МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

Лабораторный практикум и практические занятия к курсу «Информатика»

разработчик
к.т.н, доцент кафедры АСУ
Суханов А.Я.

Томск 2019

Суханов А.Я.

Информатика: Учебное методическое пособие для выполнения лабораторных работ и практический заданий – 137 с.

Учебное методическое пособие содержит программу и задания для лабораторных занятий и практических занятий, а так же все необходимые формы документов для выполнения заданий.

Оглавление

1 LibreOffice Writer

Сначала идет теоретический материал, в конце главы идет задание.

LibreOffice Writer это текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с возможностью применения простейших форм алгоритмов в виде макросов. LibreOffice это свободный независимый офисный пакет с открытым исходным кодом, разрабатываемый The Document Foundation как ответвление от разработки OpenOffice.org, в который входит и текстовый процессор Writer. Довольно подробную информацию о пакете LibreOffice можно найти на сайте http://help.libreoffice.org/Writer/Welcome_to_the_Writer_Help/ru.

Любой текстовый процессор представляет собой прикладную компьютерную программу, предназначенную для производства, включая операции набора, редактирования, форматирования, печати, любого вида печатной информации. Иногда текстовый процессор называют текстовым редактором второго рода.

Текстовыми процессорами в 1970-е — 1980-е годы называли предназначенные для набора и печати текстов машины индивидуального и офисного использования, состоящие из клавиатуры, встроенного компьютера для простейшего редактирования текста, а также электрического печатного устройства. Позднее наименование «текстовый процессор» стало использоваться для компьютерных программ, предназначенных для аналогичного использования. Текстовые процессоры, в отличие от текстовых редакторов, имеют больше возможностей для форматирования текста, внедрения в него графики, формул, таблиц и других объектов. Поэтому они могут быть использованы не только для набора текстов, но и для создания различного рода документов, в том числе официальных. Программы для работы с текстами также можно разделить на простые текстовые процессоры, мощные текстовые процессоры и издательские системы.

1.1 Запуск LibreOffice Writer

Прежде всего нужно запустить программу LibreOffice Writer.

В зависимости от используемой операционной системы, например, Linux или Windows нужно действовать по следующему алгоритму, во многом он одинаков для указанных операционных систем:

В меню Пуск(Windows) выбрать Программы+LibreOffice и запустить WordProcessor LibreOffice Writer или Office LibreOffice, в графической оболочке KDE или GNOME Linux можно выбрать меню Запуска приложений (Application Louncher) и в подменю Приложения (Applications) Office и LibreOffice. При выборе LibreOffice откроется окно создания документов

LibreOffice (рисунок 1.1), среди которых документы Writer, в указанном окне документы Writer отмечены как Text Document. При выборе WordProcessor LibreOffice Writer сразу же откроется окно с пустым бланком документа (рисунок 1.2).

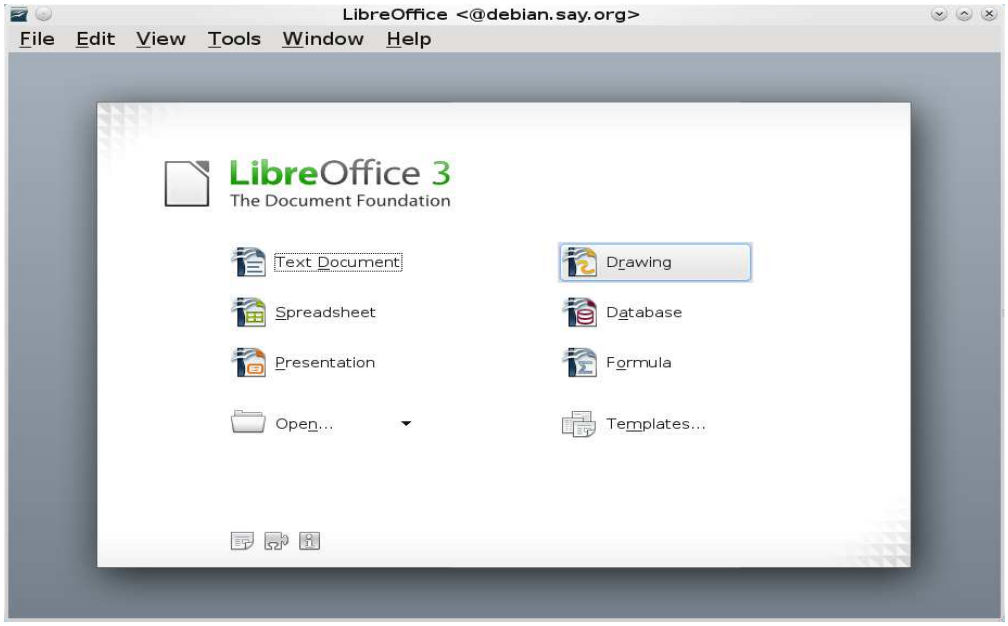


Рисунок 1.1 – LibreOffice

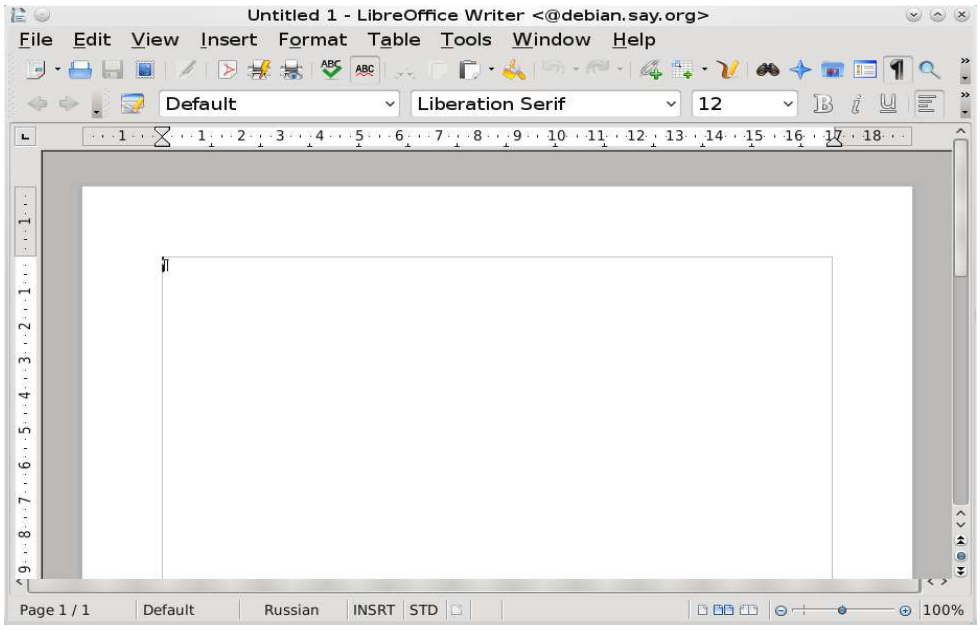


Рисунок 1.2 - Окно с документом LibreOffice Writer

1.2 Ввод текста

Основной составляющей документов LibreOffice Writer: писем, записок, плакатов, деловых бумаг — обычно является текст. Введите какой-нибудь текст в новый документ Writer, который открывается при запуске программы.

1 . Введите какое-нибудь предложение.

2. Нажмите клавишу Enter.

Чтобы переключиться с русской раскладки клавиатуры на английскую, нужно нажать клавиши Ctrl+Shift или Alt+Shift — в зависимости от настроек Windows или Linux. Индикатор клавиатуры отображается в панели задач рядом с часами. Вы можете переключить раскладку также с помощью мыши. Для этого щелкните левой кнопкой мыши на индикаторе и выберите нужную раскладку в появившемся меню. Чтобы удалить символ слева от курсора (мерцающая вертикальная черта), нажмите клавишу Backspace. Чтобы удалить символ справа от курсора, нажмите клавишу Delete.

Правка текста

После первоначального ввода текста вам наверняка потребуется изменять его. Давайте попробуем добавить и затем удалить текст в документе. Курсор показывает, в каком месте документа будут появляться символы, вводимые с клавиатуры. Один раз щелкните левой кнопкой мыши в документе, чтобы изменить положение курсора. Курсор также можно переместить с помощью клавиш со стрелками.

По умолчанию Writer работает в режиме вставки. Это значит, что при вводе весь текст справа от курсора сдвигается, чтобы освободить место для нового текста.

4. Дважды щелкните на каком-нибудь слове.

5. Нажмите клавишу Delete.

Существующий текст сдвинется обратно и заполнит освободившееся место.

1.3 Форматирование текста

1. Щелкните левой кнопкой мыши на поле страницы.

2. Щелкните на пункте Символы в строке меню.

3. Выберите вкладку Шрифт (Character).

4. В списке Family (Семейство шрифтов) выберите шрифт с названием Liberation Serif.

5. В списке Начертание (Style) выберите пункт Полужирный (Bold).

6. Прокрутите список Размер (Size) с помощью полосы прокрутки и выберите значение 24.

7. В области Эффекты (Font Effects) установите флажок С тенью (Shadow).

8. Щелкните на кнопке ОК.

9. Щелкните в любом месте документа, чтобы снять выделение с текста.

Кроме того, можно уплотнить шрифт, это делается, например, чтобы текст занимал определенное число страниц или определенный объем, если вдруг полученный объем больше чем требуемый (Вкладка Положение и Интервал, выбрать разреженный или уплотненный (Scale Width)).

Ту же страницу параметров символов можно выбрать в меню Формат.

Кроме того, можно отдельно форматировать параметры абзаца и страницы, их также можно выбрать в меню Формат. Параметры страницы и абзаца рассматриваются далее.

1.4 Сохранение документа

Документы обязательно нужно сохранять. Частота сохранения документа соответствует времени, которое вам не жалко тратить на восстановление данных, потерянных в случае сбоя компьютера.

Выберите в строке меню пункт Файл (File).

Выберите команду Сохранить (Save). На экране появится окно диалога Сохранение документа (Save As), показанное на рисунке 1.3. Writer автоматически предлагает имя для документа (обычно Untitled 1). Введите любое другое имя

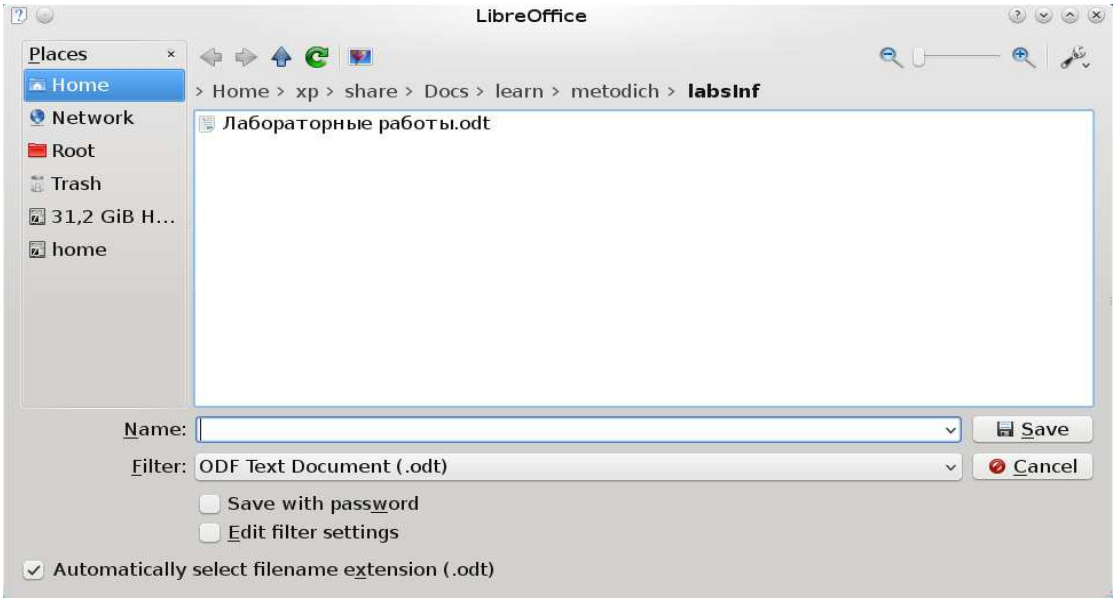


Рисунок 1.3 - Окно сохранения файла LibreOffice

В текстовом поле Имя файла (name) введите имя файла.

Вводимый вами текст заменит текст, выделенный в поле Имя файла (name).

Кроме того, для удаления текста здесь также можно использовать клавиши Backspace и Delete.

Раскройте список Папка (Save in) в верхней части окна диалога.

Выберите любой ваш диск или папку.

Предположим, что вы решили добавить еще пару слов в свой документ. Как снова его открыть?

Выберите в строке меню пункт Файл (File).

Выберите команду Открыть (Open).

Выберите диск. Раскройте список Папок и файлов.

Щелкните на значке своей папки.

Выделите значок своего документа

Щелкните на кнопке Открыть (Open).

Панели инструментов предоставляют доступ к некоторым наиболее часто используемым командам меню. Если вы владеете мышью лучше, чем клавиатурой, вам будет удобнее работать с панелями инструментов.

Рисунок 1.4 - Вид панелей инструментов

Вывод панели инструментов на экран Word содержит множество панелей инструментов, которые обычно объединяют кнопки, относящиеся к какой-нибудь большой теме, например, Таблицы и границы (Tables and Borders), Рисование (Drawing), Базы данных (Database) и Веб-узел (Web).

Их можно выводить на экран и убирать с него по мере необходимости.

Щелкните правой кнопкой мыши на любой панели инструментов или строке меню и выберете Customize Toolbar. На экране появится выпадающий список всех панелей инструментов и текущая панель на которой вы открыли меню, причем флажками будут помечены те инструменты, которые в данный момент отображаются на экране. Панели Стандартная (Standard) и Форматирование (Formatting) занимают одну строку (рисунок 5). Обычно Панель Рисование (Drawing) закреплена в нижней части экрана, Панель Таблицы и границы (Tables and Borders) «плавает» на экране.

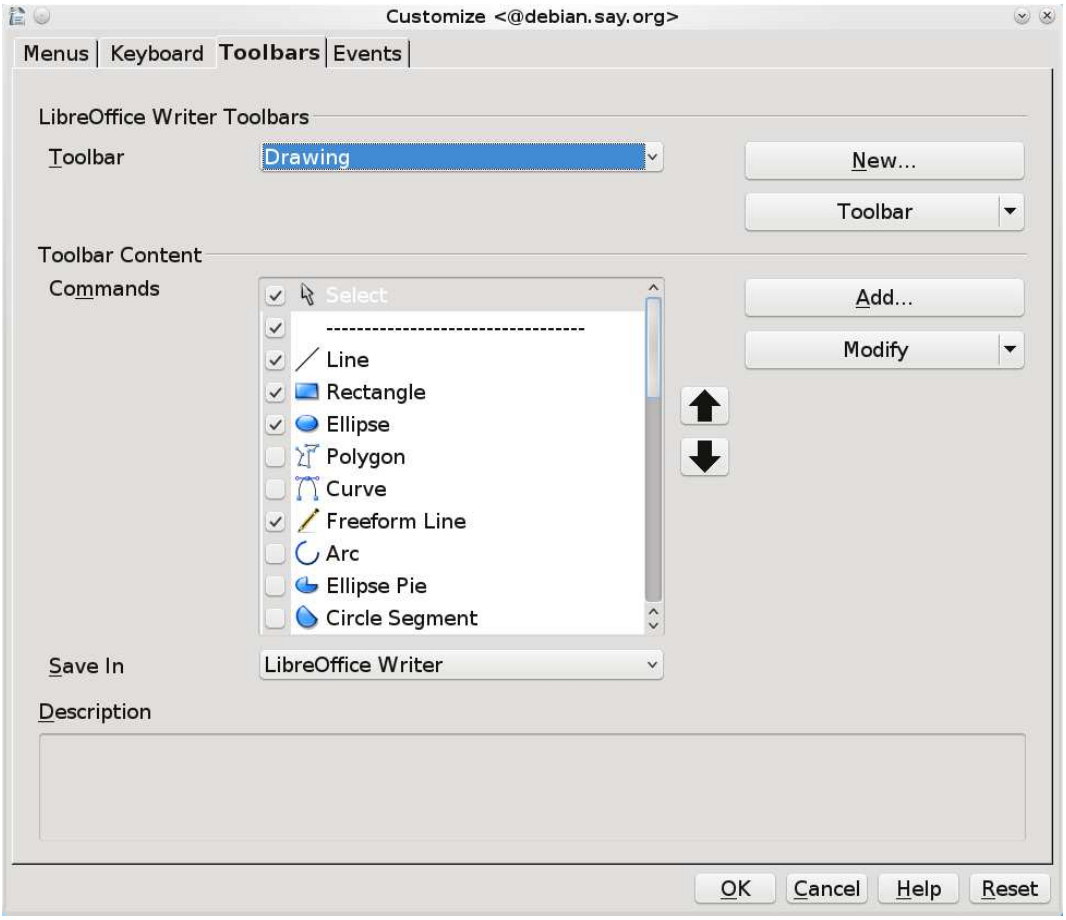


Рисунок 1.5 - Настройка Панели инструментов (ToolBar) Drawing

В списке панелей выберете имя необходимой панели инструментов, например, на строке Рисование (Drawing). На экране появится еще одна панель инструментов.

1.5 Использование панелей инструментов

Давайте посмотрим, как создать, сохранить, закрыть, а затем снова открыть документ и отправить его по электронной почте, пользуясь кнопками панелей инструментов. Названия кнопок отображаются на экране, если ненадолго задержать на них указатель мыши.

Щелкните на кнопке New (New Blank Document). Writer создаст новый документ. Его название появится в строке заголовка окна.

Введите слово Пример.

Щелкните на кнопке Сохранить (Save). На экране появится окно диалога сохранения документа. В поле Имя файла (File name) Writer предлагает свой вариант имени для файла.

Щелкните на кнопке Сохранить (Save) в окне диалога Сохранение документа (Save As).

Выберите команду Файл > Закрыть (File > Close). Только что созданный нами документ будет закрыт.

Щелкните на кнопке Открыть (Open).

В окне диалога Открытие документа (Open) выделите последний сохраненный документ и щелкните на кнопке Открыть (Open).

Введите слова Мой первый и щелкните на кнопке Сохранить (Save). Теперь документ будет автоматически сохранен.

Выберите команду Файл Закрыть (File > Close).

1.6 Добавление новых возможностей на панель инструментов.

Иногда удобно вынести на панель инструментов какие-то дополнительные функциональные возможности, например, средство для обрезки рисунков, возможность писать подстрочные и надстрочные знаки, формулы. Для этого необходимо выбрать Tools, Customize (рисунок 1.6).

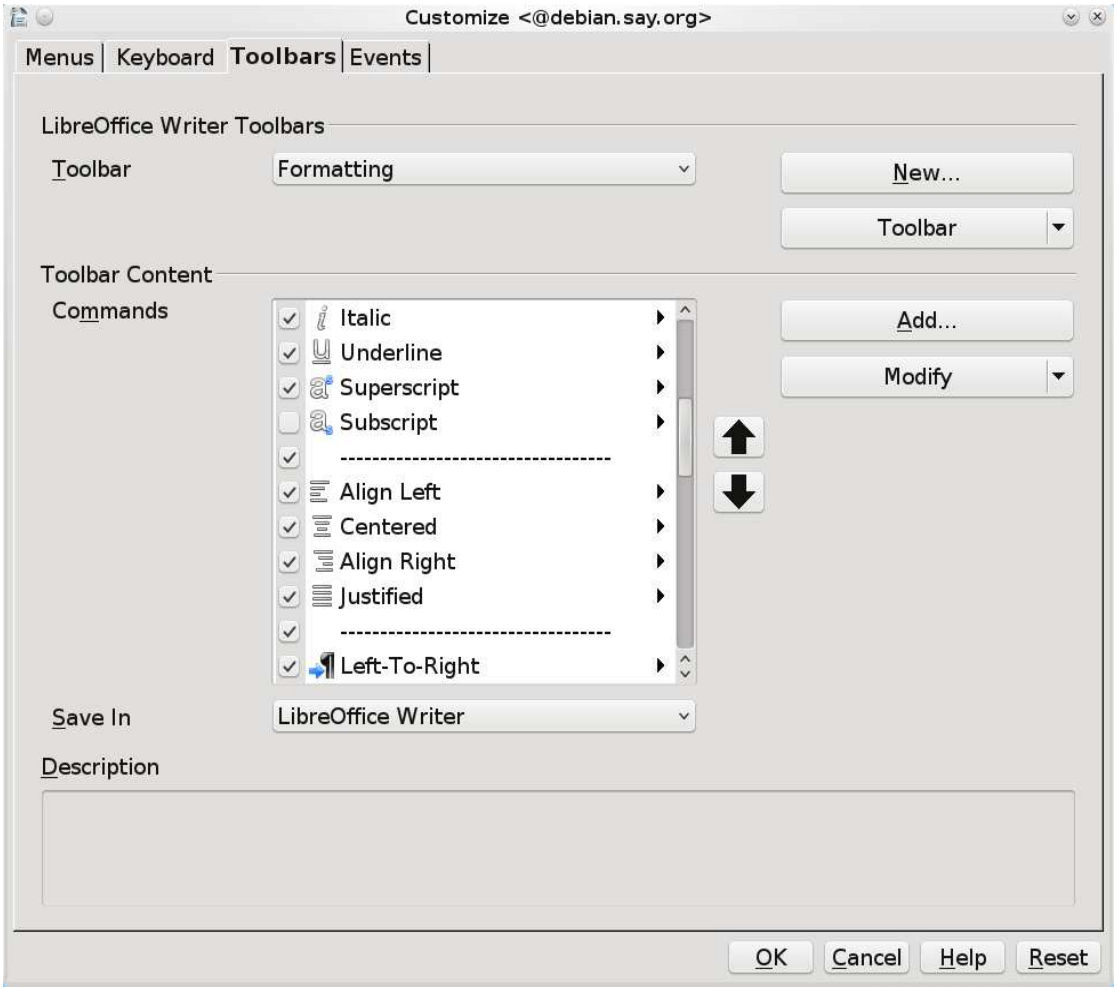


Рисунок 1.6 - Пример добавления надстрочного знака

Например, выбираем в списке панелей Формат (Formatting), в в нижнем списке ищем нужное поле «Надстрочный знак» (as) и ставим галочку, можно нужный элемент добавить на другую панель, с помощью кнопки Добавить (Add).

Таким же образом можно добавить редактор формул, или знак отображения невидимых и специальных символов, если нет необходимости добавлять всю панель, можно добавить нужный элемент на уже отображенную панель (рисунок 1.7). При этом в окне Customize в списке должна быть выбрана панель Standard.

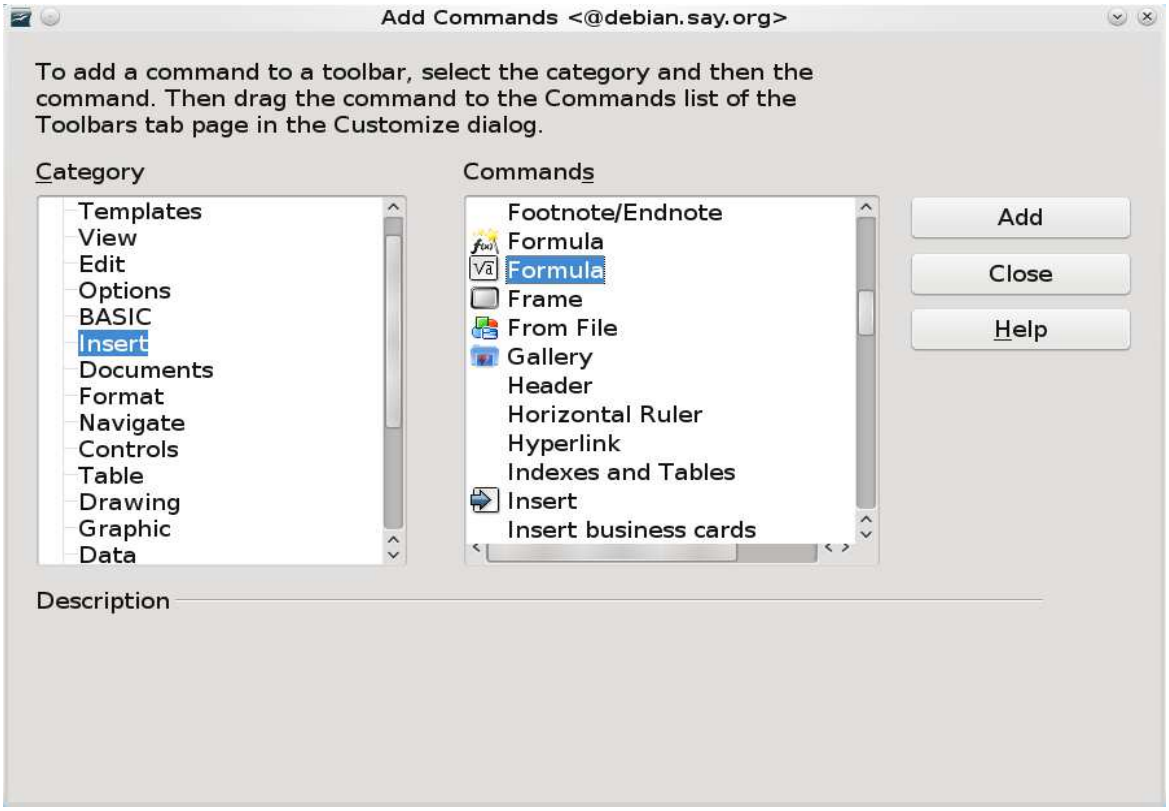


Рисунок 1.7 - Пример добавления редактора формул на панель Standard

Кроме того можно сделать видимой одну из панелей инструментов выбрав соответствующую вкладку отметив нужную панель во View-Toolbars.

1.7 Редактирование текста

Приемы выделения текста

Прежде чем выполнить какую-либо операцию с текстом, нужно указать программе Writer, какой именно фрагмент (часть слова, слово, абзац или весь текст) вы хотите изменить. Для этого существуют средства выделения текста. Текст можно выделять с помощью клавиш и с помощью мыши. Основные сочетания клавиш и другие приемы для выделения текста перечислены ниже.

Сочетание клавиш или прием для выделения текста.

Один символ справа от курсора Shift+→

Один символ слева от курсора Shift+←

Одно слово справа от курсора Shift+Ctrl+→

Одно слово слева от курсора Shift+Ctrl+←

Одну строку выше курсора Shift+↑

Одну строку ниже курсора Shift+↓.

Весь текст от курсора до конца строк!-: Shift+End

Весь текст от курсора до начала строки Shift+Home

Весь текст от курсора до конца документа Shift+Ctrl+End

Весь текст от курсора до начала документа Shift+Ctrl+Home

Для выделения одного слова можно воспользоваться двойным щелчком левой кнопки мыши на слове, для выделения предложения повторение двойного щелчка.

Для выделения колонки текста можно воспользоваться переходом в режим копирования блока текста Ctrl+Shift+F8.

Выделить весь текст Ctrl+A.

Удаление, копирование и вставка текста для удаления текста мы пользовались клавишами Backspace и Delete.

Можно просто выделить ненужный фрагмент и нажать клавишу Delete (или Backspace).

Для отмены удаления можно воспользоваться кнопкой Undo (отмена ввода)

Копирование выделенного текста в буфер — Ctrl+Ins, Ctrl+C

Вставка из буфера – Shift+Ins, Ctrl+V

Удаление и копирование в буфер — Shift+Del.

1.8 Параметры страницы

Меню Формат(Format)+Страница(Page) позволяет задать такие свойства как отступы от краев листа (рисунок 1.8), колонтитулы, поворот страницы – альбомная или книжная, тип печатаемой страницы, например, А4, стандартный лист для принтера, А5 – половина данного листа, А3 - два листа А4.

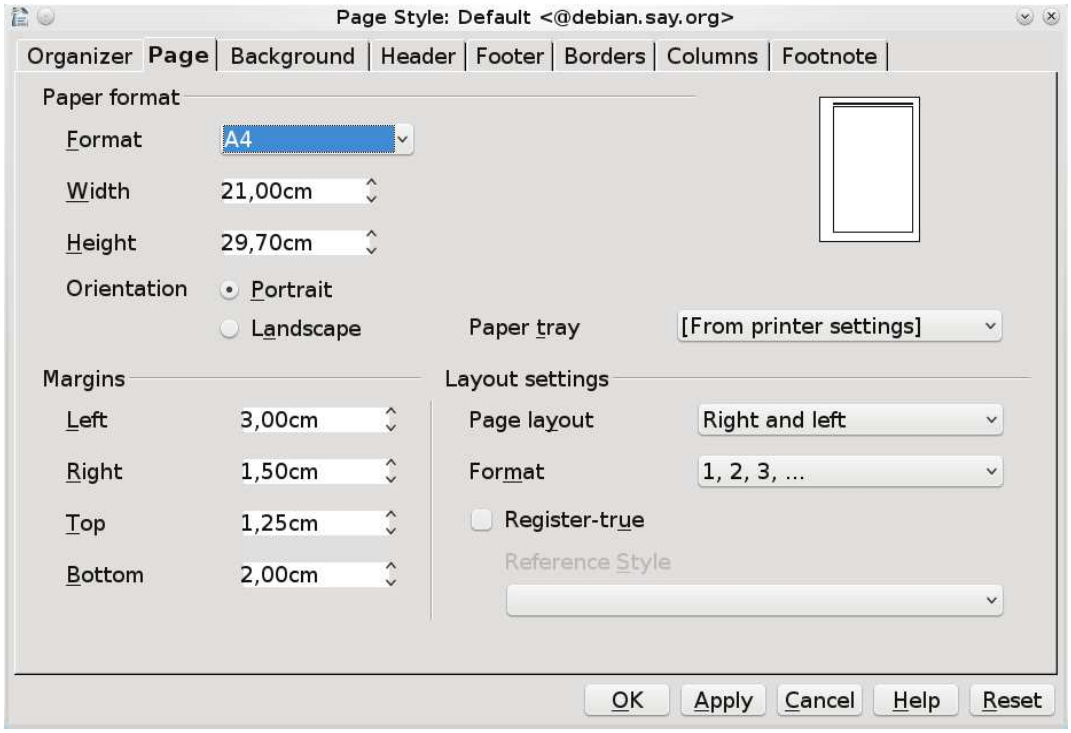


Рисунок 1.8 - Настройка параметров страницы

1.9 Оформление абзацев (Paragraphs)

Чаще всего требуется менять такие параметры оформления абзаца, как выравнивание по левому, правому краю, выравнивание по ширине страницы, центрирование абзаца, изменение межстрочного расстояния и отбивка абзаца. Для того, чтобы переформатировать текущий абзац, выберите команду в меню Формат(Format)+Абзац (Paragraph) (рисунок 1.9). Если необходимо переформатировать более одного абзаца, необходимо их предварительно выделить.

Координатная линейка, которая отображается по команде в меню Вид (View)+Линейка (Ruler), позволяет менять отступы и величину красной строки, средний – меняет абзацный отступ, нижний сдвигает красную строку и абзацный отступ одновременно. Правый маркер служит для изменения правого отступа.

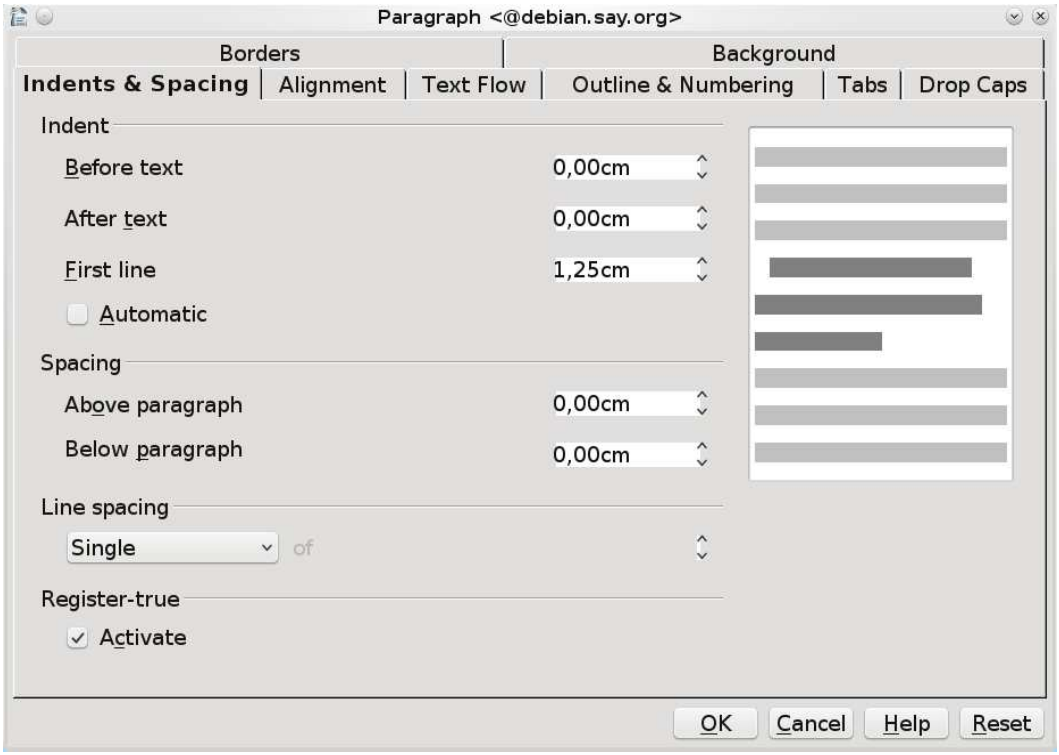


Рисунок 1.9 - Настройка параметров параграфа (абзаца)

1.10 Разделы (Sections) и разрывы

Для изменения разметки документов на одной странице или на разных страницах можно использовать разделы. Чтобы создать область с разделом, вставьте раздел и затем задайте формат для каждого из разделов. Например, можно при создании отчета отформатировать раздел введения как одну колонку, а затем отформатировать следующий раздел, представляющий собой текст отчета как две колонки.

