пустой строке справа от названия события ***onActivate***. Появится шаблон процедуры FormActivate, которая запускается автоматически при запуске программы:

Набрать следующий текст

```
procedure TForm1.FormActivate(Sender: TObject);
var
  i: integer;
begin
  StringGrid1.Cells [0,0] :='День месяца';
  StringGrid1.Cells [1,0] :='Количество осадков, мм';
  For i:=1 to 31 do
    StringGrid1.Cells [0,i] := IntToStr(i);
end;
```

Ввод текста программы для обработки массива и решения задачи.

Вернитесь на форму. Дважды щелкните по командной кнопке на форме и введите следующий текст :

```
procedure TForm1.Button1Click(Sender: TObject);
var
   A : array [1..31] of integer;
   i, Sum : integer;
begin
{ Ввод элементов массива  }
    For i:=1 to 31 do
     A[i] := StrToInt(StringGrid1.Cells[1,i]);
{ Суммирование значений элементов массива  }
    Sum:=0;
    For i:=1 to 31 do Sum:=Sum + A[i];
{ Вывод результата  }
  Label2.Caption  :=  'Суммарное  количество  осадков  в  марте'+
IntToStr (Sum);
end;
```

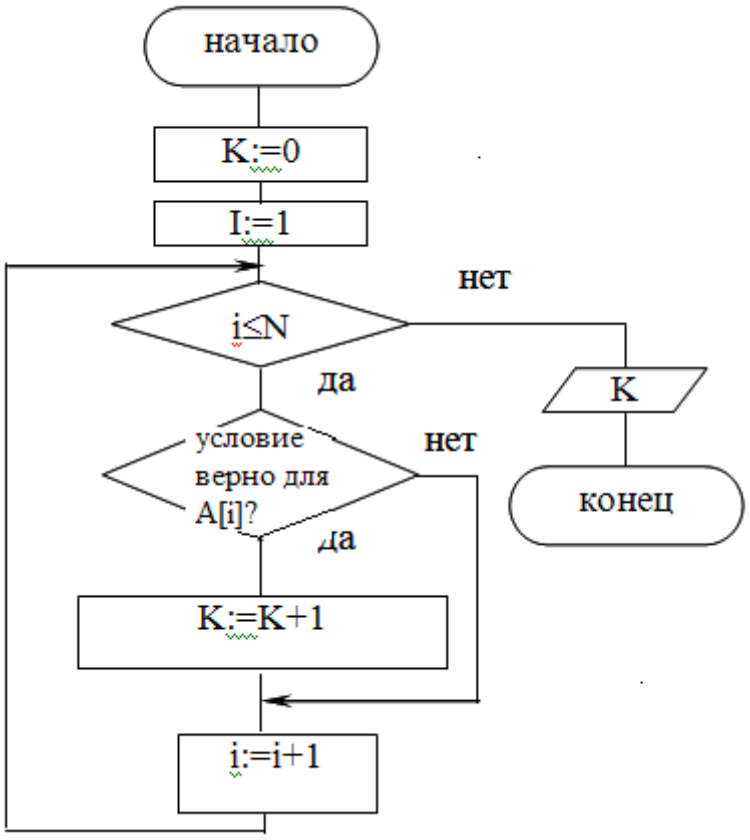
Сохранить проект и запустить его на выполнение. Ввести числа и получить их сумму.

2. Алгоритм подсчета количества элементов массива, удовлетворяющих заданному условию

Общая формулировка задачи.

Дан числовой массив . Сколько элементов массива удовлетворяют определенному условию?

Обобщенный алгоритм



A – массив исходных данных
N – количество элементов в массиве A
K – искомое количество элементов массива A, для которых выпоняется заданное условие
I – счетчик цикла

Рис.2. Алгоритм подсчета количества элементов массива, удовлетворяющих заданному условию

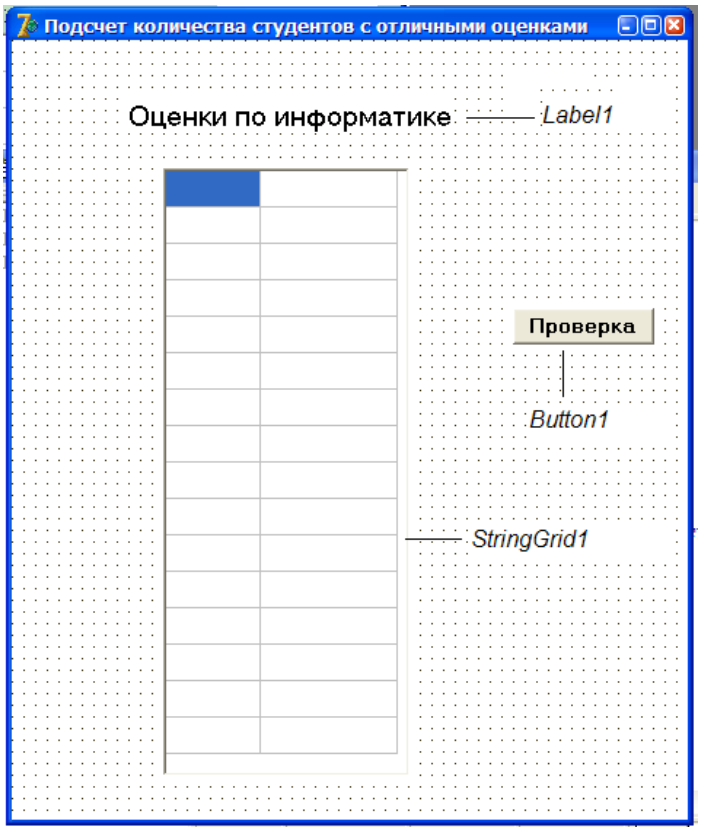
Пример

Даны оценки 15-ти студентов по информатике. Сколько студентов имеют отличные оценки?

Исходные для решения задачи представляются в виде таблицы

№ по порядку	Оценка
1	5
2	4
15	5

Форма имеет следующий вид.



Текст процедуры обработки нажатия командной кнопки

```
procedure TForm1.Button1Click(Sender: TObject);
var
   A : array [1..15] of integer;
   i, k : integer;
begin
{ Ввод элементов массива  A }
    For i:=1 to 15 do
     A [i] := StrToInt(StringGrid1.Cells[1,i]);
{ Подсчет количества элементов массива  A, равных 5 }
    k:=0;
    For i:=1 to 15 do if A[i]=5 then k:=k+1;
{ Вывод результата  - сообщения о количестве студентов с отличными оценками }
  ShowMessage ('Отличные оценки получили '+ IntToStr (k)+ ' студентов');
end;
```

ЗАДАНИЕ

1. Создать форму для примера .
2. Задать необходимое количество строк и столбцов таблицы.
3. Разрешить ввод данных в таблицу.
4. Создать процедуру обработки события OnActivate для автоматической нумерации строк таблицы и вывода заголовков столбцов.
5. Создать процедуру обработки события нажатия командной кнопки.
6. Сохранить проект в отдельной папке.
7. Запустить программу на выполнение, ввести исходные данные и получить результат.