Для вставки раздела нужно выбрать Вставка+Раздел (Insert+Section) и соответствующие параметры нового раздела, где он начинается (рисунок 1.10). В новом разделе можно установить другое количество колонок текста, чем то, количество, что было в другом разделе, настроить количество колонок можно во вкладке Колонки (Columns).

Для вставки разрыва необходимо выбрать Вставка+Разрыв (Insert + Manual Break) (рисунок 11), в этом случае можно выбрать разрыв на следующую страницу, строку.

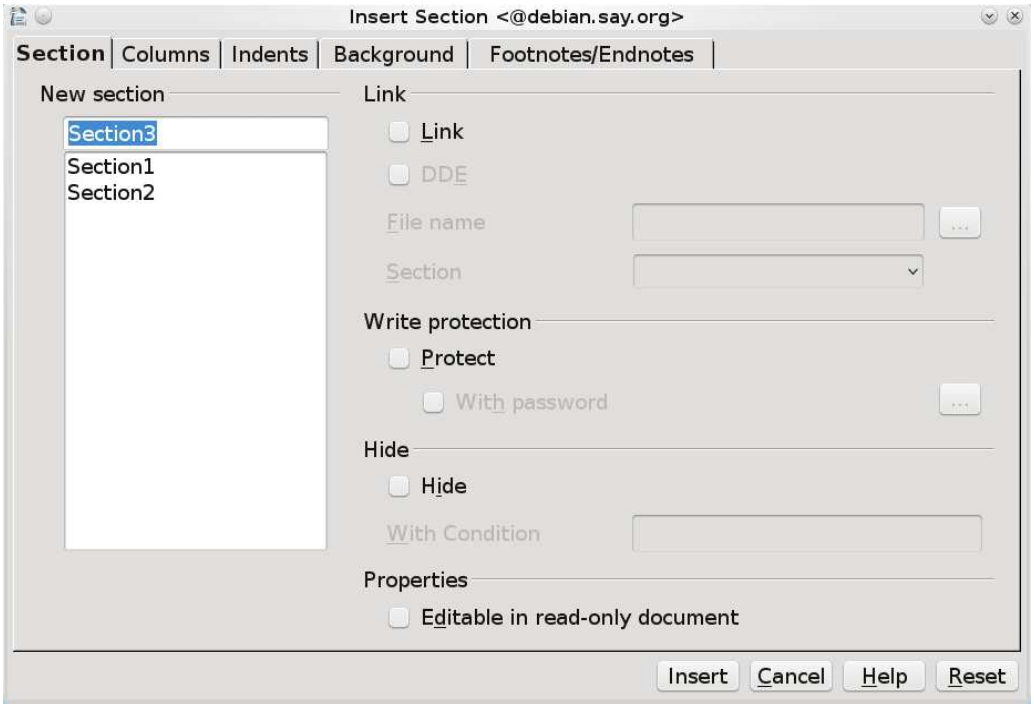


Рисунок 1.10 - Окно для вставки раздела и установка параметров раздела

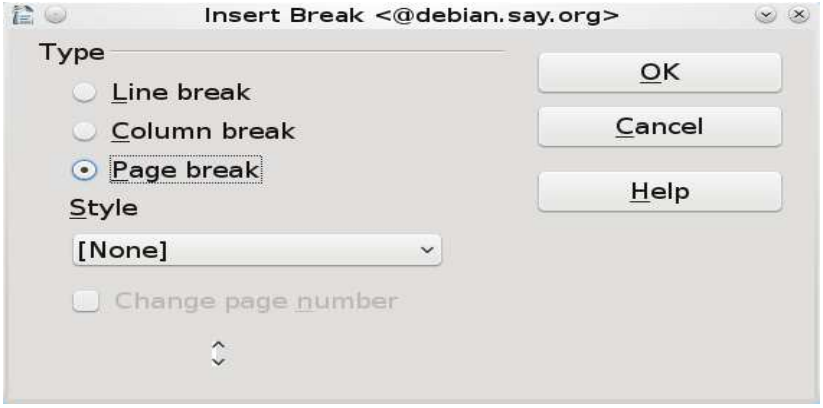


Рисунок 1.11 - Вставка разрыва на другую страницу

Для изменения ориентации страницы для всех страниц с одинаковым стилем сначала следует создать соответствующий стиль страницы, а затем применить этот стиль:

Выберите команду **Формат - Стили и форматирование**.

Щелкните значок **Стили страницы**.

Щелкните стиль страницы правой кнопкой мыши и выберите **Новый**. Новый стиль страницы изначально получает все свойства выбранного стиля страницы.

На вкладке **Управление** введите имя для стиля страницы в поле **Имя**, например "Моя альбомная ориентация".

В поле **Следующий стиль** выберите стиль страницы, который требуется применить к странице, следующей за страницей с новым стилем. См. раздел о применении стилей страниц в конце данной страницы справки.

Откройте вкладку **Страница**.

В пункте **Формат бумаги** выберите “Портретный” или “Альбомный”.

Нажмите кнопку **ОК**.

Теперь определен соответствующий стиль страницы под именем "Моя альбомная ориентация". Для применения нового стиля дважды щелкните стиль страницы "Моя альбомная ориентация" в окне **Стили и форматирование**. Изменяются все страницы текущей области стилей страницы. При выборе другого стиля в качестве "следующего стиля" изменяется только первая страница текущей области стилей страницы.

Одностраничные стили

Стиль страницы можно применить только к одной странице. В качестве примера рассмотрим стиль “Первая страница”. Для установки этого свойства определите другой стиль страницы в качестве "следующего стиля" на вкладке **Формат - Страница - Управление**.

Одностраничный стиль начинается с нижней границы текущего диапазона стиля страницы и применяется до следующего разрыва страницы. Следующий разрыв страниц появляется автоматически, когда текст переходит на следующую страницу, что иногда называется "мягкий разрыв страницы". В качестве альтернативы можно вставить разрыв страниц вручную.

Для вставки разрыва страницы вручную в положении курсора нажмите `Ctrl+Enter` или выберите **Вставка - Разрыв** и просто нажмите кнопку "OK".

1.11 Оглавление и указатели.

Добавление оглавления и указателей требует предварительной установки свойств текста, который будет выступать в качестве заголовка в оглавлении. Обычно это название параграфа или какой-то главы. Можно воспользоваться кнопкой на панели инструментов – стили и форматирование или стиль (Apply Style), обычно они находятся рядом с полями – название шрифта и размер шрифта. Выделив нужный текст, можно задать ему уровень заголовка, например заголовок 1, или заголовок 2. Заголовок 1 (Heading 1, Heading 2, Heading 3), обычно указывает введение, главы, а заголовок 2 параграфы в этих главах. После того как будут отмечены соответствующие главы и параграфы документа, текст будет полностью напечатан и отформатирован, можно воспользоваться **Вставка - Оглавления и указатели - Оглавления и указатели — Оглавление/указатель** (**Insert - Indexes and Tables - Indexes and Tables — Index/Table)**, чтобы вставить соответствующее оглавление (содержание) в начале документа .

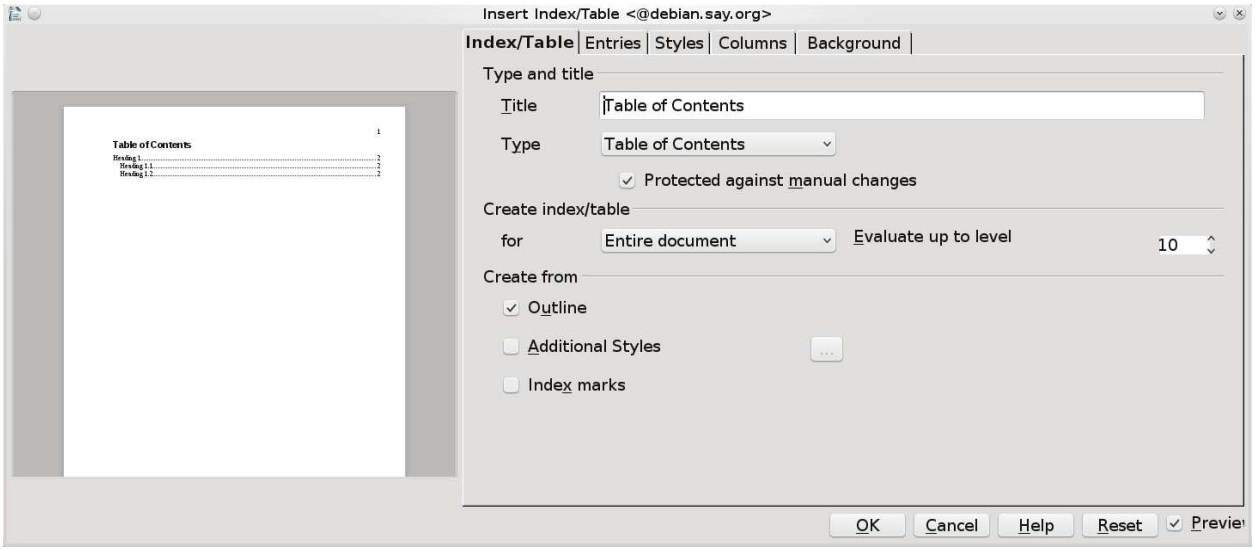


Рисунок 1.10 - Вставка оглавления и указателей

1.12 Вставка рисунка в текст.

Для вставки рисунка можно воспользоваться меню Вставка-Рисунок-Из файла (Insert-Picture-From File), либо можно скопировать изображение в буфер и вставить из буфера с помощью Shift-Ins или меню выбираемого правой кнопкой мыши (вставить - paste), либо сделать скриншот экрана или окна, скопировав его в буфер и затем осуществить вставку. Положение в тексте рисунка можно редактировать (Формат-Якорь, Format-Anchor), привязав его к странице (To Page), параграфу (To paragraph), или символу (To Character), или сделать воспринимаемым как символ (As Character) , в этом случае к рисунку можно принимать стили присущие тексту, выравнивание по ширине или по левому и правому краю и т.д. В тексте отчетов или больших текстах с нумеруемыми рисунками изображение лучше вставлять как символ, также можно задать подпись рисунка — Caption, для этого можно щелкнуть правой кнопкой мыши на рисунке или в меню Вставка-Надпись(Insert-Caption). Можно заменить стандартную надпись на свою, при этом при вставке нового рисунка с надписью он автоматически нумеруется в соответствии со своим номером в тексте.

Рисунок 1.11 - Настройка подписи рисунка

1.13 Формулы

Вставка формулы осуществляется с помощью выбора Вставка-Объект-Формула (Insert-Object-Formula), либо объект формула можно выбрать на панели управления, обычно это кнопка на которой нарисован символ квадратный корень из a. Редактирование формулы можно осуществить с использованием окна редактирования формулы и элементов формулы Вид-Элементы (View-Elements) (рисунок 1.14).

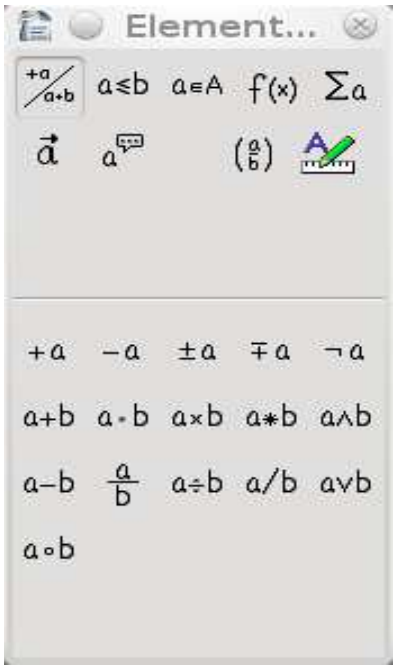


Рисунок 1.12 - Элементы редактирования формулы

Запишем формулу для расчета информационной энтропии:

$$H = \sum_{i=1}^{n} p_i \cdot \log_2\left(\frac{1}{p_i}\right) \quad \textbf{(1)}$$

где p_i - вероятность встречи символа в сообщении.

Таким образом формула выглядит в редакторе формул:

H=sum from{i=1} to{n} {p_{i} cdot log_{2}({1} over {p_{i}}) }.

При редактировании может быть использовано окно элементов редактирования. Например, чтобы выбрать значок суммы выбираем соответствующую вкладку, она будет соответствовать стандартной форме записи sum <?> (Рисунок 1.15), необходимо заменить последовательность <?> на свои данные в соответствии с вашей формулой, можно также указать элементы from и to определяющие нижнюю и верхнюю границы изменения ндекса формулы суммы. Фигурные скобочки объединяют последовательность символов в один элемент формулы.

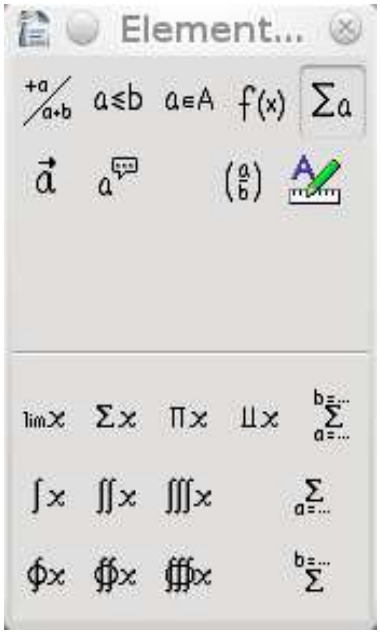


Рисунок 1.13 - Редактирование формулы

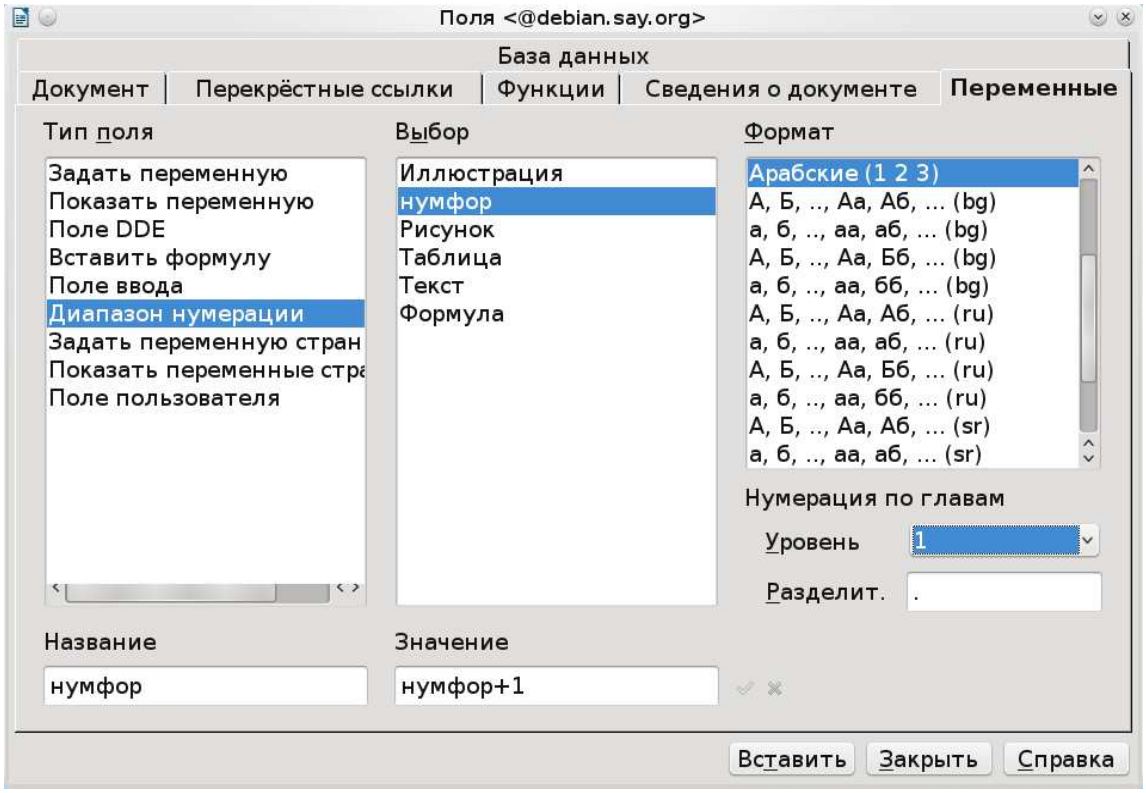


Рисунок 1.14 - Настройка поля

Для создания и вставки формулы использовалось автоматическая нумерация формул. В этом случае можно создать таблицу из двух столбцов и одной строки, сделать границы таблицы невидимыми, в правый столбец вставить формулу и в левый записать скобочки. Выровнять формулу первый и второй столбец по левому и правому краю соответственно, положение в ячейке по высоте задать — по середине. Затем с помощью Вставка-Поля-Дополнительно вызвать окно редактирования представленное на рисунке 1.16. Вставить в список Выбор-Название переменную нумфор, в значение записать нумфор+1. Если необходима нумерация внутри параграфа, то выбрать уровень 1. Ниже приведен пример.

$$H = \sum_{i=1}^{n} p_i \cdot \log_2\left(\frac{1}{p_i}\right) \qquad \textbf{(2)}$$

Затем можно копировать созданную таблицу формулы в нужно место и затем редактировать формулу, так же можно сделать авто-вставку.

Также для автоматической нумерации формул можно воспользоваться вставкой в тексте документа последовательности двух символов fn и затем нажав F3. В этом случае также, но уже автоматически создастся таблица из двух столбцов и одной строки в первом столбце будет указана стандартная формула, ее можно заменить и во втором столбце будет указан номер формулы, данный номер вставляется автоматически, при вставке формулы выше произойдет перенумерация формул. Нумерацию можно делать и не сквозной, щелкнув на соответствующем номере формулы и задав уровень при настройке поля.

$$E = mc^2 \quad (2)$$

$$E = mc^2 \quad (3)$$

1.14 Стили и форматирование

При работе в Writer лучше заранее создать набор стилей для различного типа объектов, рисунков, формул, текста, списков, заголовков и затем применять данные стили к тексту, а не настраивать все вручную.

Для создания своего стиля можно выбрать Формат-Стили и форматирование (Format-Styles and Formatting), в окне стилей выбрать нужный стиль блока текста (параграфа, символа, фрейма или страницы) , затем с помощью правкой кнопки мыши и выпадающего меню выбрать New. Например, при создании стиля во вкладке Frames будет вызвано окно, показанное на рисунке 1.17, если мы назначим доя фрейма указанный стиль, то он будет применен к данного фрейма, для применения стиля можно выделить нужный объект и щелкнуть левой кнопкой мыши на имени стиля два раз, или выбрать стиль в Apply Style. Кроме того, можно создать стиль с аналогичным именем во вкладке paragraphs и characters в свою очередь описывая для данного стиля все характеристики текста. При создании соответствующего текста в данном фрейме он будет отформатирован в соответствии с параметрами данного стиля, а текст внутри фрейма в соответствии с тем же стилем заданным во вкладках characters (стили символа) и paragraphs (стили параграфа).

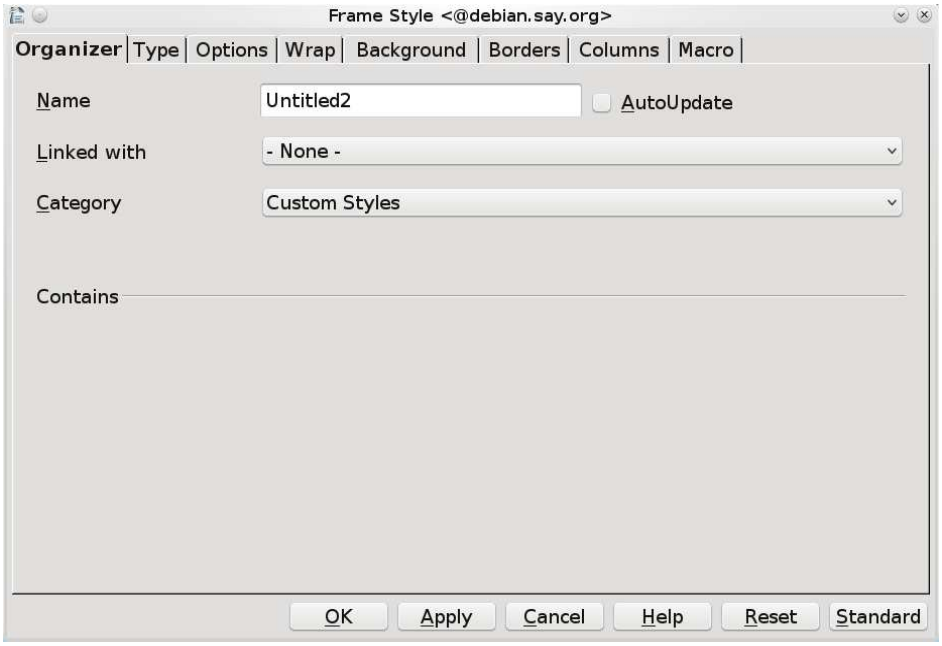


Рисунок 1.15 - Окно форматирования стиля фрейма

1.15 Автозамена и параметры автозамены

Часто бывает, что необходимо автоматически произвести замену текста на другой текст или, например, при вводе номера и точки автоматически создается список, либо первая буква абзаца автоматически становится заглавной, ниже приведен пример отключения автоматической замены номера и точки на элемент списка, иногда это не очень удобно.

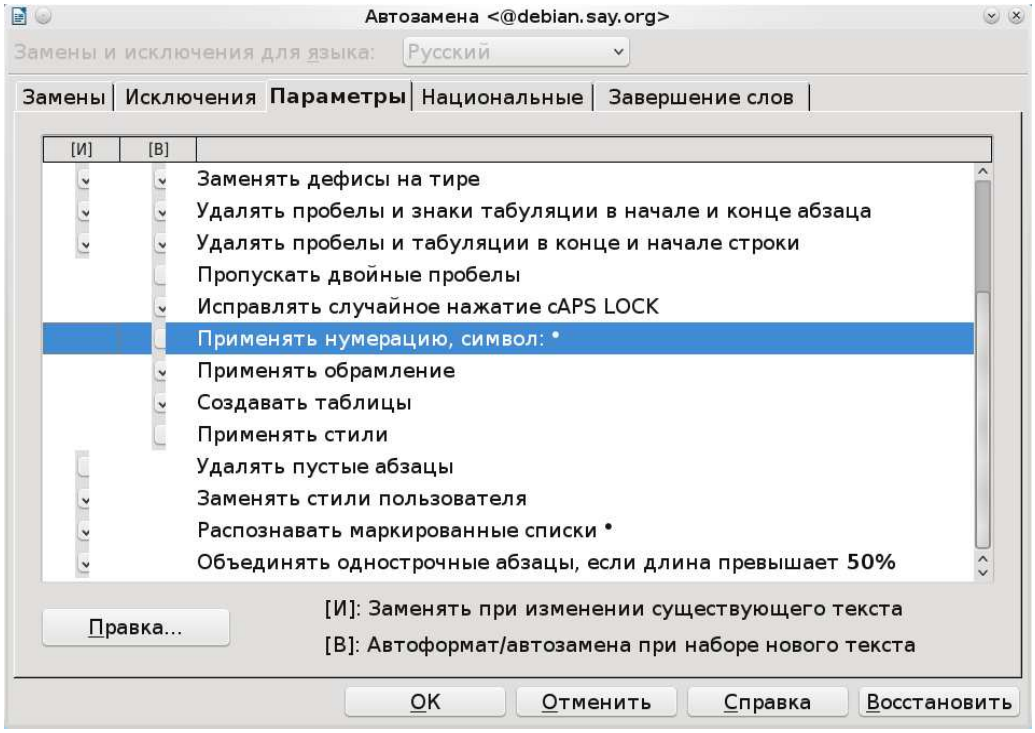


Рисунок 1.16 - Параметры автозамены

1.16 Задание Libre office Writer форматирование документа.

Внимательно изучите возможности Libre Writer описанные в методическом пособии.

Создайте документ Writer. Для этого в меню Пуск найдите приложение LibreOffice Writer. Запустите его, откроется окно с редактором и пустым документом. Сохраните документ на своем диске или в какой-нибудь созданной вами папке на вашем диске.

Работа выполняется с помощью компьютера на одной стороне листа формата А4 через 1,5 интервала с числом строк на странице не более 40. Текст работы следует писать, соблюдая следующие размеры полей:

левое – не менее 30 мм;

правое – не менее 10 мм;

верхнее – не менее 15 мм;

нижнее – не менее 20 мм.

Размер шрифта не менее 12 пт. Отступ красной строки 1.25.

Установите шрифты, параметры страницы, абзацев, создайте свои стили.

Создайте титульный лист, затем с помощью задания стиля заголовков задайте несколько примерных пунктов вашей будущей лабораторной: Оглавление (на отдельной странице), Введение (на отдельной странице), 1. Теоретические основы текстовых процессоров, 1.1. Libre office, 1.2 Microsoft Word, 1.3 Сравнение текстовых процессоров, 2. Ход работы (с подпунктами заданий), Выводы, Список использованный источников.

Предварительно для вставки оглавления необходимо параграфы, входящие в оглавление выделить и обозначить как заголовок какого-либо уровня (это можно сделать выбрав нужный стиль – Заголовок №), например 1. Первый параграф – заголовок 1 уровня, 1.3 – заголовок второго уровня, 4.3.1 – заголовок третьего уровня, формат заголовка можно задать в стилях. (см. методическое пособие). Для вставки номеров страниц необходимо встать в колонтитул и осуществить вставка-поле.

Воспользовавшись любым поисковиком в сети интернет, например, поисковиком google.ru, найдите информацию о пакете LibreOffice и Microsoft Word. Скопируйте текст во вновь созданный документ, воспользовавшись вставкой, чтобы текст был вставлен без ссылок можно воспользоваться «Правка-вставить как...» и указать — «текст без форматирования». Либо вставить текст, затем производя удаление с копированием текста (Shift-Del) и повторной вставки без форматирования вставлять текст. Опишите основные особенности текстовых процессоров входящих в данные пакеты, отличия их друг от друга. Часть информации можно скопировать из различных источников, часть описать самостоятельно. Вставьте вид окна Word любой версии и скриншот рабочего окна LibreOffice Writer. Скриншот текущего рабочего окна – Alt+PrintScreen (Prn Scr), скриншот всего экрана – Shift+PrintScreen или PrintScreen, либо осуществите поиск в сети Интернет.

Научитесь по указаниям выше выделять, копировать и вставлять текст.

Вначале документа вставьте еще одну страницу. Для этого можно воспользоваться меню Вставка (Разрыв – Разрыв страницы). Проведите форматирование и редактирование текста, установив заглавия и разделы с помощью стили и форматирование. Желательно разбить документ на параграфы, первым параграфом будет идти Введение, которое не нумеруется, затем 1. «Текстовые процессоры», затем 1.1 Libre Writer, 1.2 Microsoft Word, 2. Лабораторная работа, 2.1 – Пункт выполнения задания (название пункта). 2.2. Второй пункт (название). 2.2.x – подпункт, если необходим. 3. – Выводы.

Удалите ненужные абзацы, отобразите невидимые символы. Сохраните документ. Воспользуйтесь заменой знака абзаца на пробел в выделенном тексте, для этого выберете «Правка-Найти и заменить». Выберете «Детали» и установите галочку регулярные выражения. Затем в поле «Найти» установите знак $, что означает конец строки, и выберете «найти все»,

затем в поле «Заменить» укажите пробел и нажмите «заменить». Дополнительную информацию по использованию регулярных выражений можно найти на сайте LibreOffice, например, знак ^ означает начало строки, * означает любое количество повторений символа, который стоит перед звездочкой, при поиске "Аб*в" будут найдены "Ав", "Абв", "Аббв", "Абббв" и т. д. Установите выравнивание по ширине для текста, и по середине для рисунков. Установите где нужно списки.

Вставьте уменьшенный «скриншот» выполняемой программы, сделайте «скриншот» рабочего окна и разместите его на отдельной странице сделав ее альбомной, при этом остальные должны оставаться книжными. Рисунок подписывается снизу (выравнивание по середине) и нумеруется сквозным образом (1, 2, 3…). Пример: Рисунок 1 – Этой мой рисунок.

Вставьте в Ваш документ описание лабораторной работы или любой другой текст не более двух страниц и задайте вставленному тексту двухколончатую структуру, используя разделы, остальной текст должен остаться одноколончатым. В тексте на странице выделите первые две буквы и сделайте их красными и больше по размеру и полужирным шрифтом.

Воспользовавшись меню Файл-Свойства посмотрите статистику документа и подсчитайте средний размер слова (среднее количество букв в слове), среднее число слов на странице. Оформите данные о документе в виде таблицы. Для этого можно выбрать в Меню Таблица и подменю добавить|удалить строки, ячейки, столбцы. Попробуйте преобразовать таблицу в текст и наоборот и объединить таблицы. Таблица подписывается сверху таблицы, выравнивание по ширине или слева. Пример: Таблица 1 – Это моя таблица.

Запишите формулу расчета средней длины слова и среднего количества слов на странице в буквенном виде, представив число слов как параметр и длины слов в виде вектора, воспользовавшись редактором формул, запишите переменные, и что они обозначают, в соответствии с оформлением по ГОСТу. Запишите несколько тригонометрических формул для примера. Укажите значения переменных, пронумеруйте формулы. Формулы нумеруются справа с краю в скобочках. Например.

$$A= B+2 \qquad (2)$$

Укажите литературу и использованные источники, в качестве использованных источников могут выступать ссылки на Интернет ресурсы.

Оформите в соответствии с правилами ГОСТа рисунки и таблицы. Весь текст должен иметь один шрифт (размер, наклон, цвет - черный), одни и те же параметры абзаца, отступ красной строки 1 или 1.25 см. Шрифт 14 или 12, междустрочный интервал полуторный. Параметры страницы: (слева 3 см и справа 1 см, сверху и снизу 2 см).

Создайте оглавление, используя Вставка-Оглавление и указатели. Вставьте номера страниц. При выполнении старайтесь оформлять документ в соответствии с ГОСТом, старайтесь придерживаться одного строгого стиля, или заранее создать свой стиль.

Титульный лист приведен на следующей странице.

Приложение А. Оформление титульного листа лабораторной работы

Министерство образования и науки
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

Лабораторная работа №1

по курсу «Информатика»

Название лабораторной работы

Выполнил:
Студент 1-го курса
гр. номер
_____Фамилия И.О.
подпись

_____
дата

Принял:
к.т.н., доцент
_____ _____Фамилия И.О.
Оценка подпись

_____
дата

Томск 2018

2 Изучение макросов LibreOffice Writer

Иногда требуется провести работу с текстом или обработать текст каким-либо сложным образом и обычных средств, предоставляемых интерфейсом для этого не хватает. Кроме того, бывает необходимо провести одну и ту же последовательность рутинных действий, что порой занимает много времени, а хотелось бы свести последовательность этих действий к одному нажатию кнопки. Writer позволяет создавать специальные макросы, являющиеся по сути процедурами обработки текста, написанные на языке программирования, в нашем случае в качестве языка программирования выступает язык Бэйсик. При этом, обладая множеством всех стандартных операторов присущих языкам программирования высокого уровня, возможно получить доступ к объектам текстового редактора Writer, открытым документам, функциям открытия документов, всем объектам данного документа включая рисунки, параграфы, колонтитулы, выделенный текст, списки, слова, буквы, шрифты и т.д..

2.1 Объекты и классы.

Что же такое Объект. С точки зрения реального мира объект это нечто материальное, существующее и обладающее свойствами и поведением в реальном мире, часто объекты имеют какие-то общие свойства, благодаря которым мы относим каждый объект к какому-то классу объектов. Например – автомобиль, есть разные конкретные реализации и объекты автомобили, но общим классом является автомобиль, имеющий четыре колеса, способный ездить и управляемый водителем. То же самое можно сказать и о тексте или документе, документ это объект, который содержит объект текст, а объект текст содержит объекты слова, абзацы, буквы, текст редактируется, меняется, отображается, документ создается и сохраняется. Все эти действия мы выполняем с данными объектами, используя функции редактора, но мы можем эти функции вызвать с помощью алгоритмического языка.

Если вы уже работали с языками программирования и писали программы, то знаете что в любом языке программирования существует набор операторов или инструкций с помощью которых можно записать указания или программу, которую сможет понять и выполнить процессор. Существует набор стандартных операторов с помощью которых можно написать практически любой алгоритм, любой сложности – это ветвление, цикл, линейная последовательность выполнения операторов, арифметические действия и возможность обращения к переменным и записи в них каких то значений или результатов логических или арифметических выражений. Обычно в языках высокого уровня стараются избежать сложной работы с памятью присущей машинным языкам, вводится стандартная операция присвоения, которая позволяет некоторой – переменной – символьной последовательности присвоить какое-

то значение. Грубо говоря, используя эту символьную последовательность в выражениях вы работаете с тем, что содержится в данной переменной как в ящичке. Можно операцию присвоения описать таким образом: В стакан с названием – St1, мы заливаем молоко из кружки Cr1 и говорим, налейте мне молоко в St1 из Cr1. Таким образом, вам нальется то, что содержится в Cr1. То же самое и с переменной. Допустим, Val1 = 20; Val2 = 30; Val3 = Val1+Val2; тогда Val3 будет содержать значение 50. Вы знаете, что переменная может содержать в себе только данные определенного вида, а не все подряд (хотя существуют специальный вариантный тип данных, когда тип переменной можно определить в процессе выполнения программы). Ведь мы можем хранить и названия (строки) и числа, и объекты. Поэтому каждой переменной ставится в соответствие какой-то тип данных, или домен или область определения тех значений, которые она может принимать. Обычно в языках программирования – это целочисленные типы, вещественные, строковые, символьные, логические, перечислимые, множества, комплексные числа, тип запись или структуры. Так что же такое переменная объект, это некая ссылка на объект, являющийся сложной структурой данных, которая ко всему прочему может содержать методы работы с этими данными и объектом, а также защищать данные и ограничивать или разрешать к ним доступ.

Переменная-объект, это переменная, которая содержит в себе другие объекты, свойства и действия, производимые над объектом, объект является конкретной реализацией, какого-то класса (класс есть описание, некого множества объектов с одними и теми же свойствами). Обычно доступ к свойствам и функциям сложных типов данных (таких как классы) осуществляется путем написания имени переменной объекта, а затем через точку имени функции и или свойства данного объекта.

2.2 Переменные и объекты в Basic

Для объявления переменной указывается ключевое слово dim и затем список переменных через запятую, слово as и тип переменной.

Примеры:

Dim a,b as integer – объявление переменной целого типа.

Dim s as string – объявление переменной строкового типа.

Dim mass() as integer – объявление динамического одномерного массива целого типа.

Redim mass(100) – изменение длины массива и установка ее равной 100.

Dim desk as com.sun.star.frame.Desktop — переменная типа desktop унифицированной сетевой модели UNO, данная переменная может ссылаться на объекты типа Desktop.

В языке Basic можно обращаться к переменным представляющим собой ссылки на объекты, это могут быть объекты текст, параграфы, таблицы, отображаемые на экране окна,

они обладают набором свойств и методов работы с данными объектами. Объектная модель может быть любой, как и ее реализация, например в пакете Microsoft Office реализована своя объектная модель, в пакете LibreOffice или OpenOffice своя, потому объекты и способ взаимодействия с этим объектами в этих различных пакетах отличаются.

2.3 Операторы Basic

Оператор цикла For.

```
For index=n1 to n2 [step s]
Rem тело цикла
Next index
```

Переменная Index пробегает значения от n1 до n2 с инкрементацией s (увеличение на s), в данном случае s может быть переменной или константой целого типа, квадратные скобочки указывают на то, что конструкция является не обязательной, в случае если она не указывается то шаг равен 1.

Например,

```
val =0
For xyz = 4 to 50 step 4
val=val+xyz
next xyz
```

Алгоритм вычисляет сумму значений от 4 до 50 с шагом 4, то есть сумму 4, 8, 12, 16 ... до 48 в переменную val.

```
val1 =0
For aval = 1 to 50
val1=val1+aval
next aval
```

В данном случае рассчитывается сумма целых чисел от 1 до 50.

Оператор цикла While, делай пока выполняется условие. Операторы внутри цикла повторяются до тех пор пока выполняется условие.

```
While <условие>
операторы
Wend
```

Пример:

```
While i<N
```

I=i+1

wend

Цикл выполняется пока переменная i меньше N.

Условный оператор If,

if <условие> then

<последовательность операторов если условие выполняется>

[else

<последовательность операторов в случае невыполнения условия>]

end if

Пример: если I меньше 100 (если условие выполнено) то увеличить I на 1, иначе уменьшить на 1.

If i<100 then

i=i+1

else

i=i-1

end if

2.4 Процедуры и функции.

Функции и процедуры представляют собой отдельные блоки операторов, которые могут быть вызваны в основной программе или подпрограмме, обычно вызов функции или процедуры осуществляется в программе с помощью указания ее имени и передаваемых в нее параметров, после выполнения операторов функции управление возвращается программе или подпрограмме вызвавшей ее и начинается выполнение операторов следующих за функцией или процедурой. Очевидно, что назначение процедур и функций в том, чтобы не писать каждый раз один и тот же код для часто повторяющихся операций выполняющих определенное логически завершенное действие. При этом внутри функций и процедур возможно использовать свои локальные переменные, которые могут иметь те же названия, что и переменные в других процедурах и функциях и в основной программе. При этом извне процедуры мы не можем менять локальные переменные функции. Типичное использование процедур и функций заключается, в том что мы передаем в функцию какие то значения, на основе которых эта функция производить ряд действий и вычисление какого-то результата. Основное отличие процедур от функций в том, что имени функции ассоциирован какой то тип возвращаемых данных, грубо говоря, функцию можно использовать в выражениях, например, арифметических или в логических, в условных операторах и циклах. Процедура вызывается вне какого-либо выражения.

Примеры.

Функция возвращает сумму двух чисел, передаваемых как фактические параметры в функция из внешней программы

```
Function sum(a,b as integer) as integer
Sum=a+b
End function
```

Использование функции sum в программе.

```
Dim x as integer
x = 2
x=x+sum(x,4)*2
```

Пример процедуры позволяющей сложить два числа, значение возвращается в формальном параметре c, при вызове процедуры не должно быть константой, а должно быть переменной типа integer

```
Sub sum(a,b,c as integer)
c = a+b
End sub
Dim c as integer
Call sum(2,2,c)
```

2.5 Создание макроса в LibreOffice

Для создания макроса в LibreOffice выбираем сервис+макросы+управление макросами+LibreOffice Basic (Tools+Macros+Organize Macros). При этом отобразится окно представленное на рисунке ниже (рисунок 2.1). Для того, чтобы макрос был сохранен в самом документе, необходимо выбрать ваш документ, выбрать набор стандартных модулей «standard» и затем нажать «создать», затем необходимо ввести имя модуля. После создания модуля можно его выбрать, в окошке справа выбрать макрос Main и нажать редактировать (Edit). Либо необходимо после создания модуля (Module1), написать в поле Macro Name (Имя макроса) новое имя макроса и нажать создать (рисунок 2.2).

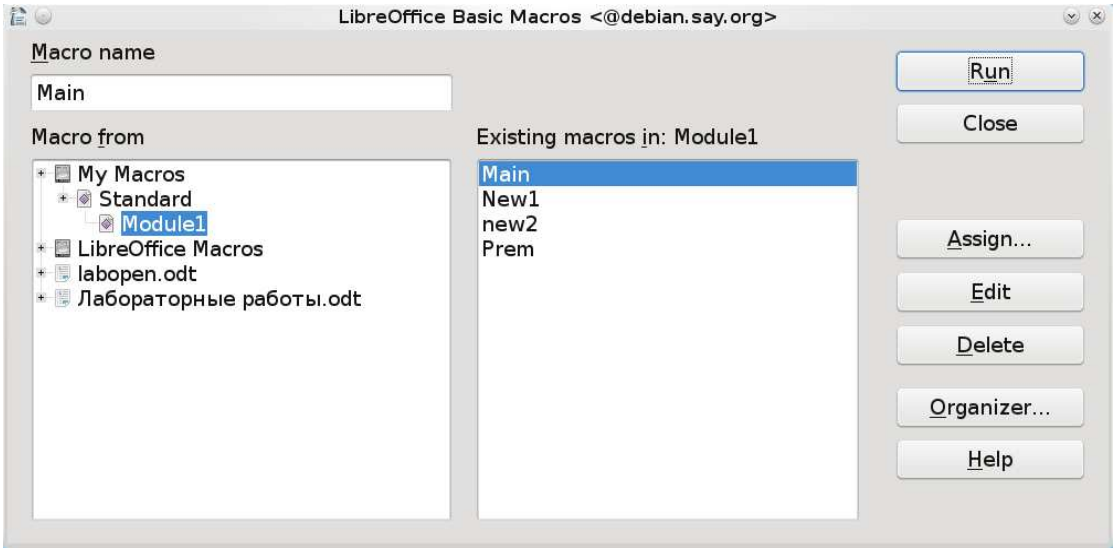


Рисунок 2.1 - Окно создания и редактирования макросов

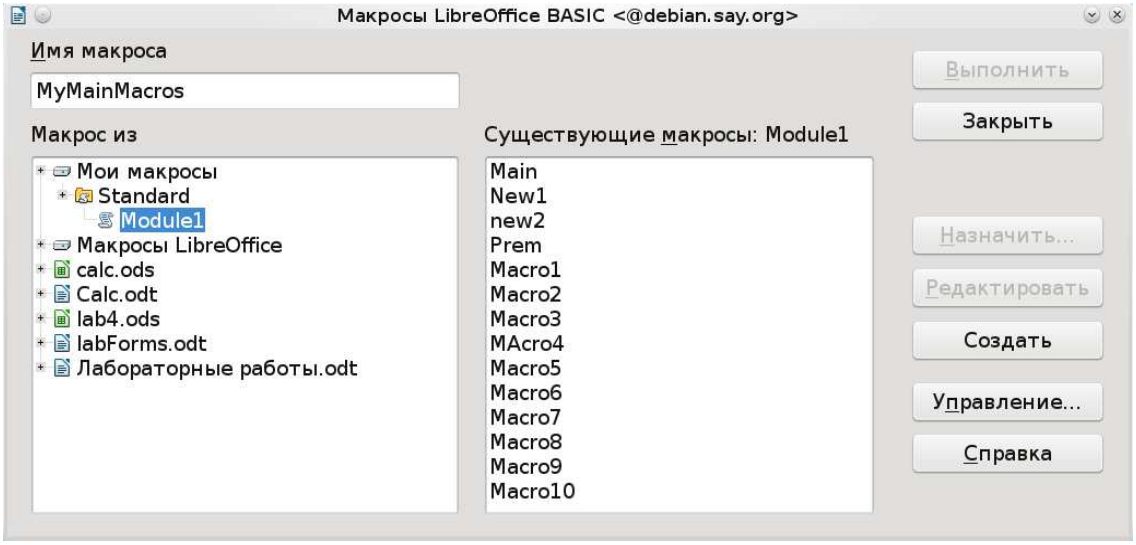


Рисунок 2.2 - Пример создания нового макроса MyMainMacros

В результате создания и редактирования макроса появляется окно редактора Бэйсика, на рисунке приведен пример с двумя макросами, естественно их может быть больше и они могут иметь входные параметры.

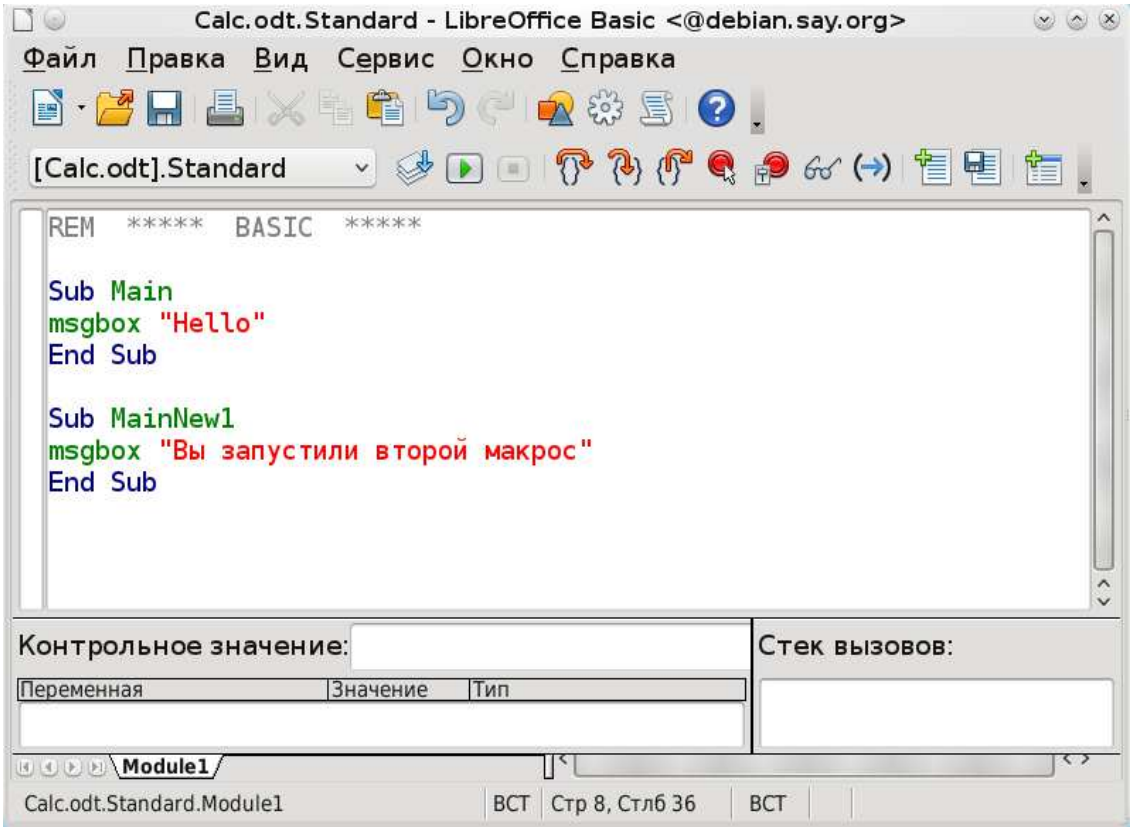


Рисунок 2.3 - Редактор Basic и два макроса

В LibreOffice, как уже отмечалось, объектная модель несколько другая нежели в Microsoft Office, в LibreOffice Basic используется так называемся унифицированная сетевая объектная модель UNO. Ниже приведен пример макроса openoffice увеличивающий размер шрифта каждого параграфа.

```
Sub Main
Dim Doc As Object
Dim Enum As Object
Dim TextElement As Object
' StarDesktop — главный объект доступный из макроса
' создание ссылки на объект document, текущий документ
Doc = StarDesktop.CurrentComponent
' создание объекта enumeration
Enum = Doc.Text.createEnumeration
' цикл по всем текстовым элементам
While Enum.hasMoreElements
TextElement = Enum.nextElement
' проверка является ли текущий блок таблицей
```

```
If TextElement.supportsService("com.sun.star.text.TextTable") Then
MsgBox "Текущий блок содержит таблицу"
End If
' проверка является ли текущий блок текста параграфом
If TextElement.supportsService("com.sun.star.text.Paragraph") Then
TextElement.CharHeight = 20
End If
Wend
' окошко с сообщением
msgbox "Конец макроса"
End Sub
```

Объект enumeration позволяет перебрать все элементы текста в цикле пока эти элементы не закончатся. Операция Enum.nextElement возвращает текущий элемент и переходит к следующему, где переменная Enum есть ссылка на объект созданный вызовом функции Doc.Text.createEnumeration. Очевидна иерархия объектов, в документе Doc есть свойство Text ссылающееся на текстовый объект документа, далее вызывается метод этого объекта createEnumeration, данный метод представляет собой функцию создающую объект Enumeration, при этом сам объект Doc является ссылкой на объект CurrentComponent объекта StarDesktop. В объекте на который ссылается Doc может быть не только свойство Text, но и другие, например, имя файла документа, активность данного документа и т. д., кроме, того в объекте могут быть и методы, например, закрыть документ, открыть, сделать активным, сохранить. StarDesktop представляет собой объект управления всеми текущими открытыми приложениями LibreOffice, CurrentComponent — является текущим активным документов LibreOffice, в общем случае это может быть ссылка и на документ Writer и на документ Calc (электронная таблица) и на документ Base (база данных). Далее для каждого параграфа, предварительно проверив, а действительно ли объект TextElement — параграф, устанавливаются свойства текста: TextElement.CharHeight = 20, что задает размера шрифта абзаца = 20.

2.6 Задания Макросы LibreOffice Writer.

1) Написать макросы для очистки формата всего текста документа, выделенного текста, первого абзаца.

Подсказка. Перебрать все параграфы и задать одни и те же общие свойства текста параграфа. Ниже приведен список свойств текста.

• CharFontName (String) – имя выбранного типа шрифта;

Например, TextElement.CharFontName = "Free Times".

• CharColor (Long) – цвет текста;

Например, TextElement.CharColor = RGB(0,255,0) — зеленый цвет

• CharHeight (Float) – высота символа в пунктах (pt);

• CharUnderline (Constant group) – тип подчеркивания (константы в соответствии с com.sun.star.awt.FontUnderline);

Например, TextElement.CharUnderline = com.sun.star.awt.FontUnderline.WAVE

• CharWeight (Constant group) – вес шрифта (константы в соответствии с com.sun.star.awt.FontWeight);

• CharBackColor (Long) – фоновый цвет;

• CharKeepTogether (Boolean) – подавление автоматического разрыва строк;

• CharStyleName (String) – имя стиля символа.

Курсив устанавливается в свойстве шрифта CharPosture, присвоив данной переменной значения 2 TextElement.CharPosture = 2 или TextElement.CharPosture = com.sun.star.awt.FontSlant.ITALIC. Кроме italic в FontSlant есть свойства NONE, OBLIQUE, ITALIC DONTKNOW, REVERSE_OBLIQUE, REVERSE_ITALIC.

Также свойства абзаца.

• ParaAdjust (enum) – вертикальная ориентация текста (константы в соответствии с com.sun.star.style.ParagraphAdjust);

• ParaLineSpacing (struct) – межстрочный интервал (структура в соответствии с com.sun.star.style.LineSpacing); Константы LineSpacingMode PROP - высота пропорциональна, MINIMUM — минимальная высота строки, LEADING — расстояние до предыдущей линии, FIX — фиксированная высота строки.

Например,

```
v = TextElement.ParaLineSpacing
v.Mode = com.sun.star.style.LineSpacingMode.FIX
v.Height = 300
TextElement.ParaLineSpacing =  v
```

• ParaBackColor (Long) – фоновый цвет;

• ParaLeftMargin (Long) – левое поле в сотых долях миллиметра;

• ParaRightMargin (Long) – правое поле в сотых долях миллиметра;

• ParaTopMargin (Long) – верхнее поле в сотых долях миллиметра;

• ParaBottomMargin (Long) – нижнее поле в сотых долях миллиметра;

• ParaTabStops (Array of struct) – тип и положение позиций табуляции (массив структур типа com.sun.star.style.TabStop);

• ParaStyleName (String) – имя стиля абзаца.

2) Написать макрос для изменения стиля каждой первой и пятой буквы каждого абзаца на курсив, изменить цвет буквы в активном документе и выделенном тексте.

Ниже приведен макрос, который используется для того, чтобы каждое первое слово предложения было выделено жирным шрифтом. При этом создается специальный объект текстовый Cursor, который позволяет осуществлять навигацию по документу, выделяя нужный текст или область нужного текста. Мы можем устанавливать все свойства символов для данного выделенного текста, при этом нужно помнить, что текст не является выделенным в понимании - выделение для копирования, а просто это область текста или блок текста свойства, которого будут меняться если мы будем присваивать какие-то значения свойствам объекта Cursor.

```
Sub Main
Dim Doc As Object
Dim Cursor As Object
Dim Proceed As Boolean
Doc = StarDesktop.CurrentComponent
'создать объект — текстовый курсор для текущего открытого документа
Cursor = Doc.Text.createTextCursor()
' начать цикл
Do
' перейти к концу слова с выделением от текущего положения курсора
Cursor.gotoEndOfWord(True)
'изменить шрифт отмеченного текста на жирный
Cursor.CharWeight = com.sun.star.awt.FontWeight.BOLD
'перейти на следующее предложение без выделения, в переменную proceed занести значение возвращаемой функцией  gotoNextSentence, она вернет True если следующее предложение есть и False если нет, т.е. фактически когда завершится текст функцяи вернет значение False
Proceed = Cursor.gotoNextSentence(False)
'перейти на начало текущего предложения без выделения
Cursor.gotoStartOfSentence(False)
'условие выполнения цикла, пока логическая переменная Proceed истинна, то-есть True
Loop While Proceed
```

```
msgbox "end"
End Sub
```

Ниже перечислены возможные способы навигации.

- goLeft (Count, Expand) – переход на Count символов влево;
- goRight (Count, Expand) – переход на Count символов вправо;
- gotoStart (Expand) – переход к началу текстового документа;
- gotoEnd (Expand) – переход к концу текстового документа;
- gotoRange (TextRange, Expand) – переход к указанному TextRange-объекту;
- gotoStartOfWord (Expand) – переход к началу текущего слова;
- gotoEndOfWord (Expand) – переход к концу текущего слова;
- gotoNextWord (Expand) – переход к началу следующего слова;
- gotoPreviousWord (Expand) – переход к началу предыдущего слова;
- isStartOfWord () – возвращает True, если TextCursor в начале слова;
- isEndOfWord () – возвращает True, если TextCursor в конце слова;
- gotoStartOfSentence (Expand) – переход к началу текущего предложения;
- gotoEndOfSentence (Expand) – переход к концу текущего предложения;
- gotoNextSentence (Expand) – переход к началу следующего предложения;
- gotoPreviousSentence (Expand) – переход к началу предыдущего предложения;
- isStartOfSentence () – возвращает True, если TextCursor в начале предложения;
- isEndOfSentence () – возвращает True, если TextCursor в конце предложения;
- gotoStartOfParagraph (Expand) – переход к началу текущего абзаца;
- gotoEndOfParagraph (Expand) – переход к концу текущего абзаца;
- gotoNextParagraph (Expand) – переход к началу следующего абзаца;
- gotoPreviousParagraph (Expand) – переход к началу предыдущего абзаца;
- isStartOfParagraph () – возвращает True, если TextCursor в начале абзаца;
- isEndOfParagraph () – возвращает True, если TextCursor в конце абзаца.

Входной параметр Expand показывает выделяется ли текст при передвижении курсора, значение True — выделяется (отмечается) и False — курсор продвигается и текст не выделяется (не отмечается). Каждая функция при этом возвращает значение true, либо false в зависимости от успешности выполнения, например, если следующего параграфа нет при вызове функции перейти на следующий параграф, то вернется значение false.

Подсказка для выполнения задания:

Использовать gotoStartOfParagraph, .goRight, CharColor , gotoNextParagraph.

3) Написать макрос для поиска в тексте запятых и замены их на троеточие.

Ниже приведен пример макроса, который позволяет заменить одни слова на другие, используя свойство параграфа textportion, это текст являющийся частью параграфа, которому присущи свои собственные свойства и характеристики.

```
Sub Main
Dim Doc As Object
Dim Enum1 As Object
Dim Enum2 As Object
Dim TextElement As Object
Dim TextPortion As Object
Doc = StarDesktop.CurrentComponent
Enum1 = Doc.Text.createEnumeration
' цикл по всем абзацам
While Enum1.hasMoreElements
TextElement = Enum1.nextElement
'проверка является ли текстовый элемент параграфом
If TextElement.supportsService("com.sun.star.text.Paragraph") Then
Enum2 = TextElement.createEnumeration
' цикл по всем элементам текущего параграфа (текстовым порциям) TextElement, пока есть еще элементы
While Enum2.hasMoreElements
TextPortion = Enum2.nextElement
' взять строку текста порции текста, произвести замену одной последовательности символов на другую, эту замену возвратит функция replace, произвести присвоение строке содержащейся в порции новой строки возвращенной функцией  replace.
TextPortion.String = Replace(TextPortion.String, "you", "U")
TextPortion.String = Replace(TextPortion.String, "o", "2")
Wend
End If
Wend
msgbox s
End Sub
```

4) Сделать макрос для обмена двух абзацев местами.

Можно воспользоваться свойством параграфа TextElement.String и поменять текст в двух параграфах местами. Заведите две переменные TextElement, присвойте им значение первого и последующего параграфа с помощью Enum.nextElement и используя третью строковую переменную произведите обмен.

5) Изменить цвет и размер шрифта каждого абзаца на произвольный. Использовать функцию rnd(), которая возвращает случайное вещественное значение от 0 до 1 и функцию rgb(r,g,b), которая возвращает преобразованное цветовое значение из последовательности интенсивностей красного, зеленого и синего, где каждое значение интенсивности варьируется от 0 до 255. Комбинация интенсивностей основных трех цветов создает для человека иллюзию различных цветов, например, три интенсивности со значениями 255, будет давать белый цвет, все значения 140, дают серый цвет.

6) Изменить цвет и размер шрифта каждого третьего слова в тексте.

Ниже приведен пример макроса, который меняет цвет каждой первой буквы параграфа на красный цвет.

```
Sub Main
Dim Doc As Object
Dim Cursor As Object
Dim Proceed As Boolean
Doc = StarDesktop.CurrentComponent
Cursor = Doc.Text.createTextCursor()
Do
Cursor.gotoStartOfParagraph(False)
Cursor.goRight(1,True)
Cursor.CharColor =  RGB(255,0,0)
Proceed = Cursor.gotoNextParagraph(False)
Loop While Proceed
msgbox "end"
End Sub
```

7) Написать макрос, в котором каждая запятая будет заменена на последовательность трех разноцветных точек. Подсказка: использовать объект Cursor и его свойство String, данное свойство содержит текущую выделенную строку, если использовались функции движения по

тексту со значением True, можно выделить одну букву, сравнить ее с запятой и заменить на три точки. Не забывайте сбрасывать выделение с помощью движения со значением False.

8) Выполнить индивидуальный макрос по вариантам ниже.

Задания. Макросы Writer.

1. В каждом втором слове каждого второго параграфа заменить первую половину слова на вторую. Цвет для первой половины задать инвертированным по красной составляющей второго слова. Все буквы слова сделать заглавными.

2. В каждом втором предложении поменять местами первое и последнее слово. Задать цвет этих слов в зависимости от количества букв (умножив соответственно на 20 и проверив на допустимость значения).

3. Перетасовать параграфы текста в случайном порядке. Порядок предварительно задать в массиве, вывести нумерацию в сообщении.

4. Удалить слова содержащие букву «а». Слова содержащие «о» покрасить в красный цвет.

5. Слова стоящие в квадратных скобочках заменить на номер слова в скобочках окрашенный в красный цвет. Скобочки оставить.

6. Слова стоящие в квадратных скобочках заменить на длину этого слова, окрасив в зеленый цвет. Скобочки оставить.

7. Слова стоящие в квадратных скобочках заменить на номер параграфа в котором они стоят. Окрасить в зеленый цвет.

8. В словах с двумя буквами «а» и более сделать замену на букву «Л». Окрасить в желтый цвет.

9. В словах с двумя буквами «а» и более сделать все буквы заглавными.

10. В каждом втором слове каждого третьего предложения поменять буквы «а» на символ «@» если буква «а» есть, если ее нет, то все слово заменить на символ «@».

11. В каждом третьем слове поменять местами первую и вторую буквы, а букву в центре слова или две буквы в центре слова в зависимости от четности сделать заглавными и синими.

12. В каждом параграфе сделать по середине вставку количества символов в параграфе, окрасив эту вставку в разные цвета.

13. Удалить в каждом третьем слове букву «а», в каждом втором слове заменить букву «е» на символ «*». Перед символом «*» окрасить символ в любой цвет и увеличить.

14. Пронумеровать все слова в тексте, вставив перед словом его номер. Номер сделать случайным цветом.

15. В тексте все запятые заменить на символы «<>». Внутри <> указать номер запятой. Цвет скобочек сделать случайным.

16. В каждом втором слове в котором встречается символ «а» удалить вторую половину слова. Первую половину слова разделить пополам, вставив пробел.

17. В каждом слове с менее чем пятью буквами, все буквы покрасить в случайные цвета.

18. В каждом втором слове первую половину покрасить в случайные цвета и случайно с вероятностью 0.5 сделать заглавной букву или жирной.

19. В каждой нечетной букве каждого второго слова сделать замену на номер нечетной буквы в слове. Каждый нечетный номер должен быть покрашен сначала в красный, потом в синий цвет.

20. Все слова с более чем пятью согласными необходимо покрасить в красный цвет.

3 Изучение электронных таблиц LibreOffice Calc

Цель работы.

Научиться пользоваться основными функциями в Calc.

3.1 Общие сведения об электронной таблице Calc пакета LibreOffice.

Calc относится к классу систем обработки числовой информации, называемых spreadsheet. Буквальный перевод термина “spreadsheet” с английского языка означает “расстеленный лист (бумаги)”. В компьютерном мире под этим термином подразумевают класс программных средств, именуемых у нас “электронными таблицами”. Ниже на рисунке приведено главное окно Calc.

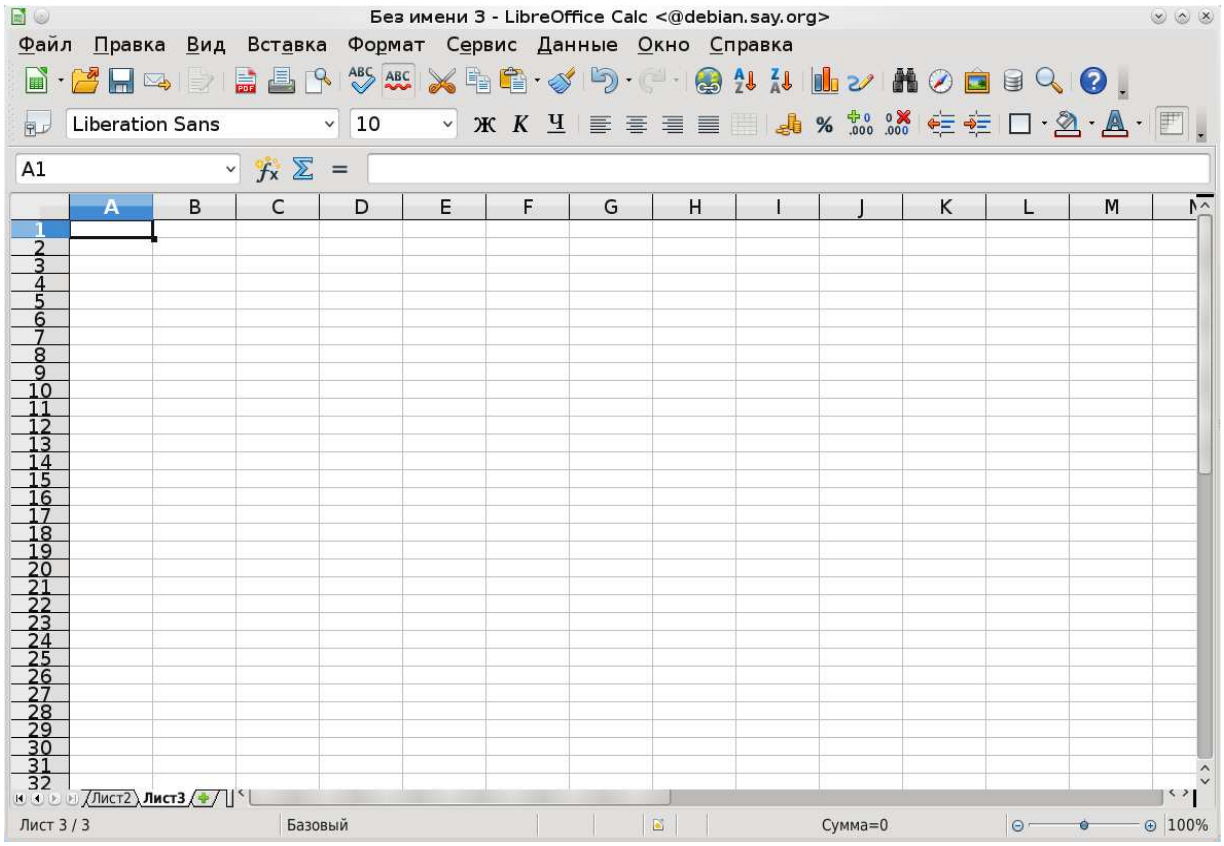


Рисунок 3.1 - Главное рабочее окно LibreOffice Calc

Области применения электронных таблиц:

- бухгалтерский и банковский учет;
- планирование распределение ресурсов;
- проектно-сметные работы;
- инженерно-технические расчеты;
- обработка больших массивов информации;

- исследование динамических процессов.

Основные возможности электронных таблиц:

- анализ и моделирование на основе выполнения вычислений и обработки данных;

- оформление таблиц, отчетов;

- форматирование содержащихся в таблице данных;

- построение диаграмм требуемого вида;

- создание и ведение баз данных с возможностью выбора записей по заданному критерию и сортировки по любому параметру;

- перенесение (вставка) в таблицу информации из документов, созданных в других приложениях, работающих в среде Windows;

- печать итогового документа целиком или частично.

Преимущества использования ЭТ при решении задач.

- Решение задач с помощью электронных таблиц освобождает от составления алгоритма и отладки программы. Нужно только определенным образом записать в таблицу исходные данные и математические соотношения, входящие в модель.

- При использовании однотипных формул нет необходимости вводить их многократно, можно скопировать формулу в нужную ячейку. При этом произойдет автоматический пересчет относительных адресов, встречающихся в формуле. Если же необходимо, чтобы при копировании формулы ссылка на какую-то ячейку не изменилась, то существует возможность задания абсолютного (неизменяемого) адреса ячейки.

3.2 Структура электронной таблицы

В таблице используются столбцы и строки.

Строки пронумерованы от 1 до 16384, столбцы помечаются латинскими буквами от A до Z, и комбинациями букв AA, AB,..., IV,

Элемент, находящийся на пересечении столбца и строки называется - ячейкой (клеткой).

Прямоугольная область таблицы называется диапазоном (интервалом, блоком) ячеек. Она задается адресами верхней левой и правой нижней ячеек блока, перечисленными через двоеточие.

Модель ячейки в Calc

Каждая ячейка таблицы имеет следующие характеристики:

- адрес;

- содержимое;

- изображение;

- формат;

- имя;

- примечание (комментарий).

Адрес ячейки - номер столбца и строки. Используется в формулах в виде относительной, абсолютной или смешанной ссылки, а также для быстрого перемещения по таблице.

Calc позволяет использовать стиль ссылок A1.

Например. Пусть в ячейке D3 нужно получить произведение чисел, находящихся в ячейках A2 (второй ряд, первая колонка) и B1 (первый ряд, вторая колонка). Это может быть записано одним из следующих способов:

Адресация указывается как буква обозначающая столбец и цифра обозначающая номер строки.

=A2 * B1

Имя столбца, имя строки, которые будут относительно изменяться, при копировании формулы в другую ячейку.

Смещение по строке, смещение по столбцу, относительно ссылающейся ячейки. Сама формула при копировании не изменяет вид, но ссылается уже на другие ячейки.

Абсолютный вид ссылок

=$A$2 * $B$1

имя столбца, имя строки, которые останутся неизменным, при копировании формулы.

Смешанный вид ссылок

=$A2 * B$1

=A$2 * $B1

Таким образом если перед адресом строки или столбца стоит знак доллара $ это обозначает абсолютную адресацию соответствующей координаты и при копировании она никак не меняется, если же знак доллара не стоит то адрес в формуле будет изменен относительно копируемого адреса на число ячеек по вертикали или по горизонтали равное смещению относительно предыдущей ячейки где находилась копируемая формула.

Содержимым ячейки может быть:

- число (целое со знаком или без (-345), дробное с фиксированной точкой (253,62) или с плавающей точкой (2,5362e+2));

- текст;

- формула.

Формула - всегда начинается со знака “=“ и может содержать: числовые константы, абсолютные или относительные ссылки на адреса ячеек, встроенные функции.

Аргументы функций всегда заключаются в круглые скобки. Стандартные функции можно как ввести с клавиатуры, так и воспользоваться меню Вставка/Функция или соответствующей кнопкой на панели инструментов.

Изображение - то, что пользователь видит на экране монитора.

Если содержимым ячейки является формула, то изображением будет ее значение.

Текст, помещенный в ячейку, может быть “виден” целиком, либо (если соседняя ячейка не пуста), из него видно столько символов, сколько позволяет ширина ячейки.

Изображение числа зависит от выбранного формата. Одно и то же число в разных форматах (дата, процент, денежный и т.д.) будет иметь различное изображение.

Формат ячейки - формат чисел, шрифт, цвет символов, вид рамки, цвет фона, выравнивание по границам ячейки, защита ячейки.

Имя - используется в формулах, как замена абсолютного адреса ячейки. Например, назначив ячейке C3 имя “Произведение” в ячейку D3 можно поместить формулу: =Произведение/3 (вместо формулы =C3/3). В этом случае, при копировании формулы, адрес ячейки меняться не будет.

Примечание - сопроводительный текст к содержимому ячейки. Ввести примечание в ячейку можно с помощью меню Вставка / Примечание. Ячейка, имеющая примечание, отмечается в рабочем листе точкой в правом верхнем углу.

Основными объектами, над которыми производятся действия в электронных таблицах, являются ячейки и диапазоны ячеек (блоки).

Блок - любая прямоугольная область таблицы, в минимальном случае - одна ячейка. Адрес блока задается так: адрес верхней левой ячейки блока, двоеточие, адрес правой нижней ячейки блока.

Примеры блоков: A1 (ячейка); A1:A9 (столбец); B2:Z2 (строка); B2:D4 (прямоугольная область).

Неотъемлемым элементом рабочего поля таблицы является курсор. В ЭТ термин “курсор” используется в следующих случаях:

- курсор ЭТ - жирная рамка вокруг текущей ячейки, перемещается с помощью клавиш управления курсором;

- текстовый курсор - мигающая (или не мигающая) черточка, отмечающая положение текущего символа при редактировании содержимого ячейки.

Для ввода данных можно произвести следующие действия:

1. Установить курсор ЭТ в ячейку, в которой должны быть размещены данные.

2. Набрать данные.

3. Для завершения ввода нажать клавишу <Enter> (при этом курсор ЭТ переместится на строку ниже), либо нажать «зеленую галочку» на панели инструментов (при этом курсор останется в текущей ячейке).

В ячейке могут размещаться данные одного из следующих типов:

1. число

2. формула

3. текст

Текст можно вводить произвольной формы, но если он начинается со знака “=“, то перед ним следует поставить апостроф, чтобы он не воспринимался как формула.

Числа также вводятся в привычном виде. Следует только помнить, что дробные десятичные числа записываются через запятую: 3,5; -0,0045, либо через точку: 3.5; -0.0045, в зависимости от установленных параметров. Изменение вида разделителя целой и дробной части производится в меню Сервис/ Параметры/ Международные.

По умолчанию текстовые поля в Calc выводятся в одну строку. Для того чтобы текст переносился в ячейке в несколько строк:

1. Выделите ячейки, для которых необходимо разрешить перенос текста.

2. Выберите пункт меню Формат/ Ячейки вкладка Выравнивание.

3. Поставьте галочку в опции Переносить по словам.

Для таблиц со сложной структурой используйте объединение ячеек, но только там, где это действительно требуется.

Для ввода формул можно воспользоваться следующей последовательностью действий:

1. Убедитесь в том, что активна (выделена курсивной рамкой) та ячейка, в которой вы хотите получить результат вычислений.

2. Ввод формулы начинается со знака “=”. Этот знак вводится с клавиатуры.

3. После ввода знака “=” Calc переходит в режим ввода формулы. В этом режиме, при выделении какой-либо ячейки, ее адрес автоматически заносится в формулу. Это позволяет избавить пользователя от необходимости знать адреса ячеек и вводить их в формулу с клавиатуры.

4. Находясь в режиме ввода формулы, вы последовательно указываете левой кнопкой мыши на ячейки, хранящие некие числовые значения, и вводите с клавиатуры знаки операций между исходными значениями.

§ Знаки операций должны вводиться между адресами ячеек.

§ Удобнее вводить знаки операций с правого цифрового блока клавиатуры. Чтобы этот блок работал в нужном режиме, индикатор <Num Lock> должен быть включен.

5. Чтобы результат вычислений появился в активной ячейке, необходимо выйти из режима ввода формулы.

§ <Enter> завершает ввод формулы, и переводит курсор в следующую ячейку.

§ “Зеленая галочка” на панели ввода формулы завершает ввод формулы, и оставляют курсор в той же ячейке.

Например, если в ячейке D2 должна помещаться разность чисел из ячеек B2 и C2, то после установки курсора на D5 следует указать мышью на B2, ввести с клавиатуры знак “-”, указать мышью на C2 и нажать <Enter> или “зеленую галочку”.

В формулах можно использовать числовые константы (-4,5), ссылки на блоки (D4), (A3:D8), знаки арифметических операций, встроенные функции (СУММ, МАКС, SIN и т.д.)

Возведение в степень ^

=3^2

Умножение *

=A8*C6

Деление /

=D4/N5

Сложение +

=B2+5

Вычитание -

=9-G6

Равно =

Меньше <

Больше >

Меньше или равно <=

Больше или равно >=

Не равно <>

Диапазон :

=СУММ(A1:C10), если какая то ячейка пустая в диапазоне, то автоматически считается, что она равна 0.

Объединение диапазонов ;

=СУММ(A1;A2;A6:D8)

Максимум

МАКС

=МАКС(A3:C5), если какие то ячейки пустые, но есть не пустые, то за максимум берется самое максимальное значение, если все пустые, то возвращается 0.

Минимум

МИН

=МИН(E2:P7)

Функция ЕСЛИ

=ЕСЛИ(A1=5;A2+A3;B2+100) – если A1=5 то сложить A2 и A3 иначе B2+100.

=ЕСЛИ(A1<5; ЕСЛИ(A1=4;A2+A3;A3+20);SIN(B2+100)) – вложенная функция ЕСЛИ и SIN. Если окажется что A1<5 то будет вычисляться функция ЕСЛИ с проверкой на равенство A1=4, иначе вычисляется функция SIN.

Функция среднего значения СРЗНАЧ:

СРЗНАЧ(A3:D7)

При вводе данных Вы можете ошибиться и должны уметь исправлять ошибки. Конечно, Вы можете просто ввести в ячейку с ошибочными данными новое правильное значение, но если исправить требуется один - два символа, то целесообразнее отредактировать содержимое ячейки.

Отредактировать данные Вы можете различными способами, но курсор ЭТ должен стоять на редактируемой ячейке.

1. Перейдите в режим редактирования содержимого ячейки. Это можно сделать одним из следующих способов:

- Щелкнете левой клавишей мыши в строке формул.
- Нажмите <F2>.
- Дважды щелкните мышью на ячейке.

2. Текстовый курсор поставьте перед неверным символом, исправьте данные.

3. Нажмите <Enter> или “зеленую галочку” на панели инструментов, чтобы выйти из режима редактирования.

Неверный формат ячейки может быть изменен только выбором другого формата в меню Формат / Ячейка.

Если ошибка допущена при вводе числа, то так как компьютер не знает, что это ошибка, Excel автоматически пытается подобрать подходящий для данного изображения формат.

Не пытайтесь исправить ошибку непосредственно в ячейке, вряд ли это удастся, так как скрытый формат этой ячейки уже сформирован. Поэтому нужно исправлять сначала формат ячейки на правильный с помощью меню Формат / Ячейка / Число.

Если при вводе формул Вы забыли поставить знак “=”, то все, что было набрано, запишется в ячейку как текст. Если Вы поставили знак равенства, то компьютер распознал, что идет ввод формулы и не допустит записать формулу с ошибкой до тех пор, пока она не будет исправлена.

Примеры ошибок:

#ИМЯ?

адрес ячейки введен с клавиатуры в режиме кириллицы

#ЗНАЧ!

в одной из ячеек, входящих в формулу, находится не числовое значение

Копирование ячеек.

В электронных таблицах часто требуется проводить операции не просто над двумя переменными (ячейками), но и над массивами (столбцами или строками) ячеек. Т.е. все формулы результирующего массива аналогичны и отличаются друг от друга только адресом строк или столбцов.

От проведения однотипных действий в каждой ячейки строки (или столбца) избавляет следующий прием копирования формулы:

1. Убедитесь, что активна (выделена курсорной рамкой) именно та ячейка, в которой находится предназначенная для копирования формула.

2. Не нажимая на кнопки мыши, подведите указатель мыши к нижнему правому углу курсорной рамки (этот угол специально выделен).

3. Отыщите положение, при котором указатель мыши превращается в тонкий черный крестик.

4. Нажмите на левую кнопку мыши и, удерживая ее, выделяйте диапазон ниже (при копировании по строкам) или правее (при копировании по столбцам) до тех пор, пока не выделятся все ячейки, в которые вы хотите скопировать данную формулу.

5. Отпустите левую кнопку мыши.

Одно из преимуществ электронных таблиц в том, что в формулах можно использовать не только конкретные числовые значения (константы), но переменные - ссылки на другие ячейки таблицы (адреса ячеек). В тот момент, когда Вы нажимаете клавишу <Enter>, в формулу вместо адреса ячейки подставляется число, находящееся в данный момент в указанной ячейке.

Другое достоинство в том, что при копировании формул входящие в них ссылки изменяются (относительная адресация).

Однако иногда при решении задач требуется, чтобы при копировании формулы ссылка на какую-либо ячейку не изменялась. Для этого используется абсолютная адресация, или абсолютные ссылки.

При копировании приведенным выше способом адреса ячеек в формуле изменялись относительно.

Если необходимо, чтобы при копировании или перемещении данных адрес какой-либо ячейки в формуле не мог изменяться (например, при умножении всего столбца данных на значение одной и той же ячейки), нужно зафиксировать положение этой ячейки в формуле до того, как вы будете копировать или перемещать данные.

Для фиксации адреса ячейки используется знак “$”.

Координата строки и координата столбца в адресе ячейки могут фиксироваться раздельно.

Чтобы относительный адрес ячейки в формуле стал абсолютным, после ввода в формулу адреса этой ячейки нажмите <F4>.

Например, при копировании формулы = $A4+$A5, находящейся в ячейке A2, в ячейку B3 получим в этой ячейке формулу =$A5+$A6, при копировании = A4+$A5, получим = B5+$A6, при копировании = A4+A$5, получим =B5+B$5, при копировании = $A$4+$A$5, получим = $A$4+$A$5. Копирование помогает избежать ввода однотипной формулы вручную для обработки целого столбца или строки однотипных данных каждого элемента строки или столбца. Варьирование меняющейся и фиксированной ссылки на ячейку позволяет управлять процессом организации формул расчета для групп данных в столбцах, строках и таблицах. Копирование можно осуществить с помощью мышки или используя клавиатуру - Ctrl-Ins, и вставку Shift-Ins.

3.3 Построение диаграмм

Одной из возможностей Calc является способность превращать абстрактные ряды и столбцы чисел в привлекательные, информативные графики и диаграммы. Calc поддерживает множество типов различных стандартных двух- и трехмерных диаграмм. При создании новой диаграммы по умолчанию в Calc установлена гистограмма.

Диаграммы - это удобное средство графического представления данных. Они позволяют оценить имеющиеся величины лучше, чем самое внимательное изучение каждой ячейки рабочего листа. Диаграмма может помочь обнаружить ошибку в данных.

Для того чтобы можно было построить диаграмму, необходимо иметь, по крайней мере, один ряд данных. Источником данных для диаграммы выступает таблица Calc.

Специальные термины, применяемые при построении диаграмм:

-Ось X называется осью категорий и значения, откладываемые на этой оси, называются категориями.

-Значения отображаемых в диаграмме функций и гистограмм составляют ряды данных. Ряд данных – последовательность числовых значений. При построении диаграммы могут использоваться несколько рядов данных. Все ряды должны иметь одну и туже размерность.

-Легенда – расшифровка обозначений рядов данных на диаграмме.

Тип диаграммы влияет на ее структуру и предъявляет определенные требования к рядам данных. Так, для построения круговой диаграммы всегда используется только один ряд данных.

Последовательность действий, при построении диаграммы

1. Выделите в таблице диапазон данных, по которым будет строиться диаграмма, включая, если это возможно, и диапазоны подписей к этим данным по строкам и столбцам.

2. Для того чтобы выделить несколько несмежных диапазонов данных, производите выделение, удерживая клавишу <Ctrl>.

3. Вызовите мастера построения диаграмм (пункт меню Вставка/ Диаграмма или кнопка на стандартной панели инструментов).

4. Внимательно читая все закладки диалогового окна мастера построения диаграмм на каждом шаге, дойдите до конца (выбирайте “Далее”, если эта кнопка активна) и в итоге нажмите “Готово”.

После построения диаграммы можно изменить:

-размеры диаграммы, потянув за габаритные обозначения, которые появляются тогда, когда диаграмма выделена;

-положение диаграммы на листе, путем перетаскивания объекта диаграммы мышью;

-шрифт, цвет, положение любого элемента диаграммы, дважды щелкнув по этому элементу левой кнопкой мыши;

-тип диаграммы, исходные данные, параметры диаграммы, выбрав соответствующие пункты из контекстного меню (правая кнопка мыши).

Диаграмму можно удалить: выделить и нажать <Delete>.

Диаграмму, как текст и любые другие объекты в LibreOffice Calc, можно копировать в буфер обмена и вставлять в любой другой документ.

1.1. Задание 1.

Для выполнения заданий вам могут пригодиться функции работы с базами данных, например, dcount, dmax, dmin, daverage, являющиеся аналогами функций без буквы d вначале названия, при этом они позволяют вычислять те же самые параметры, но с условием по какому-либо полю таблицы, при этом условия должны записываться в отдельных ячейках. Так, функция dcount -считает число элементов, dmax — максимальный элемент, dmin — минимальный, daverage — среднее значение (ДСРЗНАЧ). Так же можно использовать СРЗНАЧЕСЛИ.

Типичная форма записи таких функций такая: d****(диапазон ячеек базы данных;имя поля по которому производится расчет;диапазон ячеек с условиями); звездочками обозначено имя функции, которая вычисляет требуемые значения.

	A	B
1	Color	Weight
2	red	2
3	white	3
4	red	4
5	white	1
6	green	3
7	red	4
8		
9	Color	Weight
10	= «red»	

Например для расчета среднего веса красных деталей запишем функцию

=DAVERAGE(A1:B7; «Weight»; A9:B10)

Можно ставить условие в виде > значение, < значение и так далее.

На листе 1 создать удобочитаемую таблицу (таблицы) в соответствием заданием и вашим вариантом. При создании таблицы руководствоваться тем, что столбцов должно быть не менее 7, таблица должна быть красиво оформлена и возможен перерасчет при изменении каких-либо параметров. В качестве общего задания необходимо описать стоимость типовых товаров, их виды, имеющиеся на складе, проданные, провести расчет общей стоимости проданных товаров за какой то период времени, перерасчет стоимости товаров в евро, графики продаж за каждый год в штуках. Разместить в таблице не менее 30 товаров. Рассчитать прибыль от продажи каждого вида товара и построить графики. Рассчитать среднюю цену на каждый вид товара. Оценить процентное соотношение цен однотипных товаров различных фирм друг относительно друга (хотя бы трех фирм). Рассчитать процентное соотношение стоимости проданного товара за год от нереализованного. Учесть что на проданный товар делается процентная надбавка стоимости, на некоторые товары установлена скидка (скидки и надбавки указать в процентах, добавить столбцы с указанием стоимости продажи). Произвести расчет прибыли за какой то период времени. Для того, чтобы отметить проданный товар можно воспользоваться дополнительным столбцом, в котором будет указан данный факт с помощью

выбранной вами метки. Построить круговые диаграммы продаж какой-либо марки товара, чтобы оценить долю каждого по продажам.

Вариант 1.

В различных городах продаются квартиры 1-5-и комнатные, для каждой квартиры указана площадь, тип жилья (новостройка, вторичное), тип постройки (элитная, хрущевка, улучшенной планировки, типовое, сталинка и т.д.), общая цена за квартиру, улица и номер дома. Данные можно вводить на собственное усмотрение в соответствии с вашими представлениями, но более или менее согласующиеся с реальной действительностью, также можно воспользоваться поиском в Интернет.

Рассчитать и разместить в таблице средние цены на новостройки, вторичное жилье, среднюю цену на однокомнатные, двухкомнатные квартиры, среднюю цену на квадратный метр жилья в каждом городе. Также указать и разместить в таблице данные о том, на сколько рублей цена на квадратный метр жилья в других городах выше, чем в Томске. Рассчитать цену квартир в евро. Указать в процентах стоимость элитного жилья относительно средней цены типового.

Отобразить табличные данные в виде диаграмм. Отразить график роста цены в зависимости от числа комнат.

Рассчитать налог взятый от продажи квартир и спрос на квартиры с различными числом комнат в сравнении по годам.

Вариант 2.

Фирма торгует различными видами мебели из различной древесины. Построить таблицу продаж мебели размещенной на складе, информация о поступающей на склад мебели добавляется в таблицу, проданная мебель помечается в отдельной колонке специальным образом (сами выберете каким образом обозначить, например буквой или цифрой). Подсчитать общую сумму на которую было продано мебели за какой-либо месяц, за год. Подсчитать среднюю цену для различных видов мебели, например, среднюю цену стульев, столов и так далее. Рассчитать цену мебели в евро. Отразить график количества продаж мебели различного вида.

Вариант 3.

Фирма торгует комплектующими для компьютеров от различных производителей. Отразить цены на типовые виды комплектующих, средние цены на однотипные комплектующие. Подсчитать среднюю цену проданных комплектующих за какой-либо год, за какой-либо месяц. Отобразить график средних цен на различные виды комплектующих. Рассчитать цену комплектующих в евро. Сделать возможным перерасчет при изменении курса.

Вариант 4.

Фирма занимается продажей автомобилей. Создать таблицу описывающую итоги продаж и имеющиеся предложения. Рассчитать среднюю цену на автомобили определенных марок, среднее количество продаж марок автомобилей за какой-либо год. Рассчитать цену автомобилей в евро. Определить максимальную цену и общую сумму продаж. Нарисовать график средних цен на автомобили различных марок. Рассчитать процент стоимости автомобилей одной марки одного класса относительно другой марки автомобилей того же класса.

Вариант 5.

Фирма занимается продажей мобильных устройств. Указать несколько моделей и фирм производителей.

Вариант 6.

Фирма занимается продажей бытовой техники. Утюги, Вентиляторы, Кондиционеры, и т. д. Указать несколько фирм производителей.

Вариант 7.

Фирма занимается продажей продуктов питания. Указать производителей на различные виды продуктов. (Молоко, Хлеб, Масло, и т.д.)

Вариант 8.

Фирма занимается продажей бытовой химии. Несколько фирм производителей и различные виды бытовой химии (Ацетон, Стеклоотчистители и т.д.)

Вариант 9.

Фирма занимается реализацией напитков. Указать несколько производителей (Соки, Вина, Различные виды соков и вин).

Вариант 10.

Фирма занимается реализацией спортивных товаров. Указать фирмы производители и виды товаров.

3.4 Задание Libre Office Calc

Перейти на лист2, создать функции в табличном виде и отобразить их на графиках. Для этого от -2 до 2, с шагом 0.1 задать значения аргумента x в первом столбце, с помощью операции копирования и формулы в соответствии с вашим вариантом:

1) $y=\begin{cases}\frac{1+x^2}{\sqrt{1+x^4}}, x\le 0\\ 2x+\frac{\sin^2(x)}{3+x}, x>0\end{cases}$	2) $y=\begin{cases}\frac{3+\sin^2(2x)}{1+\cos^2(x)}, x\le 0\\ 2x+\frac{\sin^2(x)}{3+x}, x>0\end{cases}$
3) $y=\begin{cases}\frac{3+\sin^2(x)}{1+x^4}, x\le 0\\ 2x^2\cos^2(x), x>0\end{cases}$	4) $y=\begin{cases}\sqrt{1+x^2}, x\le 0\\ \frac{1+x}{\sqrt{1+e^{-0.2x}}+1}, x>0\end{cases}$
5) $y=\begin{cases}\frac{\sqrt{1+\lvert x\rvert}}{2+\lvert x\rvert}, x\le 0\\ \frac{1+x}{2+\cos^3(x)}, x>0\end{cases}$	6) $y=\begin{cases}3\sin(x)-\cos^3(x), x\le 0\\ \frac{3\sqrt{1+x^2}}{\ln(x+5)}, x>0\end{cases}$
7) $y=\begin{cases}\frac{3x^2}{1+x^2}, x\le 0\\ \sqrt{1+\frac{2x}{e^{0.5x}+x^2}}, x>0\end{cases}$	8) $y=\begin{cases}\sqrt{1+2x^2-\sin^2(x)}, x\le 0\\ \frac{2+x}{\sqrt[3]{2+e^{-0.1x}}}, x>0\end{cases}$
9) $y=\begin{cases}\sqrt{1+\lvert x\rvert}, x\le 0\\ \frac{1+3x}{\sqrt[3]{1+x+2}}, x>0\end{cases}$	10) $y=\begin{cases}\sqrt[3]{1+x^2}, x\le 0\\ \sin^2(x)+\frac{1+x}{1+e^x}, x>0\end{cases}$

Во втором столбце создать копию столбца со значениями функции.

Отобразить графики функций с помощью графиков функций (диаграмм). Пользоваться копированием формул, использовать функцию ЕСЛИ. Например, задать первое значение как -2, затем ниже как =A1+0.1 и скопировать.

Реализовать заполнение табличной функции на листе из Задания 2. Ввод интервалов и шага функции осуществлять из какой-либо ячейки Листа calc.

При выполнении заданий пользоваться форматированием и оформлением таблиц, таблицы должны быть отделены границами, хорошо читаться, представляя собой готовый форматированный отчет. Все графики должны иметь подписи осей. Будет оцениваться качество и творческий подход при выполнении заданий.

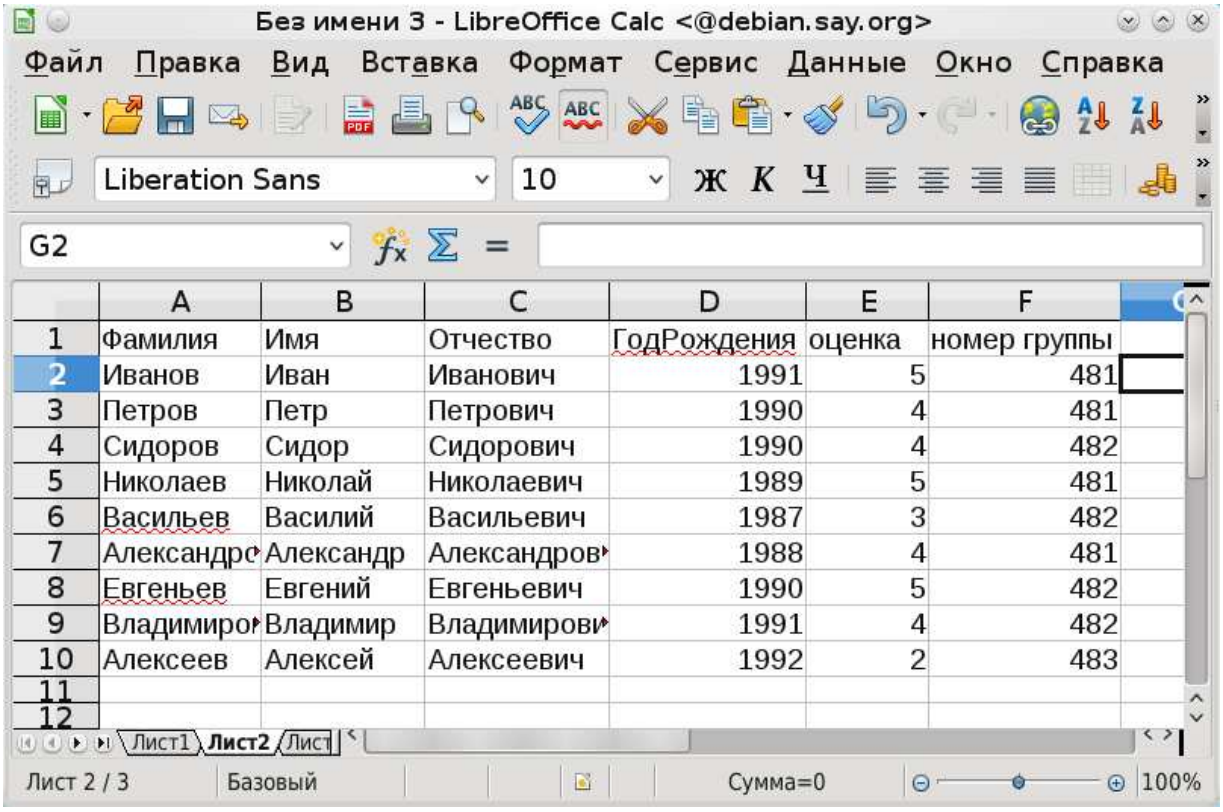
4 Использование Calc как базы данных, изучение макросов

Цель работы. Освоить применение процедур и функций в макросах Basic, научиться применять в расчетах различные виды операторов цикла, создавать пользовательские функции и использовать стандартные функции работы с массивами с использованием электронных таблиц Calc.

Базы данных необходимы для хранения различных сведений о какой-либо предметной области, выполнять фильтрацию, поиск, группировку данных по необходимым признакам. В лабораторной работе рассмотрены простейшие возможности Calc, которые позволяют выполнять типичные операции над данными, более подробное знакомство с СУБД и БД проводится при изучении дисциплины Базы данных.

4.1 Фильтрация данных

Дана таблица с шапкой как в примере представленном на рисунке 4.1, необходимо дополнить ее до 15-20 записей:



	A	B	C	D	E	F
1	Фамилия	Имя	Отчество	ГодРождения	оценка	номер группы
2	Иванов	Иван	Иванович	1991	5	481
3	Петров	Петр	Петрович	1990	4	481
4	Сидоров	Сидор	Сидорович	1990	4	482
5	Николаев	Николай	Николаевич	1989	5	481
6	Васильев	Василий	Васильевич	1987	3	482
7	Александро	Александр	Александров	1988	4	481
8	Евгеньев	Евгений	Евгеньевич	1990	5	482
9	Владимиро	Владимир	Владимирови	1991	4	482
10	Алексеев	Алексей	Алексеевич	1992	2	483
11						
12						

Рисунок 4.1 - Таблица с результатами экзамена

Студентов пронумеровать с помощью формулы, а не вручную, начиная от 1, добавив еще один столбец – номер студента. Скопировать формулу ниже по столбцам на остальные строки таблицы.

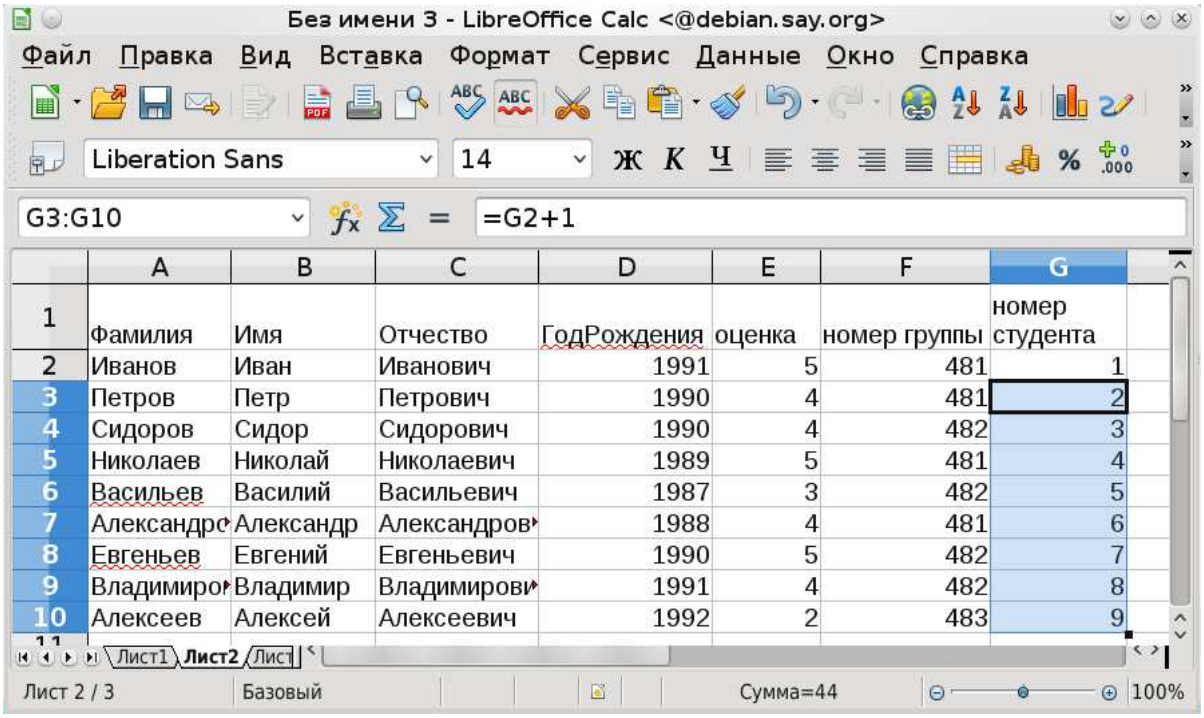


Рисунок 4.2 - Добавление столбца с нумерацией

В одной из ячеек на рисунке 4.2 осуществлен перенос внутри ячейки, это осуществляется с помощью вызова меню Формат-Ячейки, в результате появляется следующее окно (рисунок 4.3), необходимо выбрать вкладку выравнивание и установить флажок переносов.

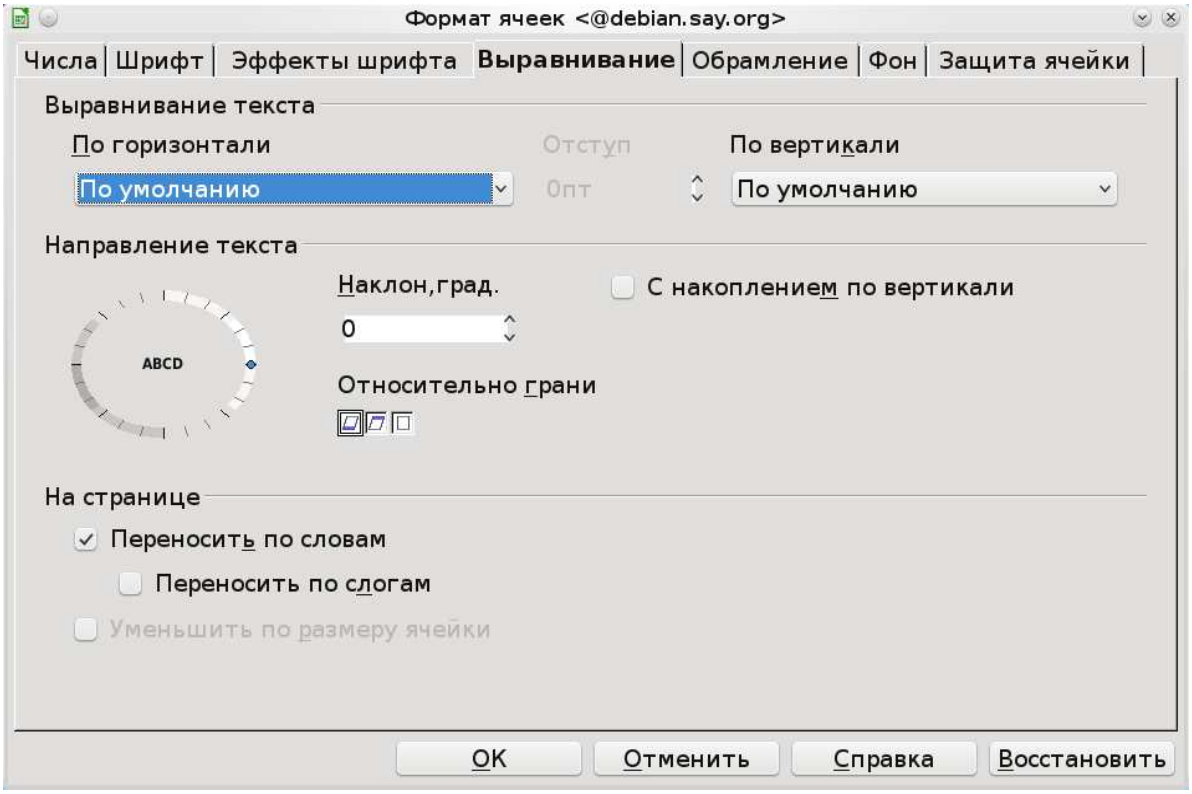
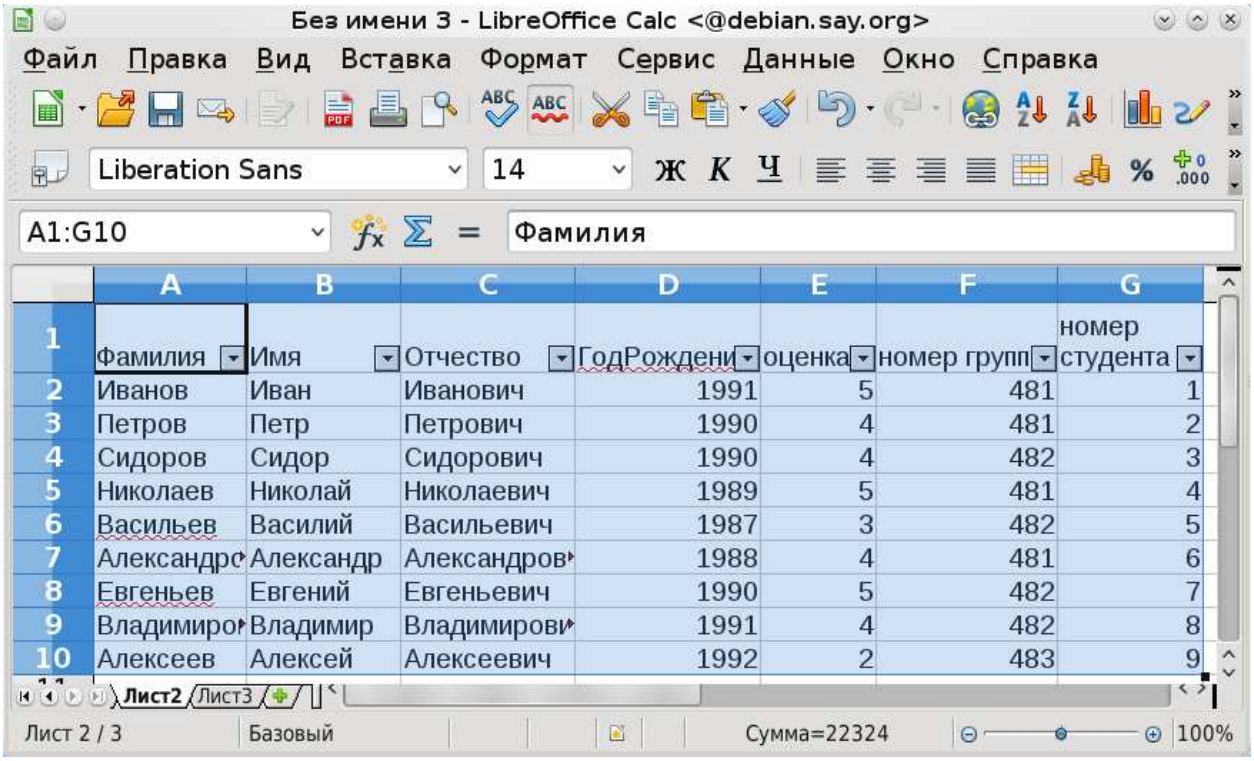


Рисунок 4.3 - Форматирование ячейки

Используя меню «Данные Фильтр-Автофильтр» вывести данные по студентам оценка, которых выше 4. Выбрать студентов оценка которых выше 2 и меньше 5. Для этого необходимо выделить всю таблицу и выбрать «Данные Фильтр-Автофильтр».



	A	B	C	D	E	F	G
1	Фамилия	Имя	Отчество	ГодРождени	оценка	номер групп	номер студента
2	Иванов	Иван	Иванович	1991	5	481	1
3	Петров	Петр	Петрович	1990	4	481	2
4	Сидоров	Сидор	Сидорович	1990	4	482	3
5	Николаев	Николай	Николаевич	1989	5	481	4
6	Васильев	Василий	Васильевич	1987	3	482	5
7	Александро	Александр	Александров	1988	4	481	6
8	Евгеньев	Евгений	Евгеньевич	1990	5	482	7
9	Владимиро	Владимир	Владимирови	1991	4	482	8
10	Алексеев	Алексей	Алексеевич	1992	2	483	9

Рисунок 4.4 - Автофильтр

В выпадающем списке выбрать Стандартный фильтр (рисунок 4.5) условие по нужному столбцу, появится окошко, представленное ниже на рисунке. В окне, представленном ниже установить необходимые условия.

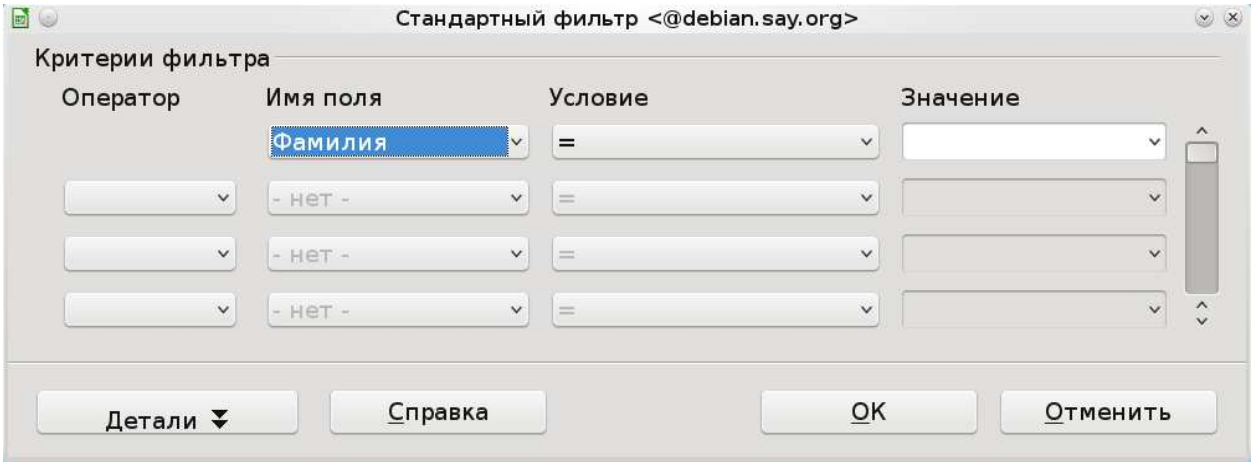


Рисунок 4.5 - Условие на поле таблицы

Отсортировать таблицу по группам, используя «данные - сортировка» с помощью выделения всей таблицы.

Используя «данные – фильтр - расширенный фильтр» сформировать таблицу, где имена студентов Иван или Петр, а оценка выше 3. Ниже приведен пример, где задаются условия для расширенного фильтра. При этом должны быть указаны имена столбцов, для которых проводится фильтрация (полное совпадение имени и формата названия), а также условия, условия расположенные по строкам определяют операцию «И», условия по столбцам дают условие «Или. При применении сравнения со строковыми константами необходимо помнить, что они помещаются в кавычки – “строка”. То есть условия задаются в ячейках Calc, необходимо в ячейках указать нужные нам имена полей, причем поля должны совпадать с названиями полей в таблице для которой мы проводим фильтрацию, а ниже в ячейке указывается условие, больше >, меньше <, больше или равно >=, меньше или равно <=. Выделяем исходную таблицу и выбираем «данные – фильтр - расширенный фильтр», затем в появившемся окне вводим диапазон ячеек в, которых указаны условия, для этого можно мышкой выделить прямоугольную область прямо на листе Calc.

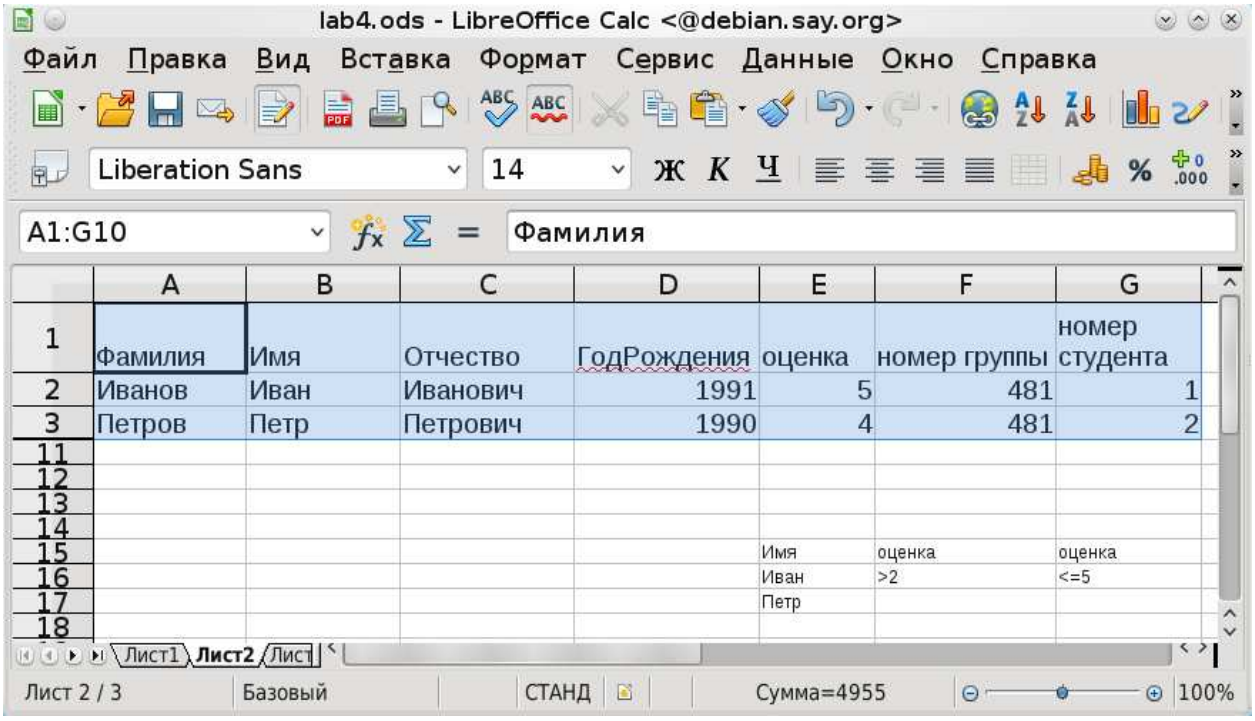


Рисунок 4.6 - Использование расширенного фильтра

Пример:

Имя	оценка	оценка	Год рождения
Иван	<5	>3	
Петр			
			>1990

Читается как Иван с оценкой меньше 5 и больше тройки или студент с именем Петр или кто угодно с годом рождения >1990.

Создать еще одну таблицу на основе предыдущей, где фамилия, имя, отчество стоит в одном столбце, для этого использовать функцию CONCATENATE(СЦЕПИТЬ) (текст1;текст2;...). Текст1, текст2, ... — это от 1 до 30 элементов текста, объединяемых в один элемент текста.

Синтаксис:

CONCATENATE("Текст1"; ...; "Текст30")

Текст 1; текст 2; ...: до 30 текстовых элементов, которые требуется объединить в одну строку.

Пример:

=CONCATENATE("Доброе ";"утро ";"миссис ";"Доу") возвращает значение "Доброе утро, миссис Доу".

Элементами текста могут быть текстовые строки, числа или ссылки, которые ссылаются на одну ячейку. Строковая константа записывается с использованием кавычек («строка»). Шапка будет состоять из столбцов в порядке – номер студента, ФИО, группа, оценка. Необходимо подсчитать средний балл для студентов. Отсортировать по группам, в группах по ФИО.

4.2 Сводные таблицы.

Сводная таблица это инструмент Calc для обработки больших списков с данными. Сводная таблица обслуживается мастером сводных таблиц (Данные + Сводная таблица), позволяющим подводить итоги, выполнять сортировку и фильтрацию списков.

Подведение итогов в сводной таблице производится с помощью итоговой функции (например, "Сумма", "Кол-во значений" или "Среднее"). В таблицу можно автоматически поместить промежуточные или общие итоги, а также добавить формулы в вычисляемые поля или элементы полей. В сводной таблице содержатся поля, подводящие итоги исходных данных в нескольких строках. Переместив кнопку поля в другое место сводной таблицы, можно изменить представление данных.

Сводные таблицы предназначены для удобного просмотра данных больших таблиц, т.к. обычными средствами делать это неудобно, а порой, практически невозможно.

Они содержат часть данных анализируемой таблицы, показанные так, чтобы связи между ними отображались наглядно. Сводная таблица создается на основе отформатированного

списка значений. Поэтому, прежде чем создавать сводную таблицу, необходимо подготовить соответствующим образом данные.

Создайте таблицу вида, дополните ее дополнительными марками телефонов и датами продажи:

Таблица 1 - Рабочая таблица о продажах телефонов

Дата	Магазин	Марка	Серия	Продажа (штуки)	Цена(рубли)	Итого
12.12.2002	1	Самсунг	c300	3	100	300
12.12.2002	1	Нокия	c200	4	1221	4884
12.12.2002	2	Самсунг	c300	5	1212	6060
12.12.2002	2	Нокия	c200	6	121	726
12.12.2002	3	Самсунг	c300	7	122	854

Выделяем всю таблицу. Вызываем Данные+Сводная+Создать таблица. В появившемся окне мастера выбираем текущее выделение и нажимаем Ok.

В появившемся окне задается макет сводной таблицы. В данном случае в строках будут указаны магазины, в столбцах марка телефона и итоговая прибыль по продажам.

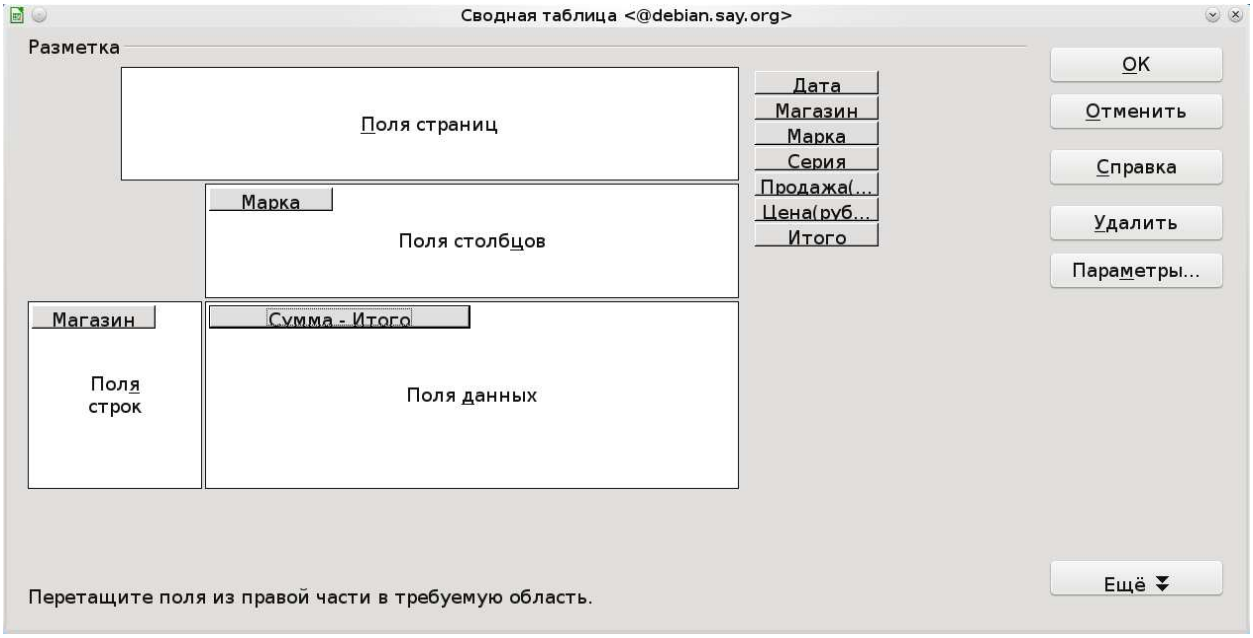


Рисунок 4.7 - Установка параметров сводной таблицы

В результате на новом листе будет получена таблица со следующими данными.

Таблица 2 - Сводная таблица

Сумма по полю Итого	Марка		
Магазин	Нокия	Самсунг	Общий итог
1	4884	300	5184
2	726	6060	6786
3		854	854
Общий итог	5610	7214	12824

Для того, чтобы создать собственное представление таблиц необходимо в окне на рисунке перетаскивать нужные поля в нужные области по строкам, по столбцам и в область данных.

Создать сводную таблицу по продажам конкретных марок телефонов в штуках и в рублях, создать сводную таблицу сгруппированными данными по дате, то есть сколько телефонов определенной марки было продано в определенный день.

4.3 Итоговые поля и группировка

Просматривать и создавать Итоговые поля и проводить группировку по какому-то полю можно, используя Данные+Промежуточные Итоги. Необходимо выделить исходную таблицу и затем выбрать Данные+Промежуточные Итоги и в появившемся окне можно выбрать операцию группировки (например, Сумма), а также поля, по которым необходимо получить итоговые значения (рисунок 28).

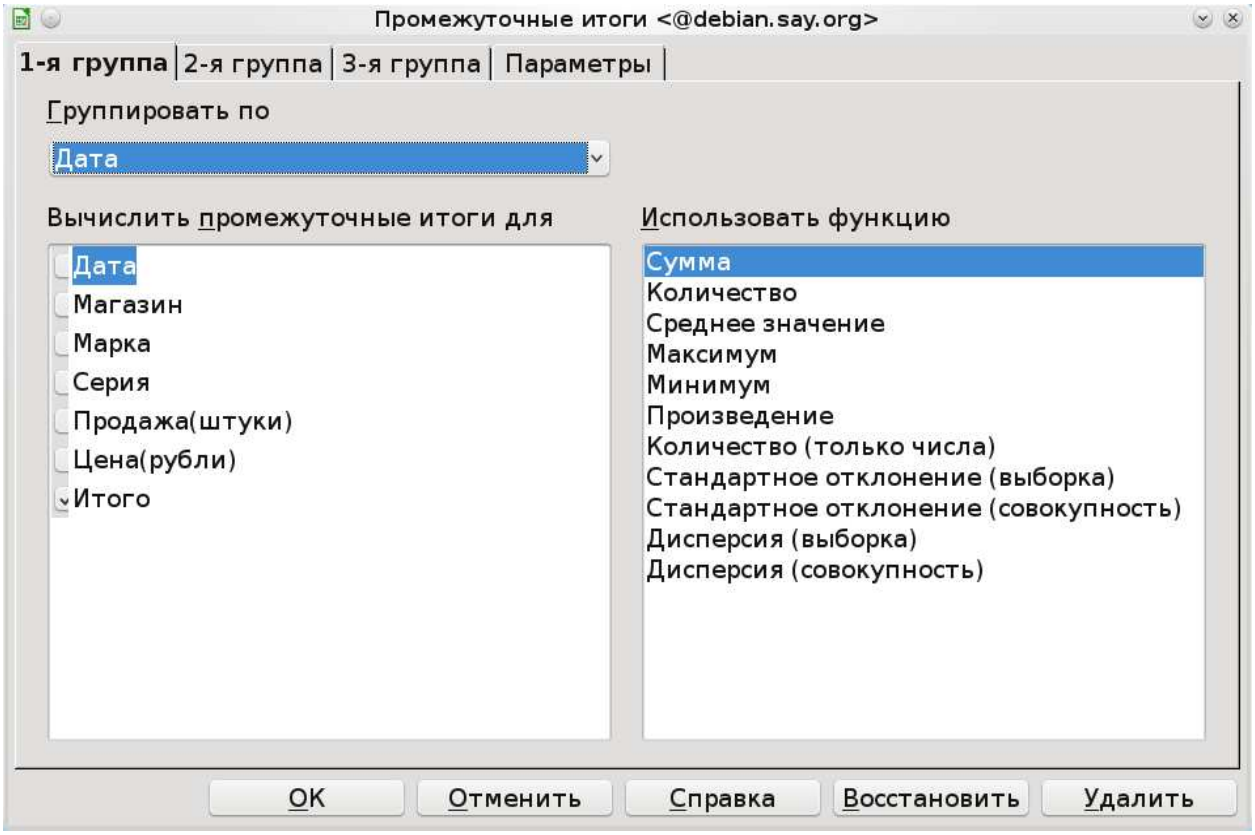


Рисунок 4.8 - Группировка и итоги

В результате будет получена таблица, представленная ниже, путем добавления дополнительных полей к исходной. Появились еще две строки указывающие на сумму по полю итого при группировке по дате.

Таблица 3 - Результирующая таблица

Дата	Магазин	Марка	Серия	Продажа(штуки)	Цена(рубли)	Итого
12.12.02	1	Самсунг	c300	3	100	300
12.12.02	1	Нокия	c200	4	1221	4884
12.12.02	2	Самсунг	c300	5	1212	6060
12.12.02	2	Нокия	c200	6	121	726
12.12.02	3	Самсунг	c300	7	122	854
12.12.02 Сумма						***12824***
Общий итог						***12824***

Проведите группировку по магазинам, а также по маркам телефонов, вот как будет выглядеть таблица в случае группировке по серии.

Таблица 4 - Таблица с итогами по сериям

Дата	Магазин	Марка	Серия	Продажа (штуки)	Цена(рубли)	Итого
12.12.02	1	Нокия	c200	4	1221	4884
12.12.02	2	Нокия	c200	6	121	726
			c200 Результат	***10***	***1342***	
12.12.02	1	Самсунг	c300	3	100	300
12.12.02	2	Самсунг	c300	5	1212	6060
12.12.02	3	Самсунг	c300	7	122	854
			c300 Результат	***15***	***1434***	
			Общий итог	***25***	***2776***	

Также можно убирать из таблицы итоговые или промежуточные поля, в примере ниже представлены только итоговые результаты группировки. Для этого необходимо щелкнуть на кнопках 1, 2 или 3, а также +, -, которые находятся слева.

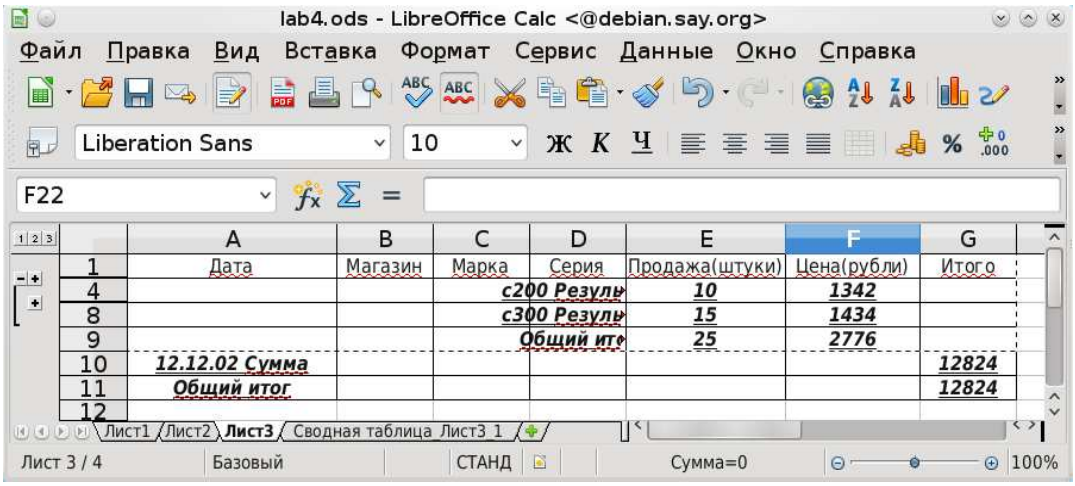


Рисунок 4.9 - Пример итоговой таблицы без исходных данных

Кроме того, можно группировать сначала по одному полю, затем по другому. Вот пример группировки сначала по марке, затем по серии, одну из серий «самсунга» мы сделали c200.

Таблица 5 - Группировка по двум полям

Дата	Магазин	Марка	Серия	Продажа(штуки)	Цена(рубли)	Итого
12.12.02	1	Нокия	c200	4	1221	4884
12.12.02	2	Нокия	c200	6	121	726
			c200 Сумма	***10***		

		Нокия				
		Сумма		***10***		
12.12.02	1	Самсунг	с200	3	100	300
			с200			
			Сумма	***3***		
12.12.02	2	Самсунг	с300	5	1212	6060
12.12.02	3	Самсунг	с300	7	122	854
			с300			
			Сумма	***12***		
		Самсунг				
		Сумма		***15***		
		Общий итог		***25***		

4.4 Задание

Повторить все пункты данного параграфа. Изучить сводные, итоговые таблицы и фильтрацию.

5 Изучение макросов Calc Basic

5.1 Вычисление премиальных по процентам

Составить таблицу начисления премиальных по итогам работы сети 4-х магазинов за три месяца по следующему правилу:

- если продукции продано меньше чем на 60000 рублей, то премиальные составляют 2% от суммарной выручки магазина;

- за первое место дополнительно начисляется 4% премиальных, за второе 2%, за третье 1% от суммарной выручки магазина.

Сначала составим таблицу и заполним значениями как на рисунке.

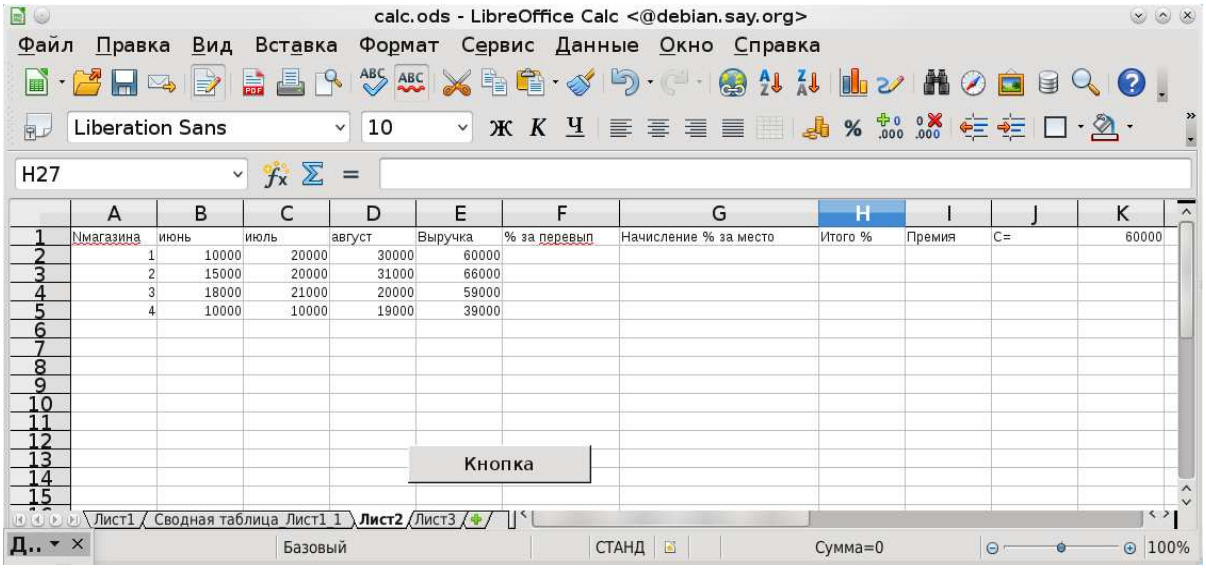


Рисунок 5.1 - Расчетная таблица по магазинам

Поместим на лист Calc кнопку. Как вытащить кнопку на лист. Сначала с помощью Вид-Панели инструментов-Элементы управления выведем нужную панель с элементами управления. Выбрать элемент управления в открывшемся окне — кнопка и поместить ее на листе. Чтобы назначить событие для данной кнопки, можно щелкнуть правой кнопкой мыши и выбрать во всплывающем меню «Элемент управления» События, назначить макрос обрабатывающий одно из событий (Выполнить действие). Предварительно нужно создать макрос, можете его назвать, например, OnClick, напомним, что он создается так же как и во Writer, нужно открыть окно редактора Basic и в нем записать пустую процедуру. В макросе для начала можно записать начальный код MsgBox «Hello». Затем на элементах управления

перевести состояние из режима конструктора в режим выполнения, используя кнопку — «линейка-треугольник».

Вызов на кнопке контекстно-зависимого меню и выбор опции Элемент управления и Общие. В поле Текст заменяем стандартное имя на любое свое, это будет видимое название кнопки.

Рассмотрим кратко некоторые свойства объекта кнопка общие и для остальных визуальных объектов.

Свойство доступно (Enabled) объекта позволяет запретить или разрешить доступ к объекту. При свойстве со значением False (Нет) кнопку нельзя нажимать, она отображается серым цветом.

Свойства Width (Ширина) и Height (Высота) – задают ширину и высоту объекта кнопка. Свойства Top (Позиция y) и Left (Позиция x) – задают смещения от верхнего и левого краев формы. BackColor (цвет фона) – цвет фона объекта. Font (Шрифт) – свойство шрифта объекта, шрифт надписей на объекте. Picture (Изображения) – путь к картинке. Visible (Видимость) – задает видимость объекта. WordWrap (разрыв слова) – перенос слов отображающихся в имени.

Напишем следующий код для этой процедуры с использованием циклических структур. Текст, начинающийся с кавычки - примечание.

```
Sub OnClick()
Dim Doc As Object
Dim Sheet, Cell As Object
'Пример получения доступа к текущему открытому листу по его номеру.
'Doc = StarDesktop.CurrentComponent
'Sheet = Doc.Sheets(1)
'получение доступа к листу по его имени
Doc = StarDesktop.CurrentComponent
Sheet = Doc.Sheets.getByName("Лист2")
‘цикл суммирования выручки за 3 месяца
for i =1 to 4
Cell = Sheet.getCellByPosition(4, i)
Cell.Value = Sheet.getCellByPosition(1, i).Value + Sheet.getCellByPosition(2, i).Value + Sheet.getCellByPosition(3, i).Value
‘сделать сложение по столбцам с помощью цикла !!!!!
next i
```

```
‘Создание вспомогательного массива box
‘заполнение его значениями выручки
‘создание массива из четырех элементов вещественного типа с индексацией от 1 до 4.
‘ найти ошибку
Dim box(4) As Double
box(0) = Sheet.getCellByPosition(4, 0).Value
box(1) = Sheet.getCellByPosition(4, 1).Value
box(2) = Sheet.getCellByPosition(4, 2).Value
box(3) = Sheet.getCellByPosition(4, 3).Value

‘присвоение реализовать с помощью цикла !!!!!
‘Сортировка выручки за 3 месяца по убыванию методом «пузырька», найти ошибку.
‘При этом в box(0) окажется максимальное значение выручки:
For I =0 to 3
For j=0 to 4-i
If box(j)<box(j+1) then
q = box(j+1)
box(j+1)=box(j)
box(j)=q
Endif
Next
Next
‘Реализовать сортировку методом простого выбора!!!!

‘Начисление процентов в зависимости от места
For i=1 to 4
If Sheet.getCellByPosition(4, i).Value = box(0) Then Sheet.getCellByPosition(6, i).Value=4
If Sheet.getCellByPosition(4, i).Value = box(1) Then Sheet.getCellByPosition(6, i).Value=2
If Sheet.getCellByPosition(4, i).Value = box(2) Then Sheet.getCellByPosition(6, i).Value=1
If Sheet.getCellByPosition(4, i).Value = box(3) Then Sheet.getCellByPosition(6, i).Value=0
Next i
```

‘тоже самое реализовать, используя данные из таблицы и цикл, или из предварительно созданного массива, чтобы не использовать четыре строчки сравнения

Начисление процентов, если выручка за 3 месяца больше плановой выручки.

```
For i=1 to 4
If Sheet.getCellByPosition(4,i).Value>=Sheet.getCellByPosition(10,1).value then
Sheet.getCellByPosition(5,i).value = 2
Else: Sheet.getCellByPosition(5,i).value =0
End if
Next i

Суммирование процентов

For i=1 to 4
Sheet.getCellByPosition(7,i).value = Sheet.getCellByPosition(5,i).value + Sheet.getCellByPosition(6,i).value
Next

Расчет итоговой премии
For i=1 to 4
Sheet.getCellByPosition(8,i).value = Sheet.getCellByPosition(4,i).value + Sheet.getCellByPosition(7,i).value/100
Next

End sub
```

Закроем редактор и нажмем на кнопку. Получим заполненную таблицу.

В результате в столбце E окажется сумма выручек за три месяца, в столбце F – процент, назначенный за перевыполнение плана, в столбце G – процент, назначенный в зависимости от занятого места, в столбце H – итоговый процент, в столбце I - величина премии.

5.2 Начисление премиальных. Использование функции.

Составить таблицу начисления премии по итогам работы сети 4 магазинов за три месяца по следующему правилу:

- если продукции продано на сумму меньше чем 20 тыс. руб. то премия не начисляется.

- если продукции продано на сумму от 20 до 40 тыс. рублей, премия составляет 3% от выручки.

- если продукции продано на сумму от 40 до 80 тыс. рублей, премия составляет 4.5 % от выручки.

- если продукции продано больше чем на 80 тыс. руб. то премия составляет 6.5%.

Аналогично первому заданию составим исходную таблицу и создадим кнопку.

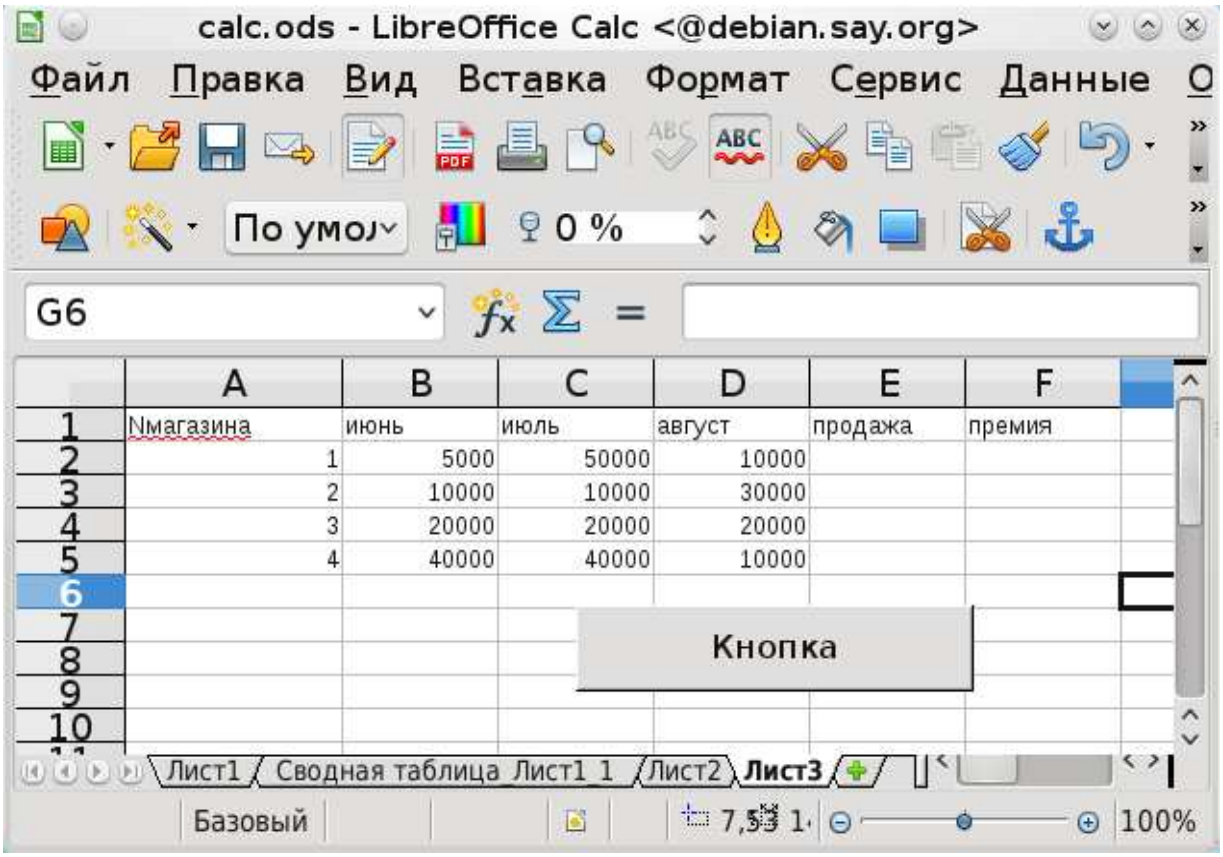


Рисунок 5.2 - Исходная таблица с данными о магазинах

Создадим отдельно функцию «Премия» - начисление премии в зависимости от объема продаж. Для этого ниже процедуры Onclick напишем функцию:

```
Function Premia(Prodaja as Double) As Double
Select case  Prodaja
Case 0 to 20000
Premia = 0
Case 20001 to 40000
Premia = 0.03 * Prodaja
Case 40001 to 80000
Premia = 0.045 * Prodaja
End Select
End Function
```

Пример еще одной функции:

```
Function Prem(Prod as Double) as Double
Prem = Prod/2
end function
```

Допустимо объявить функцию ниже по коду, после кода в котором происходит ее вызов. После того, как вы определите функцию в коде, вы сразу можете ее использовать и вызывать непосредственно на листе Calc, например, =Prem(A1).

Функция это отдельная подпрограмма, которая может вызываться из основной по отношению к ней подпрограммы или программы. При вызове функции основная программа передает ей управление, функция выполняет свои операторы и завершает выполнение, передавая основной программе вычисленные значения, при этом начинается выполняться оператор основной программы, следующий после вызова функции.

При этом в функцию могут передаваться какие-то значения или ссылки на переменные, в первом случае говорят о передаче в функцию фактических параметров, во втором о передаче формальных параметров. При передаче фактических параметров в функцию передается копия объекта, или копия значения, которая не может изменяться внутри процедуры или функции. При передаче ссылки ан объект или формального параметра, функция может его изменять и соответственно возвращать в нем какие то вычисленные значения. Таким же формальным параметром выступает и имя самой функции, вызов которой можно осуществлять непосредственно в выражениях основной программы,

Например

```
Sheet.getCellByPosition(5,i).value = Премия(Sheet.getCellByPosition(4,i).value) или
Sheet.getCellByPosition(5,i).value = 100 +
Премия(Sheet.getCellByPosition(4,i).value*1.1+5000)
```

В первом случае в функцию передается значение Sheet.getCellByPosition(4,i).value, и функция возвращает расчетное значение в соответствии с алгоритмом и это значение присваивается Sheet.getCellByPosition(5,i).value.

Во втором случае в функцию передается значение выражения Sheet.getCellByPosition(4,i).value*1.1+5000. Затем функция вычисляется, происходит суммирование со значением 100 и результат помещается в Sheet.getCellByPosition(5,i).value.

Добавьте в редакторе Бэйсика код макроса обработчика onclick2 для реализации задания с использованием функции Премия, то есть реализуйте вызов функции премия, таким образом,

чтобы был произведен расчет и полей продажа и полей премия. Очевидно, что поле продажа получена суммированием продаж по месяцам.

5.3 Вычисление формул, реализация вычислительных функций.

Вычислить значение $S = \frac{2\sum_{i=1}^{m} a_i + \left(\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\right)}{\left(1+\sum_{i=1}^{m} a_i\right)\left(1+\sum_{i=1}^{m} a_i^2\right)}$, где c матрица размерности $n \times n$, причем $n = 3$, $m = 4$. $a = [3,1,2,3]$.

$$c = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{bmatrix}.$$

Введите данные на лист Calc, матрицу в виде таблицы и вектор в виде строки. Рассчитайте значение S используя стандартные функции Calc в какой либо ячейке.

Для того, чтобы использовать встроенные функции calc можно воспользоваться следующим кодом, пример расчета среднего:

```
CellRange = Sheet.getCellRangeByName("A1:C3")
CellRange.computeFunction(com.sun.star.sheet.GeneralFunction.AVERAGE)
```

или, пример расчета суммы

```
CellRange = Sheet.getCellRangeByName("A1:C3")
CellRange.computeFunction(com.sun.star.sheet.GeneralFunction.SUM)
```

Кроме AVERAGE и SUM допустимо использовать следующие функции :

- SUM – сумма всех числовых значений;
- COUNT – общее количество всех значений (включая нечисловые значения);
- COUNTNUMS – общее количество всех числовых значений;
- AVERAGE – среднее арифметическое всех числовых значений;
- MAX – наибольшее числовое значение;
- MIN – наименьшее числовое значение;
- PRODUCT – произведение всех числовых значений;
- STDEV - стандартное отклонение;
- VAR – дисперсия;
- STDEVP - стандартное отклонение, основанное на генеральной совокупности;

• VARP - дисперсия, основанная на генеральной совокупности.

Реализуйте отдельно функцию расчета суммы квадратов элементов.

Пример реализации с помощью стандартных функций приведен ниже, сделайте расчет суммы с помощью своей функции, по аналогии с суммой квадратов элементов.

Функция считающая сумму квадратов, в качестве входного параметра служит объект cellsrange, который должен быть получен функцией GetCellRangeByName.

```
Function sumqw(cell as Variant) as Variant
dim d as double
d = 0
'цикл по строкам и столбцам передаваемого диапазона
for i = 0 to cell.Rows.Count-1
for j = 0 to cell.Columns.Count-1
'накапливаем сумму
d = d + Cell.getCellByPosition(j,i).Value^2
next j
next i
'присваиваем накопленную сумму
sumqw =  d
end function
'функция вычисляющая формулу, в качестве параметров в нее должны передваться:
'list — строка указывающая имя листа в документе
'a1 — строка с диапазоном ячеек массива a
'c1 — строка с диапазоном ячеек матрицы c1
Function Formula1(list as string, a1 as string, c1 as string) as Double
dim suma,sumc,sumaqw as double
dim doc,cellsc,cellsa as Object
'получаем текущий открытый документ
doc = StarDesktop.CurrentComponent
'получаем диапазон ячеек по строке c1
cellsc = doc.sheets.getbyname(list).GetCellRangeByName(c1)
'получаем диапазон ячеек по строке a1
cellsa = doc.sheets.getbyname(list).GetCellRangeByName(a1)
' с помощью нашей функции считаем сумму квадратов
sumaqw = sumqw(cellsa)
```

```
' считаем сумму диапазона ячеек с помощью стандартной функции
suma = cellsa.computeFunction(com.sun.star.sheet.GeneralFunction.SUM)
sumc = cellsc.computeFunction(com.sun.star.sheet.GeneralFunction.SUM)
'вычисляем формулу
Formula1 = (2*suma+sumc)/((1+suma)*(1+sumaqw))
end function
```

Пример вызова формулы из Calc:

```
=FORMULA1("Лист5";"A1:C1";"A1:C3")
```

Другой способ определения функции, когда диапазон ячеек передается не как строка, а обычным способом, как и в других функциях Calc. Ниже приведен пример расчета функции суммы квадратов:

```
'в функцию передается массив range
Function sumqw1(range) as Variant
dim d as double
d = 0
'цикл с нижней до верхней границы массива по строкам
for i = LBound(Range,1) to UBound(Range,1)
'цикл с нижней до верхней границы массива по столбцам
for j = LBound(Range,2) to UBound(Range,2)
d = d + range(i,j)^2
next j
next i
sumqw1 =  d
end function
```

Вызов функции из calc осуществляется обычным способом:

```
=SUMQW1(A1:C3)
```

Написать функцию расчета формулы используя обычный способ задания диапазона.

5.4 Задание

Освоив основные возможности работы с макросами и выполнив предыдущие задания параграфа, выполните по вариантам следующие задания.

На таблице Calc размещена числовая таблица (матрица). Написать функцию для расчетов по диапазону ячеек в виде XXNN:YYMM. Например = func(A1:B6). Использовать массив. Напоминание: если передать в функцию переменную как диапазон, то это будет двумерный массив границы которого определяются функциями LBound и UBound. Например UBound(Range, 1), где первый параметр указывает на переданный диапазон (массив) и второй параметр на номер измерения, 1 – строки, 2 – столбцы. Таким образом, если 1, то Lbound – верхняя граница, UBound – нижняя, если 2, то левая и правая. По умолчанию, левая и нижняя границы равны 1.

1. Посчитать сумму столбца по середине матрицы и строки по середине. Если матрица с четным числом строк или столбцов брать две строки или два столбца.

2. Посчитать сумму двух диагоналей матрицы.

3. Посчитать сумму верхней и нижней строки и одной диагонали.

4. Посчитать среднее нижней строки и левого столбца.

5. Посчитать сумму верхней строки и главной диагонали.

6. Посчитать произведение суммы главной диагонали на верхний левый и на нижний правый элементы матрицы.

7. Посчитать сумму разниц четных и нечетных элементов диагонали. Матрица с четной размерностью по строкам и столбцам. Например, (2 на 2).

8. Посчитать сумму нижней строки и главной диагонали. И умножить на сумму побочной диагонали.

9. Посчитать сумму всех четных элементов матрицы по столбцам и строкам. (2,2), (2, 4) и т. д.

10. Посчитать сумму верхней и нижней строки. Умножить на сумму диагонали.

11. Посчитать сумму элементов в матрице по номерам чисел Фибоначчи. Нумерация идет от верхней строки последовательно, потом по второй строке и так далее.

Например 1, 2, 3

4, 5, 6 , считать элементы под номером 1, 2, 3, 5, 8 и т. д.

12. Посчитать сумму элементов в матрице по номерам степени двойки. Нумерация идет от верхней строки последовательно, потом по второй строке и так далее.

Например 1, 2, 3

4, 5, 6 , считать элементы под номером 1, 2, 4, 8, 16 и т. д.

13. Посчитать сумму отрицательных элементов матрицы и умножить на сумму положительных элементов.

14. Посчитать среднее значение по матрице и посчитать сумму элементов первой строки больше среднего.

15. Посчитать сумму произведений четных и нечетных строк.

16. Посчитать сумму произведений четных и нечетных столбцов.

17. Посчитать произведение сумм диагональных элементов в матрице. Например:

1 4 5 7

1 2 4 5

6 7 8 1

2 3 4 2

1*(1+4)*(6+2+5)*(2+7+4+7)*(3+8+5)*(4+1)*2

18. Посчитать сумму всех элементов матрицы, умноженную на верхний правый элемент и на среднее значение диагонали.

19. Посчитать сумму среднего по столбцу и по строке элемента матрицы, верхнего правого, левого нижнего.

20. Посчитать сумму всех элементов матрицы не превосходящих правый нижний элемент.

6 Изучение работы в командной строке

Для начала изучим начальные сведения об операционных системах и их загрузке.

6.1 Начальная загрузка компьютера

В информатике начальной загрузкой называется сложный и многошаговый процесс запуска компьютера. Загрузочная последовательность — это последовательность действий, которые должен выполнить компьютер для запуска операционной системы.

Большинство компьютерных систем могут исполнять только команды, находящиеся в оперативной памяти компьютера, в то время как современные операционные системы в большинстве случаев хранятся на жёстких дисках, загрузочных CDROM-ах, USB дисках или в локальной сети.

После включения компьютера в его оперативной памяти нет операционной системы. Само по себе, без операционной системы, аппаратное обеспечение компьютера не может выполнять сложные действия, такие как, например, загрузку программы в память. Таким образом, мы сталкиваемся с парадоксом, который кажется неразрешимым: для того, чтобы загрузить операционную систему в память, мы уже должны иметь операционную систему в памяти.

Решением данного парадокса является использование специальной маленькой компьютерной программы, называемой начальным загрузчиком, или BIOS (Basic Input/Output System). Эта программа не обладает всей функциональностью операционной системы, но её достаточно для того, чтобы загрузить другую программу, которая будет загружать операционную систему. Часто используется многоуровневая загрузка, в которой несколько небольших программ вызывают друг друга до тех пор, пока одна из них не загрузит операционную систему. В современных компьютерах процесс начальной загрузки начинается с выполнения процессором команд, расположенных в постоянной памяти (например на IBM PC — команд BIOS), начиная с предопределённого адреса (процессор делает это после перезагрузки без какой бы то ни было помощи). Данное программное обеспечение может обнаруживать устройства, подходящие для загрузки, и загружать со специального раздела выбранного устройства (чаще всего загрузочного сектора данного устройства) загрузчик ОС.

Начальные загрузчики должны соответствовать специфическим ограничениям, особенно это касается объёма. Например, на IBM PC загрузчик первого уровня должен помещаться в первых 446 байт главной загрузочной записи, оставив место для 64 байт таблицы разделов и 2 байта для сигнатуры AA55, необходимой для того, чтобы BIOS выявил сам начальный загрузчик.

Устройства, инициализируемые BIOS

Загрузочное устройство — устройство, которое должно быть проинициализировано до загрузки операционной системы. К ним относятся устройства ввода (клавиатура, мышь), базовое устройство вывода (дисплей), и устройство, с которого будет произведена загрузка ОС — дисковод, жесткий диск, CD-ROM, флэш-диск, SCSI-устройство, сетевая карта (при загрузке по сети; например, при помощи PXE).

Загрузочная последовательность стандартного IBM-совместимого персонального компьютера

После включения персонального компьютера его процессор начинает работу. Первая выполняемая команда расположена по адресу FFFF0h и принадлежит пространству адресов BIOS. Как правило, данная команда просто передает управление программе инициализации BIOS.

Программа инициализации BIOS с помощью программы POST проверяет, что устройства компьютера работают корректно и инициализирует их.

Затем BIOS опрашивает устройства, перечисляемые в заранее созданном списке, пока не найдёт загрузочное устройство. Если такое устройство найдено не будет, будет выведено сообщение об ошибке, а процесс загрузки будет остановлен. Если BIOS обнаружит загрузочное устройство, он считает с него начальный загрузчик и передаст ему управление.

В случае жесткого диска, начальный загрузчик называется главной загрузочной записью (MBR) и часто не зависит от операционной системы. Обычно он ищет активный разделы жесткого диска, загружает загрузочный сектор данного раздела и передает ему управление. Этот загрузочный сектор, как правило, зависит от операционной системы. Он должен загрузить в память ядро операционной системы и передать ему управление. Если активного раздела не существует, или загрузочный сектор активного раздела некорректен, MBR может загрузить резервный начальный загрузчик и передать управление ему. Резервный начальный загрузчик должен выбрать раздел (зачастую с помощью пользователя), загрузить его загрузочный сектор и передать ему управление.

6.2 Что же такое операционная система?

Это базовый комплекс компьютерных программ, обеспечивающий интерфейс с пользователем, управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит.

ОС позволяет абстрагироваться от деталей реализации аппаратного обеспечения, предоставляя разработчикам программного обеспечения минимально необходимый набор

функций. С точки зрения обычных пользователей компьютерной техники ОС включает в себя и программы пользовательского интерфейса.

Основные функции (простейшие ОС):

Загрузка приложений в оперативную память и их выполнение;

Стандартизованный доступ к периферийным устройствам (устройства ввода-вывода);

Управление оперативной памятью (распределение между процессами, виртуальная память);

Управление доступом к данным на энергонезависимых носителях (таких как Жёсткий диск, Компакт-диск и т. д.), как правило с помощью файловой системы;

Пользовательский интерфейс;

Сетевые операции, поддержка стека протоколов

Дополнительные функции:

Параллельное или псевдопараллельное выполнение задач (многозадачность);

Взаимодействие между процессами: обмен данными, взаимная синхронизация;

Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений;

Разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

Существуют две группы определений ОС: «совокупность программ, управляющих оборудованием» и «совокупность программ, управляющих другими программами». Обе они имеют свой точный технический смысл, который, однако, становится ясен только при более детальном рассмотрении вопроса о том, зачем вообще нужны операционные системы.

Есть приложения вычислительной техники, для которых ОС излишни. Напр., встроенные микрокомпьютеры содержатся сегодня во многих бытовых приборах, автомобилях (иногда по десятку в каждом), сотовых телефонах и т. п. Зачастую такой компьютер постоянно исполняет лишь одну программу, запускающуюся по включении. И простые игровые приставки — также представляющие собой специализированные микрокомпьютеры — могут обходиться без ОС, запуская при включении программу, записанную на вставленном в устройство «картридже» или компакт-диске (часто версии ОС для таких систем называются Embedded). Тем не менее, некоторые микрокомпьютеры и игровые приставки всё же работают под управлением особых собственных ОС. В большинстве случаев, это UNIX-подобные системы (последнее особенно верно в отношении программируемого коммутационного оборудования: фаерволов, маршрутизаторов).

Операционные системы, в свою очередь, нужны, если:

вычислительная система используется для различных задач, причём программы, исполняющие эти задачи, нуждаются в сохранении данных и обмене ими. Из этого следует необходимость универсального механизма сохранения данных; в подавляющем большинстве случаев ОС отвечает на неё реализацией файловой системы. Современные ОС, кроме того, предоставляют возможность непосредственно «связать» вывод одной программы с вводом другой, минуя относительно медленные дисковые операции;

различные программы нуждаются в выполнении одних и тех же рутинных действий. Напр., простой ввод символа с клавиатуры и отображение его на экране может потребовать исполнения сотен машинных команд, а дисковая операция — тысяч. Чтобы не программировать их каждый раз заново, ОС предоставляют системные библиотеки часто используемых подпрограмм (функций);

между программами и пользователями системы необходимо распределять полномочия, чтобы пользователи могли защищать свои данные от несанкционированного доступа, а возможная ошибка в программе не вызывала тотальных неприятностей;

необходима возможность имитации «одновременного» исполнения нескольких программ на одном компьютере (даже содержащем лишь один процессор), осуществляемой с помощью приёма, известного как «разделение времени». При этом специальный компонент, называемый планировщиком, «нарезает» процессорное время на короткие отрезки и предоставляет их поочередно различным исполняющимся программам (процессам);

наконец, оператор должен иметь возможность, так или иначе, управлять процессами выполнения отдельных программ. Для этого служат операционные среды, одна из которых — оболочка и набор стандартных утилит — является частью ОС (прочие, такие, как графическая операционная среда, образуют независимые от ОС прикладные платформы). Таким образом, современные универсальные ОС можно охарактеризовать, прежде всего, как

1. использующие файловые системы (с универсальным механизмом доступа к данным),
2. многопользовательские (с разделением полномочий),
3. многозадачные (с разделением времени).

Многозадачность и распределение полномочий требуют определённой иерархии привилегий компонентов самой ОС. В составе ОС различают три группы компонентов:

ядро, содержащее планировщик; драйверы устройств, непосредственно управляющие оборудованием; сетевую подсистему, файловую систему;

системные библиотеки и

оболочка с утилитами.

Большинство программ, как системных (входящих в ОС), так и прикладных, исполняются в непривилегированном («пользовательском») режиме работы процессора и

получают доступ к оборудованию (и, при необходимости, к другим ядерным ресурсам, а также ресурсам иных программ) только посредством системных вызовов. Ядро исполняется в привилегированном режиме: именно в этом смысле говорят, что ОС (точнее, её ядро) управляет оборудованием.

В определении состава ОС значение имеет критерий операциональной целостности (замкнутости): система должна позволять полноценно использовать (включая модификацию) свои компоненты. Поэтому в полный состав ОС включают и набор инструментальных средств (от текстовых редакторов до компиляторов, отладчиков и компоновщиков).

В 1950-60-х годах сформировались и были реализованы основные идеи, определяющие функциональность ОС: пакетный режим, разделение времени и многозадачность, разделение полномочий, реальный масштаб времени, файловые структуры и файловые системы.

Пакетный режим

Необходимость оптимального использования дорогостоящих вычислительных ресурсов привела к появлению концепции «пакетного режима» исполнения программ. Пакетный режим предполагает наличие очереди программ на исполнение, причём ОС может обеспечивать загрузку программы с внешних носителей данных в оперативную память, не дожидаясь завершения исполнения предыдущей программы, что позволяет избежать простоя процессора.

Разделение времени и многозадачность

Уже пакетный режим в своём развитом варианте требует разделения процессорного времени между выполнением нескольких программ.

Необходимость в разделении времени (многозадачности, мультипрограммировании) проявилась ещё сильнее при распространении в качестве устройств ввода-вывода телетайпов (а позднее, терминалов с электронно-лучевыми дисплеями) (1960-е годы). Поскольку скорость клавиатурного ввода (и даже чтения с экрана) данных оператором много ниже, чем скорость обработки этих данных компьютером, использование компьютера в «монопольном» режиме (с одним оператором) могло привести к простою дорогостоящих вычислительных ресурсов.

Разделение времени позволило создать «многопользовательские» системы, в которых один (как правило) центральный процессор и блок оперативной памяти соединялся с многочисленными терминалами. При этом часть задач (таких, как ввод или редактирование данных оператором) могла исполняться в режиме диалога, а другие задачи (такие, как массивные вычисления) — в пакетном режиме.

Разделение полномочий

Распространение многопользовательских систем потребовало решения задачи разделения полномочий, позволяющей избежать возможности модификации исполняемой

программы или данных одной программы в памяти компьютера другой (содержащей ошибку или злонамеренно подготовленной) программы, а также модификации самой ОС прикладной программой.

Реализация разделения полномочий в ОС была поддержана разработчиками процессоров, предложивших архитектуры с двумя режимами работы процессора — «реальным» (в котором исполняемой программе доступно всё адресное пространство компьютера) и «защищённым» (в котором доступность адресного пространства ограничена диапазоном, выделенном при запуске программы на исполнение).

Реальный масштаб времени

Применение универсальных компьютеров для управления производственными процессами потребовало реализации «реального масштаба времени» («реального времени») — синхронизации исполнения программ с внешними физическими процессами.

Включение функции реального масштаба времени в ОС позволило создавать системы, одновременно обслуживающие производственные процессы и решающие другие задачи (в пакетном режиме и (или) в режиме разделения времени).

6.3 Операционная система DOS.

DOS (англ. Disk Operating System — дисковая операционная система, ДОС) — семейство операционных систем для персональных компьютеров. Ориентированно на использование дисковых накопителей, таких как жёсткий диск и дискета.

Существовали операционные системы с таким названием для больших ЭВМ производства IBM и их клонов в 1960-80-х годах.

DOS для IBM PC-совместимых компьютеров

DOS является однозадачной операционной системой. После запуска управление передаётся прикладной программе, которая получает в своё распоряжение все ресурсы компьютера и может осуществлять ввод/вывод посредством как функций предоставляемых операционной системой, так и функций базовой системы ввода/вывода (BIOS), а также работать с устройствами напрямую.

DOS имеет консольную систему ввода/вывода и поддерживает три стандартных потока: stdin, stdout и stderr.

DOS — 16-битная операционная система, работающая в реальном режиме, поэтому для расширения возможностей и преодоления ограничений реального режима были созданы так называемые расширители DOS. Они запускают программы в защищённом 32-битном режиме и эмулируют исходные сервисы операционной системы. Обычно они поддерживают стандарт

DOS Protected Mode Interface (DPMI). Самый известный и широко используемый (в компьютерных играх) расширитель — DOS/4GW.

Существует несколько ветвей ДОС для ПК. Все они схожи по наборам команд и базовой функциональности, но отличаются производительностью, стабильностью работы и дополнительными функциями.

DR-DOS (Novell DOS, Caldera DR-DOS) — выпущена Digital Research в 1991 году, перекуплена компанией Novell в 1993 году, затем компанией Caldera.

MS-DOS — выпущена компанией Microsoft в 1981 году.

PC DOS — выпущена компанией IBM в 1981 году.

PTS-DOS — выпущена компанией ФизТехСофт в 1991 году или ранее.

Paragon DOS Pro (первоначальное название — PT$-DOS). Ветка PTS-DOS, выпущенная компанией Paragon Software после того, как её основатели, включая ведущего разработчика PTS-DOS, ушли из ФизТехСофт, основав собственную компанию. Последние версии этой ветки включают поддержку FAT32.

FreeDOS — выпущена в 1994 году. Свободная ДОС, изначально называлась PD-DOS.

FreeDOS-32 — свободная 32-битная ДОС. Не требует расширителей для запуска 32-битных приложений. Планируется избавиться и от других ограничений ДОС (поддержка других файловых систем, многозадачности и т. п.).

MS-DOS (англ. Microsoft Disk Operating System — дисковая ОС от Microsoft) — коммерческая операционная система фирмы Microsoft для персональных компьютеров. MS-DOS — самая известная ОС из семейства DOS, ранее устанавливаемая на большинство IBM PC-совместимых компьютеров. Со временем она была вытеснена ОС семейства Windows 9x и Windows NT.

MS-DOS была создана в 1981 году и, в ходе её развития, было выпущено восемь крупных версий (1.0, 2.0 и т. д.) и два десятка промежуточных (3.1, 3.2 и т. п.), пока в 2000 году Microsoft не прекратила её разработку. Это был ключевой продукт фирмы, дававший ей существенный доход и маркетинговый ресурс, в ходе развития Microsoft от разработчика языка программирования до крупной компании, производящей самое разнообразное программное обеспечение.

Последняя официальная версия 6.22. Однако существует версия 7.1 в виде ядра Windows 98, которая загружается на начальном этапе загрузки системы.

6.4 Что понимается под файлом.

Файл (англ. file — папка, скоросшиватель) — концепция в вычислительной технике: сущность, позволяющая получить доступ к какому-либо ресурсу вычислительной системы и обладающая рядом признаков:

фиксированное имя (последовательность символов, число или что-то иное, однозначно характеризующее файл);

определённое логическое представление и соответствующие ему операции чтения/записи.

Может быть любой — от последовательности бит до базы данных с произвольной организацией или любым промежуточным вариантом.

Первому случаю соответствуют операции чтения/записи потока и/или массива (то есть последовательные или с доступом по индексу), второму — команды СУБД. Промежуточные варианты — чтение и разбор всевозможных форматов файлов.

В отличие от переменной, файл (в частности, его имя) имеет смысл вне конкретной программы. Работа с файлами — по крайней мере, в «простейшем» представлении — реализуется средствами операционных систем, а до их появления реализовывалась их предшественниками — мониторами и библиотеками подпрограмм.

Ресурсами, доступными через файлы, в принципе, может быть что угодно, представимое в цифровом виде. Чаще всего в их перечень входят:

области данных (необязательно на диске);

устройства (как физические, так и виртуальные);

потоки данных (в частности, вход или выход процесса);

сетевые ресурсы;

объекты операционной системы.

Файлы первого типа исторически возникли первыми и распространены наиболее широко, поэтому часто «файлом» называют и область данных, соответствующую имени.

Файловая система

По мере развития вычислительной техники файлов в системах становилось всё больше. Для удобства работы с ними их, как и другие данные, стали организовывать в структуры (тогда же появились символьные имена). Вначале это был простой массив, «привязанный» к конкретному носителю информации. В настоящее время наибольшее распространение получила древовидная организация с возможностью монтирования и вставки дополнительных связей (т. е. ссылок). Соответственно, имя файла приобрело характер пути к файлу:

перечисление узлов дерева файловой системы, которые нужно пройти, чтобы до него добраться.

Фа́йловая систе́ма (англ. file system) — регламент, определяющий способ организации, хранения и именования данных на носителях информации. Она определяет формат физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имени файла, максимальный возможный размер файла, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

Файловая система связывает носитель информации, с одной стороны, и API для доступа к файлам — с другой. Когда прикладная программа обращается к файлу, она не имеет никакого представления о том, каким образом расположена информация в конкретном файле, так же, как и на каком физическом типе носителя (CD, жёстком диске, магнитной ленте или блоке флеш-памяти) он записан. Всё, что знает программа — это имя файла, его размер и атрибуты. Эти данные она получает от драйвера файловой системы. Именно файловая система устанавливает, где и как будет записан файл на физическом носителе (например, жёстком диске).

С точки зрения операционной системы, весь диск представляет из себя набор кластеров размером от 512 байт и выше. Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

Однако файловая система не обязательно напрямую связана с физическим носителем информации. Существуют виртуальные и сетевые файловые системы, которые являются лишь способом доступа к файлам, находящимся на удалённом компьютере.

```
C:
  \Program files
     \CDEx
        \CDEx.exe
        \CDEx.hlp
        \mppenc.exe
  \Мои документы
     \Wiki.txt
     \Tornado.jpg
D:
```

```
\Music
    \ABBA
        \1974 Waterloo
        \1976 Arrival
            \Money, Money, Money.ogg
        \1977 The Album
```

(Иерархическая файловая система Windows/DOS)

```
/usr
    /bin
        /arch
        /ls
        /raw
    /lib
        /libhistory.so.5.2
        /libgpm.so.1
/home
    /lost+found
        /host.sh
    /guest
        /Pictures
            /example.png
        /Video
            /matrix.avi
            /news
                /lost_ship.mpeg
```

(Иерархическая файловая система Unix и UNIX-подобных операционных системах)

Обратите внимание на использование слешей в файловых системах Windows, UNIX и UNIX-подобных операционных системах (В Windows используется обратный слеш «\», а в UNIX и UNIX-подобных операционных системах простой слеш «/»).

Имя файла

В большинстве файловых систем имя файла используется для указания к какому именно файлу производится обращение. В различных файловых системах ограничения на имя файла сильно различаются:

В FAT16 и FAT12 размер имени файла ограничен 8 символами (3 символа расширения).

В VFAT ограничение 255 байт.

В FAT32, HPFS имя файла ограниченно 255 символами

В NTFS имя ограничено 254 символами Unicode

В ext2/ext3 ограничение 255 байт.

Помимо ограничений файловой системы, интерфейсы операционной системы дополнительно ограничивают набор символов, который допустим при работе с файлами.

Для MS-DOS в имени файла допустимы только заглавные буквы, цифры. Не допустим пробел, знак вопроса, звёздочка, символы больше/меньше, символ вертикальной черты.[1]. При вызове системных функций именами файлов в нижнем или смешанном регистре, они приводятся к верхнему регистру.

Для Microsoft Windows в имени файла разрешены заглавные и строчные буквы, цифры, некоторые знаки препинания, пробел. Запрещены символы «>», «<», «|», «?», «*», «/», «\», «:», «"».

Для GNU/Linux (с учётом возможности маскировки) разрешены все символы, кроме «/» и нулевого байта.

Большинство операционных систем требуют уникальности имени файла в одном каталоге, хотя некоторые системы допускают файлы с одинаковыми именами (например, при работе с ленточными накопителями).

Расширение имени файла

Расширение имени файла (часто расширение файла или расширение) как самостоятельный атрибут файла существует в файловых системах FAT16, FAT32,NTFS, используемых операционными системами MS DOS, DR DOS, PC DOS, MS Windows и используется для определения типа файла.Оно позволяет системе определить, каким приложением следует открывать данный файл. По умолчанию в операционной системе Windows расширение скрыто от пользоватея. В остальных файловых системах расширение — условность, часть имени, отделённая самой правой точкой в имени.

Атрибуты

В некоторых файловых системах предусмотрены атрибуты (обычно это бинарное значение «да»/«нет», кодируемое одним битом). Практически атрибуты не влияют на возможность доступа к файлам, для этого в некоторых файловых системах существуют права доступа.

READ ONLY - только для чтения - в файл запрещено писать

SYSTEM – системный - критический для работы операционной системы файл

HIDDEN – скрытый - файл скрывается от показа, пока явно не сказано обратное

ARCHIVE - архивный(требующий архивации) - файл изменён после резервного копирования или не был скопирован программами резервного копирования

Для файла могут быть определены следующие временные метки:

Время создания

Время модификации

Время последнего доступа

Владелец и группа файла

В некоторых файловых системах предусмотрено указание на владельца файла, и группу владельца.

Права доступа

В некоторых файловых системах предусмотрена возможность для ограничения доступа пользователей к содержимому файла

В UNIX-подобных операционных системах для файлов обычно выделяют три типа прав:

Право на запись

Право на чтение

Право на выполнение

Каждое право задаётся раздельно для владельца, для группы и для всех остальных. ACL позволяют расширить этот список.

В операционных системах Windows NT при работе с файловой системой NTFS права доступа задаются явно для пользователей или групп (или наследуются от вышестоящих объектов). Права в себя включают:

Право на чтение

Право на запись

Право на исполнение

Право на удаление

Право на смену атрибутов и владельца

Право на создание, удаление подпапок (для папок)

Право на чтение прав доступа

Каждое право может быть задано как разрешением, так и запретом, запрет имеет больший приоритет, чем разрешение.

Операции с файлом

Условно можно выделить два типа операций с файлом - связанные с его открытием, и выполняющиеся без его открытия. Операции первого типа обычно служат для чтения/записи информации или подготовки к записи/чтению. Операции второго типа выполняются с файлом

как с "объектом" файловой системы, в котором файл является мельчайшей единицей структурирования.

Операции, связанные с открытием файла

В зависимости от операционной системы те или иные операции могут отсутствовать.

Обычно выделяют дополнительные сущности, связанные с работой с файлом:

хэндлер файла, или дескриптор (описатель). При открытии файла (в случае, если это возможно), операционная система возвращает число (или указатель на структуру), с помощью которого выполняются все остальные файловые операции. По их завершению файл закрывается, а хэндлер теряет смысл.

файловый указатель. Число, являющееся смещением относительно нулевого байта в файле. Обычно по этому адресу осуществляется чтение/запись, в случае, если вызов операции чтения/записи не предусматривает указание адреса. При выполнении операций чтения/записи файловый указатель смещается на число прочитанных (записанных) байт. Последовательный вызов операций чтения таким образом позволяет прочитать весь файл не заботясь о его размере.

файловый буфер. Операционная система (и/или библиотека языка программирования) осуществляет кэширование файловых операций в специальном буфере (участке памяти). При закрытии файла буфер сбрасывается.

режим доступа. В зависимости от потребностей программы, файл может быть открыт на чтение и/или запись. Кроме того, некоторые операционные системы (и/или библиотеки) предусматривают режим работы с текстовыми файлами. Режим обычно указывается при открытии файла.

режим общего доступа. В случае многозадачной операционной системы возможна ситуация, когда несколько программ одновременно хотят открыть файл на запись и/или чтение. Для регуляции этого существуют режимы общего доступа, указывающие на возможность осуществления совместного доступа к файлу (например, файл в который производится запись может быть открыт для чтения другими программами - это стандартный режим работы log-файлов).

Операции

Открытие файла (обычно в качестве параметров передается имя файла, режим доступа и режим совместного доступа, а в качестве значения выступает файловый хэндлер или дескриптор), кроме того обычно имеется возможность в случае открытия на запись указать на то, должен ли размер файла изменяться на нулевой.

Закрытие файла. В качестве аргумента выступает значение, полученное при открытии файла. При закрытии все файловые буферы сбрасываются.

Запись — в файл помещаются данные.

Чтение — данные из файла помещаются в область памяти.

Перемещение указателя — указатель перемещается на указанное число байт вперёд/назад или перемещается по указанному смещению относительно начала/конца. Не все файлы позволяют выполнение этой операции (например, файл на ленточном накопителе может не «уметь» перематываться назад).

Сброс буферов — содержимое файловых буферов с незаписанной в файл информацией записывается. Используется обычно для указания на завершение записи логического блока (для сохранения данных в файле на случай сбоя).

Получение текущего значения файлового указателя.

Операции, не связанные с открытием файла

Операции, не требующие открытия файла оперируют с его «внешними» признаками — размером, именем, положением в дереве каталогов. При таких операциях невозможно получить доступ к содержимому файла, файл является минимальной единицей деления информации.

В зависимости от файловой системы, носителя информации, операционной системой часть операций может быть недоступна.

Список операций с файлами

Удаление файла

Переименование файла

Копирование файла

Перенос файла на другую файловую систему/носитель информации

Создание симлинка или хардлинка

Получение или изменение атрибутов файла

Типы файлов

В различных операционных и/или файловых системах могут быть реализованы различные типы файлов; кроме того, реализация различных типов может различаться.

«Обыкновенный файл» — файл, позволяющий операции чтения, записи, перемещения внутри файла

Директория (англ. directory — алфавитный справочник, часто переводится как каталог) — файл, содержащий записи о входящих в него файлах. Директории могут содержать записи о других директориях, образуя древовидную структуру.

Жёсткая ссылка (англ. hardlink, часто используется калька хардлинк) — в общем случае, одна и та же область информации может иметь несколько имён, указывающих на одни и те же данные. В таком случае имена называют жёсткими ссылками (хардлинками). В общем случае после создания хардлинка сказать где «настоящий» файл а где хардлинк невозможно, так как

имена равноправны. Сама область данных существует до тех пор пока существует хотя бы одно из имён. Хардлинки возможны только на одном физическом носителе.

Символьная ссылка (симлинк, софтлинк) — файл, содержащий в себе ссылку на другой файл или директорию. Может ссылаться на любой элемент файловой системы, в том числе, и расположенный на другом физическом носителе.

Логический диск или том (англ. volume) — часть долговременной памяти компьютера, рассматриваемая как единое целое для удобства работы. Термин «логический диск» используется в противоположность «физическому диску», под которым рассматривается долговременная память одного конкретного дискового носителя.

Для операционной системы не имеет значения, где располагаются данные — на лазерном диске, в разделе жёсткого диска, или во флеш-памяти. Для унификации представляемых участков долговременной памяти вводится понятие логического диска.

В дисковых операционных системах (например, MS-DOS) и производных от них (например, MS Windows) логические диски обозначаются буквами латинского алфавита. Каждый том имеет собственную файловую систему.

Помимо хранимой информации, том содержит описание файловой системы — как правило, это таблица с перечислением всех файлов и их атрибутов (Таблица размещения файлов). По этой таблице определяется, в частности, в каком каталоге (папке) находится тот или иной файл. Благодаря этому при переносе файла из одной папки в другую в пределах одного тома, не осуществляется перенос данных из одной части физического диска на другую, а просто меняется запись в таблице размещения файлов. Если же файл переносится с одного логического диска на другой (даже если оба логических диска расположены на одном физическом диске), обязательно будет происходить физический перенос данных (копирование с дальнейшим удалением оригинала в случае успешного завершения).

По этой же причине форматирование и дефрагментация каждого логического диска не затрагивает другие.

В UNIX-подобных операционных системах обозначения жёстких дисков и разделов на них несколько отличаются от видимых пользователю в Windows. В Linux диски получают буквенное обозначение типа sdX, где X соответствует номеру из последовательности a, b,... а разделы на устройствах нумеруются и обозначаются цифрами, причём нумерация логических разделов, которые в Windows соответствуют логическим дискам в расширенном разделе, начинается с 5, так как номера 1-4 зарезервированы для обозначения первичных разделов и, собственно, расширенного раздела.

Например, обозначения разделов для ОС Windows будет sda1 (для C:) и sda5 (для D:). Если бы было четыре основных раздела или два основных и два логических (пусть C:, D:, E:, F:) то в первом случае они обозначались бы как sda1 - sda4, а во втором как sda1, sda2, sda5, sda6, соответственно.

Чтобы было удобнее работать с разделами на жёстком диске, в UNIX-подобных операционных системах их монтируют в каталоги корневой файловой системы, обозначаемой /, которая обязана существовать. Более того, системой реализуется принцип: любое устройство есть файл, и жёсткие диски, как и остальные устройства компьютера, также являются файлами и доступны в каталоге dev корневой файловой системы. Отсюда и полное обозначение жёсткого диска /dev/sda.

Так-же, в UNIX-подобных операционных системах все логические диски должны иметь точку монтирования. Точка монтирования соответствует определенному каталогу файловой системы. Дерево каталогов логического диска представляется поддеревом файловой системы, включенным в него в точке монтирования. Логический диск может быть примонтирован к любому каталогу существующей файловой системы. В свою очередь, к любому каталогу на подмонтированном носителе можно подмонтировать еще один носитель и т.д. Пути, используемому в качестве точки монтирования, должен соответствовать пустой каталог (хотя, например, в системах на базе FreeBSD и Linux, если каталог не пуст, его содержимое просто замещается содержимым логического диска). Хотя логический том можно примонтировать куда угодно, сменные носители (флешки, компакт-диски и т.п.) принято монтировать к подкаталогам папок /mnt или /media. В настольных дистрибутивах Linux этот процесс обычно происходит автоматически. При этом в каталоге /media (/mnt) создается подкаталог, имя которого совпадает с именем монтируемого тома.

Для управления точками монтирования логических дисков UNIX-подобные операционные системы предоставляют команду «mount».

Пример: Если компакт-диск, содержащий файл «info.txt», был смонтирован в каталог «/mnt/iso9660», то этот файл будут доступен как «/mnt/iso9660/info.txt».

Тома и разделы в дисковых ОС Microsoft

Том — это не то же самое, что раздел диска. Например, информация на гибком диске является информацией одного тома, разделов же на гибком диске не создают.

Вот один из примеров — рассмотрен компьютер, в котором имеется один дисковод гибких дисков (со вставленной дискетой) и два жёстких диска. Первый жёсткий диск разбит на два раздела, а на втором выделен только один.

Директория (англ. directory - справочник, указатель), син. каталог, папка — сущность в файловой системе, упрощающая организацию файлов. Типичная файловая система содержит большое количество файлов, и директории помогают упорядочить её путём их группировки. Например, в каждом каталоге (директории) MS-DOS есть специальные символы «.» точка и «..» две точки обозначающие текущий каталог и родительский каталог, используя эти специализированные названия можно перейти в соответствующую директорию.

Термин «Папка»

Термин папка был введён для упрощения файловой системы в глазах пользователя путём аналогии с офисными папками. Он был впервые использован в Mac OS, а в системах семейства Microsoft Windows он появился с выходом Windows 95 [1]. Эта метафора на сегодня используется в большом числе операционных систем: Windows NT, Mac OS, Mac OS X, а также в большом количестве сред рабочего стола для систем семейства UNIX (например, в KDE или GNOME).

В этой терминологии, папка, находящаяся в другой папке, называется подпапка или вложенная папка. Все вместе, папки на компьютере представляют иерархическую структуру, представляющую собой дерево каталогов. Подобная древообразная структура возможна в операционных системах, не допускающих существование «физических линков» (DOS и старые версии Windows допускали только аналог символических линков — Shortcut (Ярлык)). В общем случае файловая система представляет собой ориентированный граф.

Директория которая не является поддиректорией ни одной другой директории называется корневой. Это значит, что эта директория (папка) находится на самом верхнем уровне иерархии всех директорий. В Linux системах - корневая директория обозначается как правило "/", в Windows каждый из дисков имеет свою корневую директорию C:\, D:\ и т. д. Папки в Windows бывают системные (служебные, созданные ОС) и пользовательские (созданные пользователем). Все папки, создаваемые пользователем, по умолчанию имеют одинаковые значки, системные же папки обычно имеют разные значки. Пример системных папок: «Рабочий стол», «Корзина», «Сетевое окружение», «Панель управления», папки логических дисков и т. п.

Иерархия папок в Microsoft Windows

В иерархии папок Windows системная папка «Рабочий стол» является самой главной папкой верхнего уровня, содержащей все остальные папки компьютера. В Windows 4.x она соответствует директории «C:\WINDOWS\Рабочий стол» В папке «Рабочий стол» находятся системные папки «Корзина» («C:\RECYCLE»), «Сетевое окружение», «Мой компьютер» и

созданные пользователем папки. В папке «Мой компьютер» находятся системные папки дисков всех устройств для хранения информации, подключенных к компьютеру (дисководы гибких дисков, жесткие диски, CD-ROM и т. д.). Папки дисков обозначаются именами этих дисков, как в DOS — буквами латинского алфавита от «A:\» до «Z:\». Буквы «A:\» и «B:\», как правило, используются только для дисководов гибких дисков. Начиная с буквы «C:\» идут папки жестких дисков, логических, сетевых и внешних дисков, CD и DVD приводов и т. д.

6.5 Задание

Необходимо освоить основные принципы работы с командной строкой в Windows. Для того, чтобы запустить командную строку необходимо в операционной системе Windows выбрать Пуск+Выполнить, в качестве запускаемой программе указать cmd и запустить на выполнение нажав Ok.

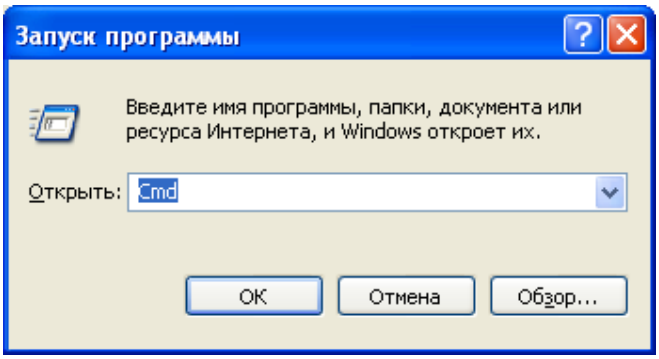


Рисунок 6.1 - Запуск программ

После запуска отобразится командная строка, где ввод команд осуществляется с клавиатуры. Соответствующее описание команд можно найти, пользуясь командой help, то есть необходимо ввести в строку help и нажать Enter. Более подробное описание каждой команды можно прочитать введя help и далее имя интересующей команды, например:

>help help

>help dir

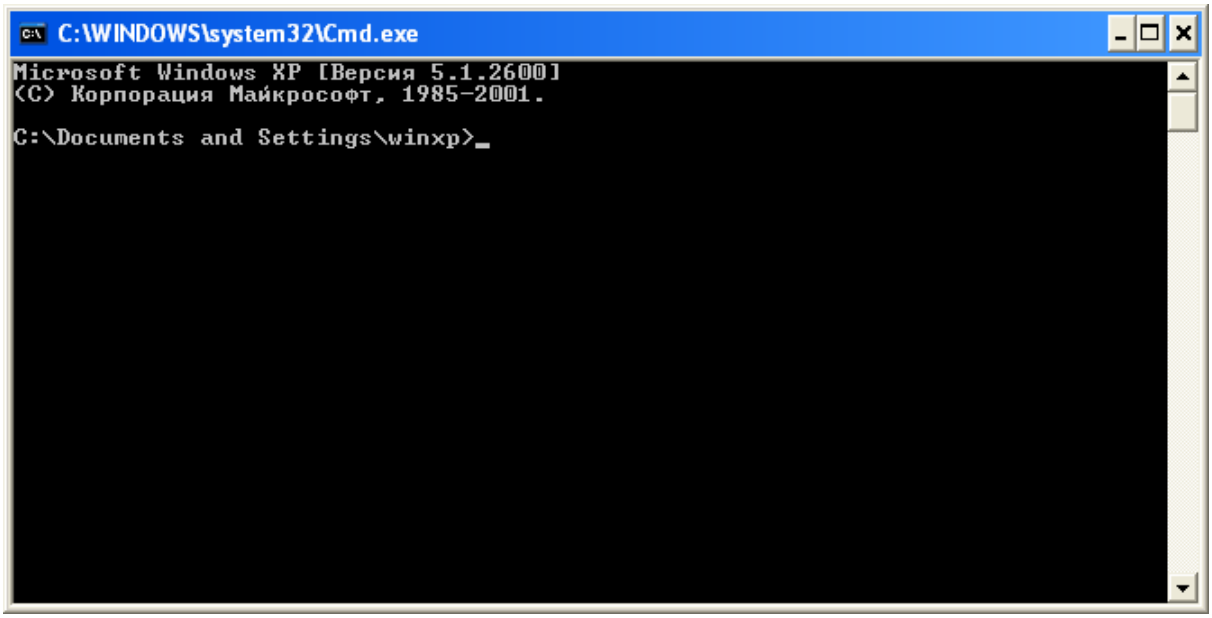


Рисунок 6.2 - Командная строка

На представленном рисунке виден черный экран и строка в верхней левой части окна с указанием операционной системы, ниже указан путь к текущему каталогу. Первая буква и двоеточие C: - означают логический диск или том, далее через слэш указано имя папки или директории (каталога) Documents and Settings на диске C, и далее через слэш указана папка winxp содержащаяся в директории Documents and Settings.

1. Проверить и установить дату и время, команды DATE и TIME. Просмотреть сначала параметры команд с помощью

>help date

>help time

Любая команда имеет входные параметры, которые перечисляются через пробел, среди параметров в особую группу входят параметры ключи, которые определяют особые режимы работы каждой команды, например указав какой-то ключ, в команде включается дополнительная опция и при ее выполнении срабатывает еще какое-либо дополнительное действие. Обычно ключ указывается с помощью обратного слэша и какой-либо буквы.

При вызове help <имя команды> указываются все ключи данной команды и режимы ее работы, если параметры заключены в квадратные скобочки значит они являются необязательными и их можно опустить, естественно, что если вы хотите включить данный параметр или ключ, то когда вы его указываете при выполнении команды квадратные скобочки не ставятся.

Пример help date.

Вывод или изменение даты.

DATE [/T | дата]

Команда DATE без параметров отображает текущую дату и запрашивает ввод новой даты. Для сохранения текущей даты нажмите клавишу ENTER.

Когда расширенная обработка команд включена, команда DATE поддерживает ключ /T, позволяющий просто вывести текущее значение даты без запроса новой даты.

Пример команды с ключом.

>Date /T

Значок | - обозначает или, то есть вы можете в качестве параметра указать или ключ /T или дату в формате даты.

>Date 10.10.2012.

2. Определить версию ОС, команда VER.

3. Сделать свой рабочий диск текущим (команда CD).

Для смены диска можно воспользоваться командой:

имя диска:

Например,

>z:

Команда cd позволяет сменить диск только при использовании ключа /D, в ином случае диск не меняется.

Вывод имени либо смена текущего каталога.

CHDIR [/D] [диск:][путь]

CHDIR [..]

CD [/D] [диск:][путь]

CD [..]

cd .. обозначает переход в родительский каталог.

Например, в каталоге home есть каталог me и all, в каталоге me есть каталог inf и text, пусть текущий каталог inf. Тогда ..\text путь к каталогу text в каталоге me, ..\..\all – путь к каталогу all из папки inf.

Таким образом, две точки (..) указывает на каталог выше по уровню.

Точка (.) указывает на текущий каталог. .\ - ссылается на каталог inf.

Команда CD диск: отображает имя текущего каталога указанного диска.

Команда CD без параметров отображает имена текущих диска и каталога.

Параметр /D используется для одновременной смены текущих диска и каталога.

Изменение команды CHDIR при включении расширенной обработки команд:

Имя текущего каталога в строке вызова преобразуется к тому же регистру символов, что и для существующих имен на диске. Так, команда CD C:\TEMP на самом деле сделает текущим каталог C:\Temp, если он существует на диске.

Команда CHDIR перестает рассматривать пробелы как разделители, что позволяет перейти в подкаталог, имя которого содержит пробелы, не заключая все имя каталога в кавычки. Например:

cd \winnt\profiles\username\programs\start menu

приводит к тому же результату, что и:

cd "\winnt\profiles\username\programs\start menu"

При отключении расширенной обработки команд используется только второй вариант.

4. Создать на своем диске каталог и в нем два подкаталога (MKDIR).

Создание каталога.

MKDIR [диск:]путь

MD [диск:]путь

Изменение команды MKDIR при включении расширенной обработки команд:

Команда MKDIR создает при необходимости все промежуточные каталоги в пути. Например, если \a не существует, то:

mkdir \a\b\c\d

приводит к тому же результату, что и:

mkdir \a
chdir \a
mkdir b
chdir b
mkdir c
chdir c
mkdir d

При отключении расширенной обработки команд используется только второй вариант.

5. Просмотреть дерево каталогов вашего диска (TREE, DIR). Сделать текущим один из каталогов. Для создания каталогов с именами содержащими пробелы необходимо брать имена в двойные кавычки, например: «Моя папка». Самостоятельно просмотрите параметры команды Tree.

DIR

Вывод списка файлов и подкаталогов из указанного каталога.

DIR [диск:][путь][имя_файла] [/A[[:]атрибуты]] [/B] [/C] [/D] [/L] [/N] [/O[[:]порядок]] [/P] [/Q] [/S] [/T[[:]время]] [/W] [/X] [/4]

[диск:][путь][имя_файла]

Диск, каталог и/или файлы, которые следует включить в список.

/A Вывод файлов с указанными атрибутами.

атрибуты D Каталоги, R Доступные только для чтения, H Скрытые файлы, A Файлы для архивирования, S Системные файлы Префикс "-" имеет значение НЕ

/B Вывод только имен файлов.

/C Применение разделителя групп разрядов для вывода размеров файлов (по умолчанию). Для отключения этого режима служит ключ /-C.

/D Вывод списка в несколько столбцов с сортировкой по столбцам.

/L Использование нижнего регистра для имен файлов.

/N Отображение имен файлов в крайнем правом столбце.

/O Сортировка списка отображаемых файлов.

порядок N По имени (алфавитная) S По размеру (сперва меньшие)

E По расширению (алфавитная) D По дате (сперва более старые) G Начать список с каталогов Префикс "-" обращает порядок

/P Пауза после заполнения каждого экрана.

/Q Вывод сведений о владельце файла.

/S Вывод списка файлов из указанного каталога и его подкаталогов.

/T Выбор поля времени для отображения и сортировки время C Создание

A Последнее использование

W Последнее изменение

/W Вывод списка в несколько столбцов.

/X Отображение коротких имен для файлов, чьи имена не соответствуют стандарту 8.3. Формат аналогичен выводу с ключом /N, но короткие имена файлов выводятся слева от длинных. Если короткого имени у файла нет, вместо него выводятся пробелы.

/4 Вывод номера года в четырехзначном формате

Стандартный набор ключей можно записать в переменную среды DIRCMD. Для отмены их действия введите в команде те же ключи с префиксом "-", например: /-W.

6. Установить текущим каталог windows на диске C:. Просмотреть список всех файлов этого каталога и файлов типа .COM в режиме постраничного вывода (команды DIR, MORE). Использовать маски файлов – «*» - любая последовательность букв и цифр, «?» – любой

символ. Например, любые файлы с любыми расширениями из двух символов будет представлены как следующая последовательность символов

*.??,

любой файл с первой буквой f, следующими двумя любыми буквами, затем буквой a и затем любой последовательностью символов, любой длины и расширением txt будет выглядеть как

f??a*.txt.

Таким образом, одна операция может быть проведена над группой файлов, которые соответствуют какой-то маске. Например, *.* - все файлы текущего каталога с любым расширением, file*.txt — все файлы начинающиеся с последовательности символов file и расширением txt, * - любые файлы.

Команда MORE позволяет вывести данные выводящиеся на консоль (экран) в постраничном режиме, например, когда данных очень много, чтобы они не все скопом были отображены и часть из них из-за ограничения количества буфера строк стали не видимы пользователю. Команда More позволяет вывести как результаты команды в постраничном режиме, так и содержание файла. В первом случае сначала указывается команда, затем вертикальная черта | и команда More. Вертикальная черта определяет конвейерное выполнение, когда результат первой команды поступает на вход другой и обрабатывается ей, можно реализовать несколько конвейеров proc1|proc2|proc3| … procN. В этом случае каждая последующая команда использует результаты предыдущей.

Допустим dir | more. Результаты команды dir обрабатываются командой more, как вы теперь знаете команда more выводит данные постранично.

Последовательный вывод данных по частям размером в один экран.

MORE [/E [/C] [/P] [/S] [/Tn] [+n]] < [диск:][путь]имя_файла имя_команды | MORE [/E [/C] [/P] [/S] [/Tn] [+n]]

MORE /E [/C] [/P] [/S] [/Tn] [+n] [файлы]

[диск:][путь]имя_файла Файл, отображаемый

по фрагментам.

имя_команды Команда, вывод которой

отображается на экране.

/E Разрешение использования дополнительных возможностей.

/C Очистка экрана перед выводом каждой страницы.

/P Учет символов перевода страницы.

/S Сжатие нескольких пустых строк в одну строку.

/Tn Замена символов табуляции n пробелами (по умолчанию n = 8).

Стандартный набор ключей можно поместить в переменную среды MORE.

+n Начало вывода первого файла со строки с номером n. файлы Список отображаемых файлов. Для разделения имен файлов в списке используйте пробелы.

Если использование дополнительных возможностей разрешено, в ответ на приглашение -- More -- можно вводить следующие команды:

P n Вывод следующих n строк.

S n Пропуск следующих n строк.

F Вывод следующего файла.

Q Завершение работы.

= Вывод номера строки.

? Вывод строки подсказки.

<пробел> Вывод следующей страницы.

<ENTER> Вывод следующей строки.

7. Скопировать все файлы с диска (или какой либо папки содержащей файлы) C: расширением .bat и .txt в один из созданных подкаталогов на вашем диске (COPY и маска). Скопировать все файлы, имеющие в своем названии букву а, скопировать все файлы, имеющие начальной букву, совпадающую с первой буквой вашего имени в латинской транскрипции. Скопировать все файлы, имеющие в названии слово te и имеющие вначале три любых символа и в конце имени имеющие любую последовательность символов, учесть файлы имеющие и не имеющие расширения. Выдать на экран список скопированных файлов.

Команда Copy позволяет скопировать файлы из источника в приемник. Например,

copy *.jpg z:\text - копировать файлы с расширением jpg из текущей папки в папку text диска z.

Копирование одного или нескольких файлов в другое место.

COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/A | /B] источник [/A | /B]

[+ источник [/A | /B] [+ ...]] [результат [/A | /B]]

источник Имена одного или нескольких копируемых файлов.

/A Файл является текстовым файлом ASCII.

/B Файл является двоичным файлом.

/D Указывает на возможность создания зашифрованного файла результат Каталог и/или имя для конечных файлов.

/V Проверка правильности копирования файлов.

/N Использование, если возможно, коротких имен при копировании файлов, чьи имена не удовлетворяют стандарту 8.3.

/Y Подавление запроса подтверждения на перезапись существующего конечного файла.

/-Y Обязательный запрос подтверждения на перезапись существующего конечного файла.

/Z Копирование сетевых файлов с возобновлением.

Ключ /Y можно установить через переменную среды COPYCMD.

Ключ /-Y командной строки переопределяет такую установку.

По умолчанию требуется подтверждение, если только команда COPY не выполняется в пакетном файле.

Чтобы объединить файлы, укажите один конечный и несколько исходных файлов, используя подстановочные знаки или формат "файл1+файл2+файл3+...".

8. С терминала ввести в файл MYFILE.TXT несколько строк текста со своими анкетными данными, ФИО, место и дата рождения, факультет, группа.

COPY CON <имя файла>

Копирование в файл с консоли ввода (с клавиатуры).

Ввод строки завершается вводом Enter, ввод файла завершается вводом Ctrl+Z и Enter.

COPY <имя файла> CON

COPY CON CON – ввод с клавиатуры и вывод на экран.

Вывод на консоль содержимого файла (вывод на экран)

CON, PRN, COM1, COM2, COM3, COM4, LPT1, LPT2 — обозначают специализированные имена файлов относящиеся к устройствам. Например, PRN — устройство принтера, так как prn воспринимается как имя файла, то при копировании и записи в этот файл, будет осуществляться последовательная печать на принтере. Аналогично когда мы производим копирование файла в файл CON, производится вывод на экран, так как CON — это консоль (ввод с клавиатуры, вывод на экран). COM1, COM2 — последовательные порты и LPT1, LPT2 — параллельные, хотя интерфейс LPT давно устарел.

Осуществить ввод данных с помощью EDIT.

Скопировать файл в другой подкаталог.

9. Выдать на экран полное дерево каталогов, записать это дерево в файл.

Использовать команду TREE и перенаправление ввода-вывода > (вывод в файл) и < (считывание из файла), например

DIR > имя файла

запись результатов команды dir в файл,

добавление к файлу >>.

TYPE FILE1.TXT >> RESULT.TXT

добавит к файлу RESULT данные файла FILE1.

PROG < MYFILE.DAT

обеспечит ввод данных из файла в программу PROG.

Общий формат.

Команда [>|< |>>|<<] файл

Один знак > создает новый файл или переписывает старый, да знака >> создают новый файл если его нет, либо дописывают данные в существующий файл.

10. Удалить скопированные файлы и содержащий их подкаталог. Команда DEL — удалить файлы, команда RD или RMDIR — удаление каталога.

Для удаления каталога можно воспользоваться сначала командой DEL, удалив все файлы, а затем командой RD, удалив пустой каталог, для того чтобы удалить непустой каталог необходимо воспользоваться ключом команды rmdir /s.

Удаление одного или нескольких файлов.

DEL [/P] [/F] [/S] [/Q] [/A[[:]атрибуты]] имена

ERASE [/P] [/F] [/S] [/Q] [/A[[:]атрибуты]] имена

имена это Имена одного или нескольких файлов. Для удаления сразу нескольких файлов используются подстановочные знаки. Если указан каталог, из него будут удалены все файлы.

/P Запрос на подтверждение перед удалением каждого файла.

/F Принудительное удаление файлов, доступных только для чтения.

/S Удаление указанных файлов из всех подкаталогов.

/Q Отключение запроса на подтверждение при удалении файлов.

/A Отбор файлов для удаления по атрибутам.

атрибуты S Системные файлы R Доступные только для чтения

H Скрытые файлы A Файлы для архивирования

Префикс "-" имеет значение НЕ

Изменение команд DEL и ERASE при включении расширенной обработки команд:

Результаты вывода для ключа /S принимают обратный характер, то есть выводятся только имена удаленных файлов, а не файлов, которые не удалось найти.

Например, Del ***.*** - удаление файлов с расширением из текущего каталога.

Удаление каталога.

RMDIR [/S] [/Q] [диск:]путь

RD [/S] [/Q] [диск:]путь

/S Удаление дерева каталогов, т. е. не только указанного каталога, но и всех содержащихся в нем файлов и подкаталогов.

/Q Отключение запроса подтверждения при удалении дерева каталогов с помощью ключа /S.

11. Написать удобный BAT или CMD файл, который позволит вводить или добавлять анкетные данные в указанный файл (обеспечить вывод подсказок для ввода и меню для выхода). Это исполнимые файлы которые могут в себе содержать последовательность команд DOS, а также специальные конструкции присущие языкам программирования, что позволяет создавать некие системные программы выполняющие определенную последовательность действий, упрощая работу пользователя. Все ниже перечисленные команды можно подробно посмотреть используя команду help <имя команды>.

Основные команды BAT файлов

call Вызов одного пакетного файла из другого.

echo Вывод сообщений и переключение режима отображения команд на экране.

for Запуск указанной команды для каждого из файлов в наборе.

goto Передача управления в отмеченную строку пакетного файла.

if Оператор условного выполнения команд в пакетном файле.

pause Приостановка выполнения пакетного файла и вывод сообщения

rem Помещение комментариев в пакетные файлы и файл CONFIG.SYS.

shift Изменение содержимого (сдвиг) подставляемых параметров для пакетного файла.

set — установка значения переменной или ввод значения с клавиатуры.

Пример BAT файла.

```
Rem это комментарий в бат файле
Rem отключение режима вывода запуска команд
@echo OFF
Rem вывод строки на экран
echo Input information
Rem ввод в переменную val1 данных с клавиатуры
set /P val1= Input value :^>
Rem задание числовых данных
set /A val2= 4
Rem вывод информации о переменных
set val2
set val1
```

```
Rem условный оператор
IF "%val1%"=="1" (echo asdasdasdssa1)

Rem вывод случайного значения
ECHO inf1 %RANDOM%
Rem вывод переменной на экран
ECHO inf2 %val2%
ECHO inf3 "%val1%"
```

Rem вывод входного параметра бат файла 0-й параметр – название и путь к самому бат файлу, остальные параметры задаются через пробел при запуске бат файла

```
ECHO inf4 %0%
Rem ожидание нажатия любой клавиши
Pause
```

12. Написать файл для запуска редактора компилятора Borland Pascal (или Borland C) с заготовкой программы, которая содержит основные ключевые заголовки программы (и позволяет выводить – Hello world).

Например,

```
Program prog1;
Var
Begin
Writeln(“Hello world”)
End.
```

Или

```
#include <stdio.h>
int main()
{
printf(“Hello world\n”);
return 0;
}
```

Сделать задание в соответствии со своим вариантом. Написать cmd или bat файл:

1.Который позволяет осуществлять ввод чисел и записывает в файл сумму каждых двух введенных.

2.Который реализует вывод чисел Фибоначчи в файл до N включительно. N вводится с клавиатуры.

3.Который реализует вывод степеней двойки в файл до N включительно. N вводится.

4.Вывести разность и сумму двух массивов из N элементов.

5.Ввести два числа и минимальное записать в файл.

6.Ввести две координаты в двумерном пространстве и найти квадрат расстояния между ними.

7.Ввести два трехмерных вектора, найти их скалярное произведение и вывести в файл.

8.Написать вывод факториалов до N.

9.Ввести 9 чисел и вывести их в файл в виде матрицы 3 на 3.

10.Ввести число, вывести все его цифры в файл.

11.Ввести число, определить количество цифр в числе и вывести в файл.

12.Ввести два вектора одной длины, вывести в файл целые деления их друга на друга и остатки. Длину вектора ввести с клавиатуры.

13.По двум катетам треугольника найти квадрат гипотенузы и вывести в файл.

14.Вывести в файл квадрат с помощью звездочек, число звездочек для стороны квадрата ввести с клавиатуры.

15.Вывести в файл треугольник из звездочек, сначала одна звездочка, потом две и т.д., количество звездочек в самом низу задается с клавиатуры.

Для некоторых задач может быть полезен следующий пример:

@echo off

echo.>f.txt

echo.>>f.txt – перенос строки

set val2=a

echo %val2%>>f.txt

set val2=%val2%b – конкатенация строк.

echo %val2%>>f.txt перенаправление строки в файл.

set val2=%val2%cd конкатенация строк.

echo %val2%>>f.txt

pause

А так же следующий пример:

```
rem отключаем вывод на экран команд
@echo off
```

```
rem переходим на метку старт
goto start

rem процедура которую мы будем вызывать
:myproc
set val2=%1% World
exit /b

:start
rem вызов процедуры
call :myproc Hello
rem возвращение значение которое было задано в процедуре myproc
echo %val2%

rem вызов отдельного исполнимого файла с двумя параметрами
call f.cmd per1 per2
rem вывод переменной которую мы задали в f.cmd
echo %newf%
set /A A=11
rem выполняем операцию взятия остатка от деления на 3
set /a B=%A%%%3
echo %B%
rem выводим на экран возрастающие значения используя  цикл
FOR /L %%j IN (0,1,10) DO echo %%j
rem создаем импровизированный массив создавая переменные вида
rem per[0] per[1] per[2] и т.д.
FOR /L %%j IN (0,1,15) DO (set per[%%j]=%%j_ok)
rem выводим перменную на экран
echo %per[1]%
set /a myindex=15
rem выводим как бы элемент массива, просто у нас получается имя
rem в виде per[15]
call echo %%per[%myindex%]%%
rem делаем присвоение такой переменной "массива"
set /a myindex1=2
call set per[%myindex%]=some str
rem присвоение между элементами массива
call set per[%myindex%]=%%per[%myindex1%]%%
rem выводим на экран
call echo %%per[%myindex%]%%

rem Setlocal EnableDelayedExpansion
```

```
rem set /a strnv=0
rem for /F %%A in  (f1.cmd) do (
rem echo !strnv!
rem  echo %%A
rem set /a strnv=!strnv!+1
rem )
rem echo %strnv% >> 1.txt

rem приводится пример того как обращаться к таким "массивам"
rem используя переменную цикла
rem если не использовать !! и не установить expansion
rem то переменная не будет меняться при присовении
Setlocal EnableDelayedExpansion
set /a n=15
set /a ind=0
for /L %%j IN (1,1,%n%) do (
set /a val=%%j-1
call echo %%per[!ind!]%%
call set /a per[%%j]=%%per[!val!]%%
call echo %%per[!ind!]%%
set /a ind=!ind!+1
)
echo %strnv% >> 1.txt
```

Файл f.cmd:

```
@echo off
echo %0
echo %1
echo %2
set newf=12345abcde
```

Еще один пример где номер элемента массива отделяется точкой.

```
@echo off
goto start

:myproc
set val2=%1% World
exit /b
```

```
:start

call :myproc Hello
echo %val2%

call f.cmd per1 per2
echo %newf%
set /A A=11
set /a B=%A%%%3
echo Ostatok ot delenia
echo %B%

FOR /L %%j IN (0,1,10) DO echo %%j

FOR /L %%j IN (0,1,15) DO (set per.%%j=%%j_ok)

echo %per.1%
set /a myindex=15
call echo %%per.%myindex%%%
set /a myindex1=2
call set per.%myindex%=some str

call set per.%myindex%=%%per.%myindex1%%%
call echo %%per.%myindex%%%

Setlocal EnableDelayedExpansion
set /a strnv=0
for /F %%A in  (f1.cmd) do (
 echo !strnv!
 echo %%A
 set /a strnv=!strnv!+1
)
```

7 Изучение Форм и визуальных элементов управления в OpenOffice или LibreOffice.

Для дополнительной помощи при выполнении лабораторной можно обращаться по адресу: http://help.libreoffice.org/Basic/Basic_Help/ru.

7.1 Изучение msgbox

Для создания простого диалогового окна с сообщением и несколькими кнопками, можно воспользоваться функцией msgbox, которая имеет следующие парметры:

MsgBox Текст As String [,Тип As Integer [,Заголовок As String]]

или

MsgBox (Текст As String [,Тип As Integer [,Заголовок As String]])

Квадратные скобочки указывают на необязательные параметры.

Текст. Строковое выражение, отображаемое как сообщение в диалоговом окне. Переносы строк можно вставить с помощью Chr$(13).

Заголовок. Строковое выражение, отображаемое в заголовке диалогового окна. Если параметр пропущен, в строке заголовка отображается имя соответствующего приложения.

Тип. Выражение из целых чисел, указывающее тип диалогового окна, а также число и тип отображаемых кнопок и тип значков. **Тип** представляет комбинацию битовых масок, то есть комбинация элементов может определяться добавлением соответствующих значений:

0 . Показать только кнопку "ОК".

1 . Показать кнопки "ОК" и "Отмена".

2 : Показать кнопки "Прервать", "Повторить" и "Пропустить".

3 . Показать кнопки "Да", "Нет" и "Отмена".

4 . Показать кнопки "Да" и "Нет".

5 . Показать кнопки "Повторить" и "Отмена".

16 . Добавить в диалоговое окно значок "Стоп", будет отображаться иконка.

32 . Добавить в диалоговое окно значок "Вопрос".

48 . Добавить в диалоговое окно значок "Восклицательный знак".

64 . Добавить в диалоговое окно значок "Сведения".

128 . Первая кнопка в диалоговом окне как кнопка по умолчанию.

256 . Вторая кнопка в диалоговом окне как кнопка по умолчанию.

512 . Третья кнопка в диалоговом окне как кнопка по умолчанию.

После 16 коды представляют собой битовые маски, например, 5+16 будет означать наличие кнопок «Повторить», «Отмена» и значка-иконки «Стоп», 1+64 — будет наличие кнопки «OK» и значка-иконки «Сведения».

Пример

Для создание макроса воспользуйтесь Сервис-Макросы-Управление Макросами-LibreOffice Basic, выберете модуль, стандартный или текущего файла, (лучше текущего файла, чтобы была привязка к конкретному файлу), выберете процедуру Main или создайте новую процедуру в окне редактора — например записав имя процедуры

```
Sub Main1
msgbox  "Наша строка сообщения", 1+16, "Наше название окна"
End Sub
```

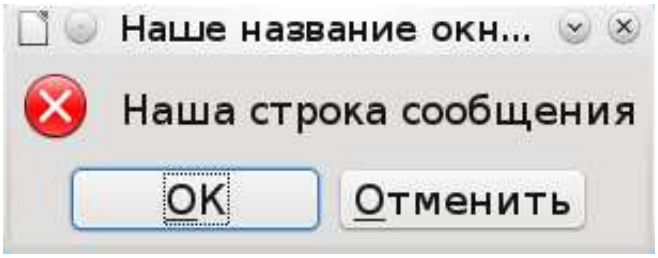


Рисунок 7.1 - пример Диалогового окна для обработки ошибок

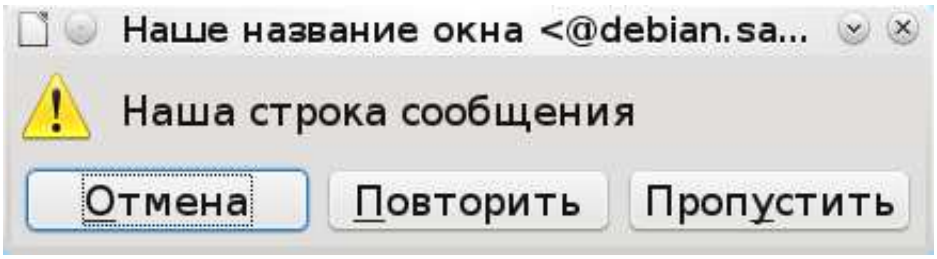


Рисунок 7.2 - Пример окна с восклицательным знаком

Задание. Изучите остальные виды окон и сообщений, комбинируя знаки и кнопки. Например, как будет вести себя окно при выборе типа 64 и других видов кнопок.

Функция msgbox может возвращать значение показывающее какая из кнопок была нажата.

```
Sub Macro3
dim val as integer
val = msgbox ("Наша строка сообщения", 3+16, "Наше название окна")
msgbox val
end Sub
```

Например, при нажатии кнопки ОК — будет возвращено значение 1, то есть переменной val будет присвоено значение 1, кнопка отменить значение 2, кнопка Отмена — 3, кнопка Повторить — 4, кнопка Пропустить — 5, Кнопка Да — 6, Кнопка Нет — 7.

Задание. Напишите макрос, который будет на вопрос перезапустить Макрос запускать сам макрос, при ответе Нет, выходить из макроса. Использовать IF и рекурсивный вызов. Например, Macro3(). Сделать небольшой Wizard, который позволяет устанавливать какие-то свойства текста, путем последовательного вызова msgbox. Например, Wizard — Увеличить текст на пункт, поднять яркость текста, при достижении яркости 255, сбрасывать на ноль, можно использовать Ок, Повторить, Пропустить.

7.2 Создание Диалогового окна со строкой ввода.

Инструкция **InputBox** является удобным методом ввода текста через диалоговое окно. Подтвердите ввод, нажав кнопку "ОК" или клавишу ВВОД. Результат передается как возвращаемое значение функции. Если это диалоговое окно закрыть с помощью кнопки "Отмена", **InputBox** возвращает строку нулевой длины ("").

InputBox (Сообщение As String[, Заголовок As String[, По_умолчанию As String[, позиция_X As Integer, позиция_Y As Integer]]]])

Сообщение. Строковое выражение, отображаемое как сообщение в диалоговом окне.

Заголовок. Строковое выражение, отображаемое в заголовке диалогового окна.

По_умолчанию. Строковое выражение, по умолчанию отображаемое в текстовом поле, если нет других выводимых данных.

позиция_X. Выражение из целых чисел, которое указывает горизонтальную позицию диалогового окна. Эта позиция является абсолютной координатой и не имеет отношения к окну приложения Office.

позиция_Y. Выражение из целых чисел, которое указывает вертикальную позицию диалогового окна. Эта позиция является абсолютной координатой и не имеет отношения к окну приложения Office.

Если значения **позиция_X** и **позиция_Y** не указаны, диалоговое окно размещается в середине экрана.

```
dim val as String
val = inputbox ("Наша строка сообщения", "Наше название окна", "Значение для ввода по умолчанию")
msgbox val
```

Задание. Используя данную функцию заполнить последовательно ячейки в таблице Calc по столбцу или по строке. Первый inputbox вводит число заполняемых ячеек, предусмотрите, чтобы при нажатии кнопка отмена, можно было прервать цикл ввода. Использовать цикл While.

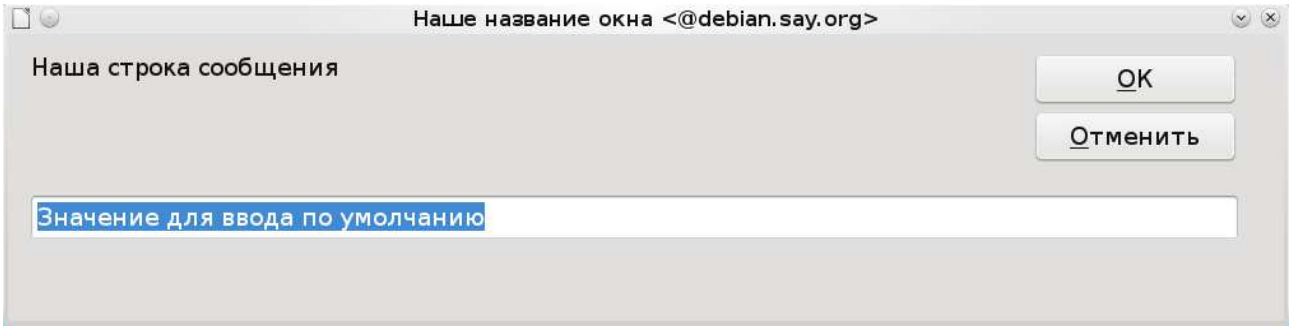


Рисунок 7.3 - Ввод текста с помощью диалога

Создание собственного диалогового окна.

Отличие диалогового окна от Формы, в том, что пока Диалоговое окно активно и работа с ним не завершена иные функции и окна LibreOffice не доступны.

7.3 Создание диалога

Создайте новый диалог, выполнив Сервис - Макросы - Управление диалогами...Можно создать новый диалог нажав на кнопку «Новый диалог» или выбрать Dialog и нажать Правка.

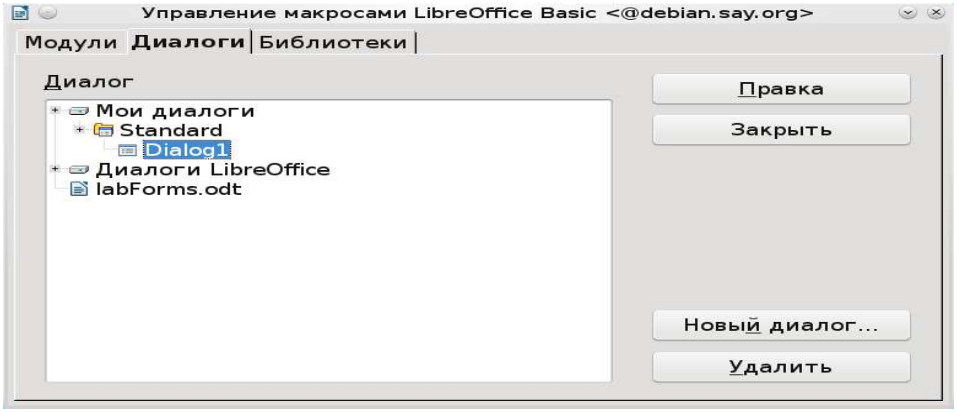


Рисунок 7.4 - Создание Диалога

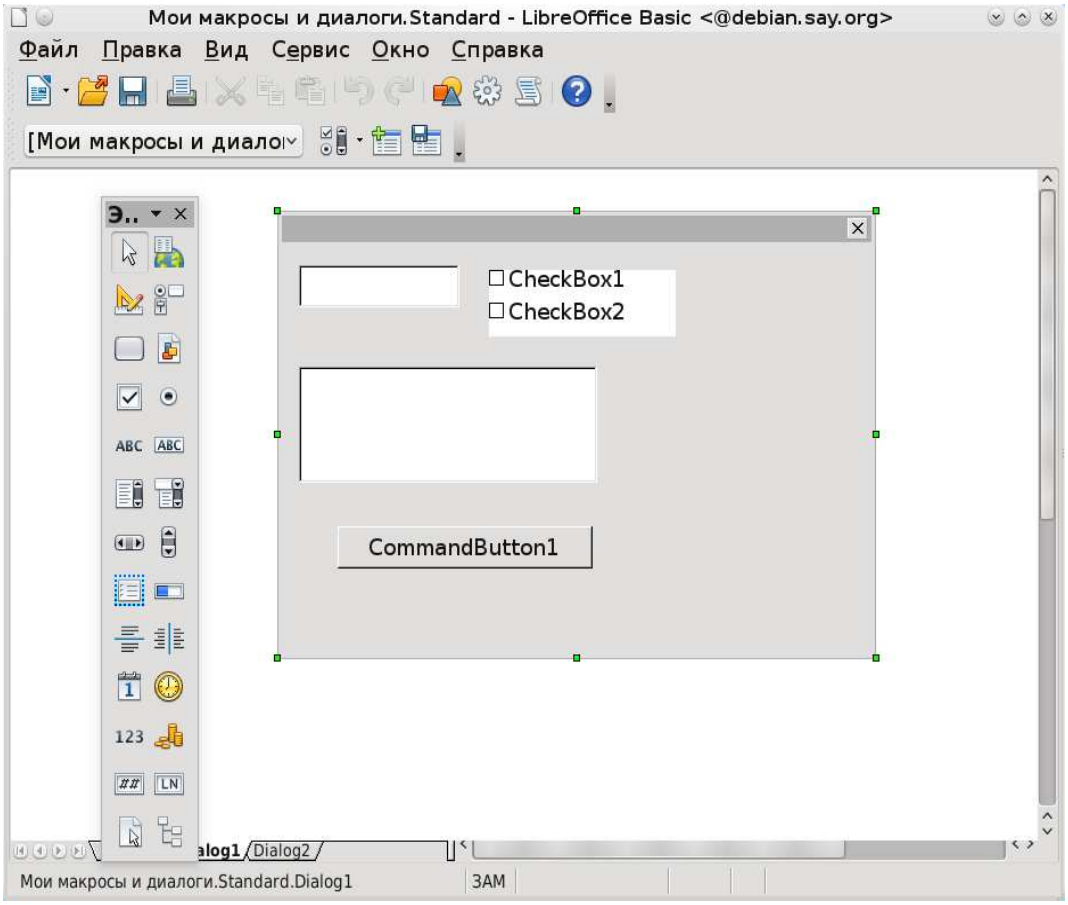


Рисунок 7.5 - Редактирование окна диалога, установка элементов управления

На рисунке представлено окно диалога и некоторые элементы управления уже перенесенные с панели элементов управления на форму диалога. Для того, чтобы это сделать необходимо вывести панель Элементов, можно воспользоваться Меню Вид-Панели Инструментов — Элементы Управления, затем щелкнув правой кнопкой мыши на нужном элементе перейти на форму Диалога и растянуть элемент на форме.

Для того, чтобы запустить созданный dialog на выполнение можно воспользоваться последовательностью операций, записав их в макрос.

```
basicLibraries.loadLibrary("Tools")
Dlg =  loadDialog("Standard", "Dialog1")
Dlg.execute()
```

Глобальная функция загрузки окон доступна только из модуля LibreOffice «Мои макросы», если созданный диалог прикреплен к конкретному файлу и соответственно макрос, вызывающий его, лучше воспользоваться вторым методом.

Данный метод представляет собой загрузку библиотеки диалогов данного файла и затем создания Диалога в соответствии с типом диалога и последующего запуска на выполнение.

```
DialogLibraries.LoadLibrary("Standard")
```

```
Dlg = CreateUnoDialog(DialogLibraries.Standard.Dialog1a)
Dlg.Execute()
Dlg.Dispose()
```

CreateUnoDialog создает объект по имени Dlg, который ссылается на связанный диалог. Прежде, чем Вы можете создать диалог, Вы должны гарантировать, что библиотека, которую он использует (в этом примере, библиотека Standard) загружена. В противном случае метод LoadLibrary выполняет эту задачу.

Как только объект диалог Dlg проинициализировался, Вы можете использовать метод

Execute для отображения диалога. Диалоги такие, как этот описываются как модальные, потому что они не разрешают никакого другого действия программы, пока они не закрыты.

В то время как этот диалог открыт, программа остается в запросе Execute.

Метод dispose в конце кода освобождает ресурсы, используемые диалогом однажды при завершении программы.

Для закрытия диалога можно воспользоваться функцией Dlg.EndExecute().

7.4 Реализация диалога с кнопкой

Например, попробуем создать событие, которое происходит при нажатии кнопки на Диалоге и закрывает Диалог, для этого откроем форму конструктора нашего диалога, выберем кнопку, щелкнем правой кнопкой мыши и выберем Свойства, появится окно, представленное ниже.

Рисунок 7.6 - Свойства объекта кнопка

Это все свойства объекта управления Кнопка. Можно изменить какое-либо свойство, например, визуально видимый текст на слово Выход, при этом имя самого объекта CommandButton1.

Для того, чтобы при нажатии на кнопку выполнялись какие-то действия необходимо связать с процедурой обработкой события, какую-то вашу процедуру. Для этого необходимо создать макрос в окне редактирования макросов, например,

```
Sub Macro5()
Dlg.EndExecute()
End Sub
```

Затем в окне свойств, выбрать вкладку События.

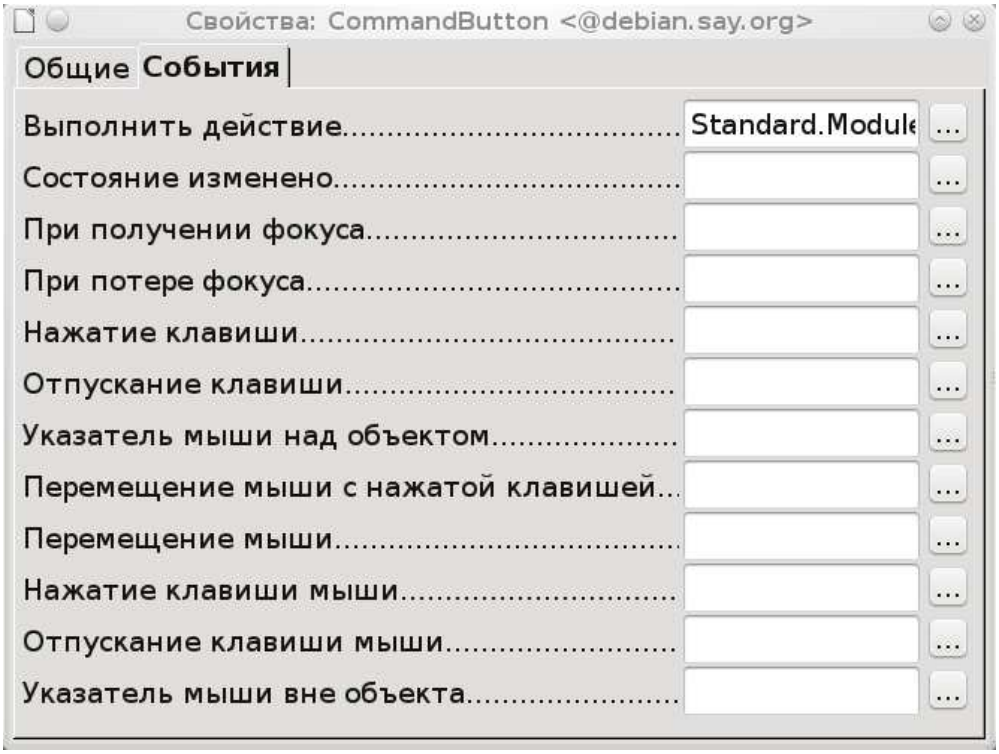


Рисунок 7.7 - События объекта кнопка

Выбрать нужное событие, в данном случае Выполнить действие. Затем связать назначенное действие с Макросом, макрос выбрать из списка ваши макросов.

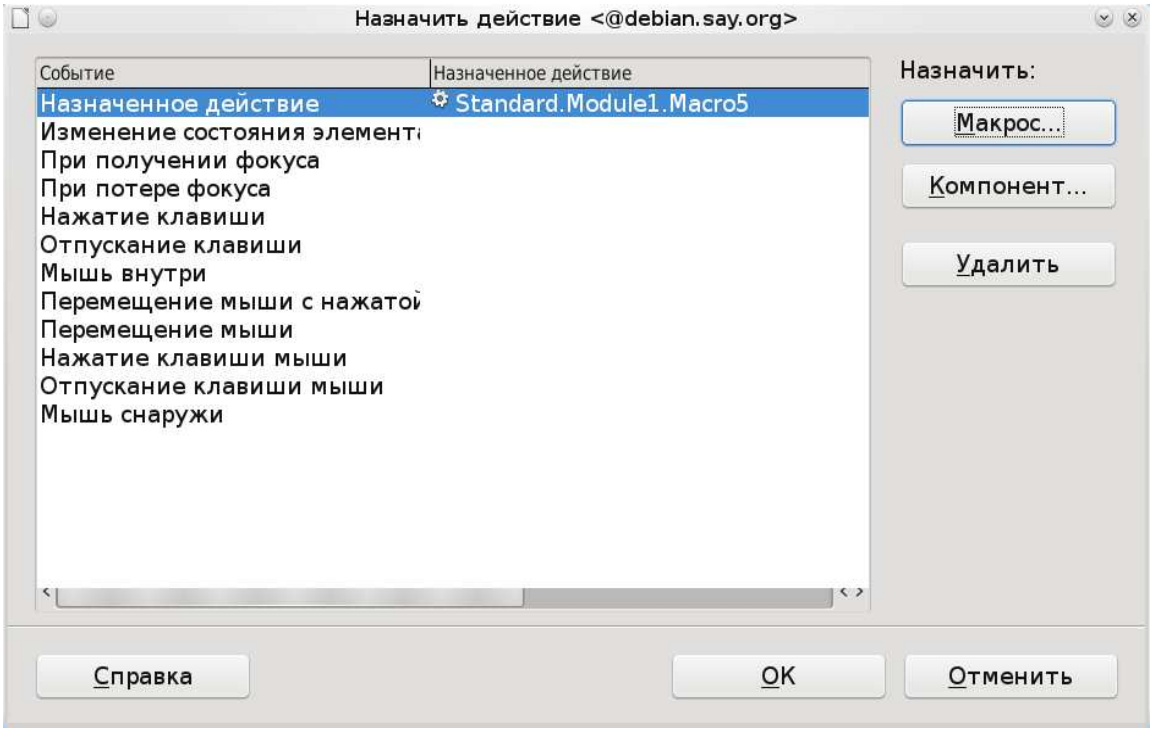


Рисунок 7.8 - Назначение события для кнопки

Общий вид программы будет выглядеть следующим образом:

```
Dim Dlg As Object
Sub MAcro4
basicLibraries.loadLibrary("Tools")
Dlg =  loadDialog("Standard", "Dialog1")
Dlg.execute()
end sub
Sub Macro5
Dlg.EndExecute()
end sub
```

Переменная Dlg вынесена за пределы обеих процедур и является глобальной, чтобы обе процедуры могли ею пользоваться, после того как сработает макрос 4, переменная dlg будет ссылаться на созданный диалог, и потому при срабатывании события и вызове макроса 5, данный диалог будет закрыт.

Для получения доступа к объекту можно воспользоваться функцией getControl, в качестве аргумента указывая имя объекта.

```
Dim Ctl As Object
Ctl = Dlg.getControl("MyButton")
```

Ctl.Label = "Новая надпись"

Если не хочется заводить еще одну переменную можно задавать свойства объекта таким образом:

Dlg.getControl("TextField1").Text = "строка по умолчанию ".

TextField1 — объект для ввода строки текста.

Весь макрос:

```
Sub Macro4
basicLibraries.loadLibrary("Tools")
Dlg =  loadDialog("Standard", "Dialog1")
Dlg.getControl("TextField1").Text = "Hello World "
Dlg.execute()
end sub
```

7.5 Модель объекта

Модель объектов управления UNO представляет собой реализацию паттерна проектирования Модель-Вид-Контроллер, когда реализация визуального представления объекта отделена от данных и управления этим объектом, это позволяет при проектировании легко изменять визуальный вид объекта, не затрагивая его внутреннее модельное представление или данные, которые связаны с данным объектом (чтобы бизнес логика приложения минимально зависела от визуализации и взаимодействия и наоборот). Обычно используется для создания приложений, где присутствует интерфейс взаимодействия пользователя с некой системой. Например, можно изменить принцип реакции на изменение данных в системе или способы изменения этих данных, используя совершенно другие виды интерфейса, при этом нет необходимости как-либо затрагиваться модельный уровень.

Разделение между видимыми элементами программы (Вид) и данными или документами позади них (Модель) происходит во многих местах в OpenOffice.org API. В дополнение к методам и свойствам элементов управления, и диалог и объекты элементов управления имеют подчиненный объект Model. Этот объект позволяет Вам получить непосредственный доступ к содержимому диалога или элемента управления.

В диалогах, различие между данными и описанием не всегда столь же ясно как в других областях LibreOffice API. Элементы API доступны и через Вид и через Модель.

Свойство Model обеспечивает программно-управляемый доступ к модели диалога и объектам элементов управления.

```
Dim cmdNext As Object
```

```
cmdNext = Dlg.getControl("cmdNext")
cmdNext.Model.Enabled = False
```

Этот пример отключает кнопку cmdNext в диалоге Dlg при помощи объекта модели cmdNext.

Имя и заголовок

Каждый элемент управления имеет свое собственное имя, которое может быть запрошено с использованием следующего свойства модели:

• Model.Name (String) – имя элемента управления.

Вы можете определить заголовок, который появляется в заголовке диалога через следующее

свойство модели:

• Model.Title (String) – заголовок диалога (применяется только к диалогам).

Положение и Размер

Вы можете запросить размер и положение элемента управления, используя следующие свойства объекта модель:

• Model.Height (long) – высота элемента управления (в единицах ma);

• Model.Width (long) – ширина элемента управления (в единицах ma);

• Model.PositionX (long) – координата X элемента управления, измеренная от левого внутреннего края диалога (в единицах ma);

• Model.PositionY (long) – координата Y элемента управления, измеренная от верхнего внутреннего края диалога (в единицах ma).

Чтобы гарантировать независимость от платформы для внешнего вида диалогов, LibreOffice использует внутренную единицу Map AppFont (ma) для определения положения и размера в пределах диалогов. Единица ma определена как одна восьмая средней высоты символа системного шрифта, определенного операционной системой и одной четверти его ширины. При использовании единицы ma, OpenOffice.org гарантирует, что диалог выглядит одинаково на различных системах при различных параметрах настройки системы.

Если Вы хотите изменить размер или положение элементов управления во время выполнения, определите полный размер диалога и регулируйте значения для элементов управления в соответствующем отношении частей.

Пример использование Model: сделать объект недоступным.

```
Dlg.getControl("TextField1").Model.Enabled = false
```

7.6 Изучение Форм и элементов управления

Формы в отличие от диалога создаются непосредственно в документах LibreOffice, то есть на листах Calc или документах Writer. При этом пользователю для управления доступны все возможности открытого документа, тогда как в случает диалога пока он не закроется, для пользователя эти возможности не доступны.

Изучим отдельно каждый из элементов управления на примере работы с LibreOffice Calc, для этого откроем рабочий лист Calc и выберем Вид-Панели Инструментов-Элементы управления. Значок Линейка на элементах управления позволяет переходить из режима конструктора, когда можно задавать свойства объекта и в режим исполнения, когда объект реагирует на внешние события. Щелкнем на объекте Button(Кнопка) и вытащим его на лист calc, для этого нужно щелкнуть на поле листа и начать растягивание объекта. В режиме конструктора выберем правой кнопкой мыши во всплывающем меню Элемент управления, назовем как-нибудь кнопку и зададим нужные нам свойства и перейдем к вкладке события. Повторим действия как при создании события кнопки диалога. Создадим макрос и назначим его в качестве события.

Теперь необходимо во вновь созданный макрос записать следующую последовательность действий для изменения свойств листов Calc.

Ниже написанный макрос может некорректно работать для различных версий LibreOffice, потому второй способ предлагает сначала создать стиль, а затем использовать данный стиль для изменения свойств диапазона ячеек.

```
Sub Macro6
Dim Doc,Sheet,Range as Object
Dim n as integer
'получаем ссылку на текущий открытый документ
Doc = StarDesktop.CurrentComponent
'получаем число листов в документе
n = Doc.Sheets.Count
'объект для задания свойств линий границ
Dim  aLineBorder as new com.sun.star.table.BorderLine
'объект для задания свойств границ
Dim  Border as new com.sun.star.table.TableBorder
'цвет линии
aLineBorder.Color = 0
```

```
'внутренняя тощина линии
aLineBorder.InnerLineWidth = 0
'внешняя толщина линии
aLineBorder.OuterLineWidth = 50
aLineBorder.LineDistance = 0
'установка что все линии рамок видимы
Border.IsTopLineValid = true
Border.IsBottomLineValid = true
Border.IsLeftLineValid = true
Border.IsRightLineValid = true
Border.IsHorizontalLineValid = true
Border.IsVerticalLineValid = true
'задание свойств верхней, нижней, правой и левой линии, внутренней вертикальной и горизонтальной
Border.TopLine = aLineBorder
Border.BottomLine = aLineBorder
Border.LeftLine = aLineBorder
Border.RightLine = aLineBorder
Border.HorizontalLine = aLineBorder
Border.VerticalLine = aLineBorder
'цикл по листам
for i=0 to n-1
'получаем лист под номером i
Sheet = Doc.Sheets(i)
'берем диапазон ячеек
Range = Sheet.getCellRangeByName("A1:Z100")
'задаем свойство прозрачности фона
Range.IsCellBackgroundTransparent = true
'задаем цвет ячеек листа
Range.CellBackColor = RGB(i*50,(i+2)*50,i*20)
'задаем свойства границ
Range.TableBorder = Border
'Выравнивание по горизонтали влево
Range.HoriJustify = com.sun.star.table.CellHoriJustify.LEFT
'Выравнивание по вертикали по верху
```

```
Range.VertJustify = com.sun.star.table.CellVertJustify.TOP
'расположение текста по буквам сверху вниз, символы горизонтальные
Range.Orientation = com.sun.star.table.CellOrientation.STACKED
next i
End Sub
```

Данный макрос просто присваивает ячейкам листа данный стиль. Предварительно стиль можно задать Формат-Стиль.

```
Sub Macro6
Dim Doc,Sheet,Range as Object
Dim n as integer
Doc = StarDesktop.CurrentComponent
n = Doc.Sheets.Count
Dim  aLineBorder as new com.sun.star.table.BorderLine
Dim  Border as new com.sun.star.table.TableBorder
for i=0 to n-1
Sheet = Doc.Sheets(i)
Range = Sheet.getCellRangeByName("A1:Z100")
'Установка стиля листа New
Range.CellStyle = "New"
next i
End Sub
```

Попробуем на кнопке разместить рисунок, для этого откроем свойства объекта и просмотрим все его свойства, найдя Изображение. Выберем графический файл щелкнув на кнопке три точки. Предварительно файл с изображением можно создать в любом графическом редакторе.

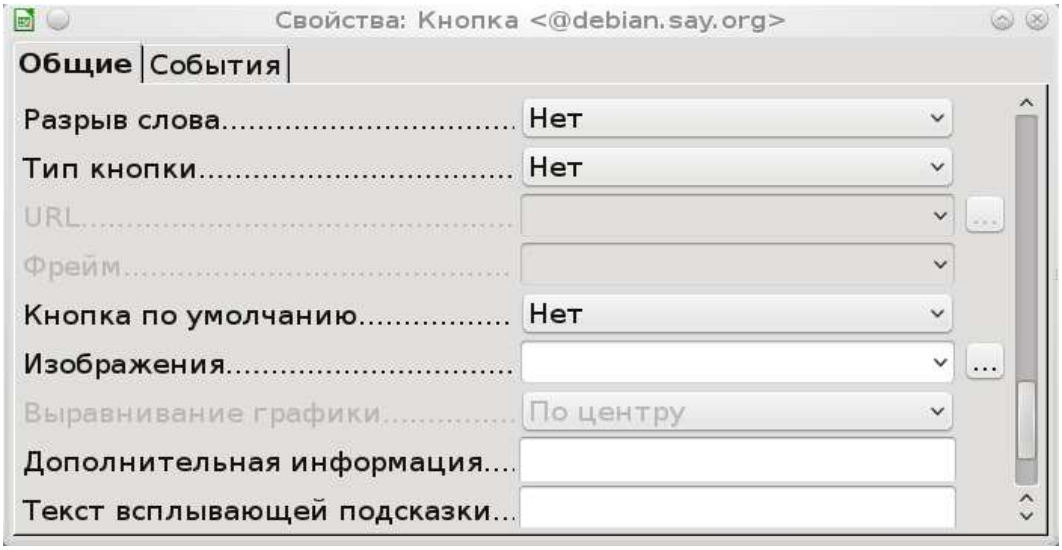


Рисунок 7.9 - Установка изображения на кнопке

7.7 Изучение флажков.

С помощью элемента управления перенесите на первый лист шесть флажков и расположите их друг за другом. Щелкнув правой кнопкой мыши над элементом флажок посмотрите его свойства, посмотрите как называется объект, если он не называется Флажок 1, Флажок 2 и т.д., назовите его так или учтите это при выполнении дальнейшего задания. Далее создайте Макрос и назовите каким-нибудь образом, в данном случае процедура макроса должна иметь стандартный вид процедуры обработчика события с наличием входного аргумента Event.

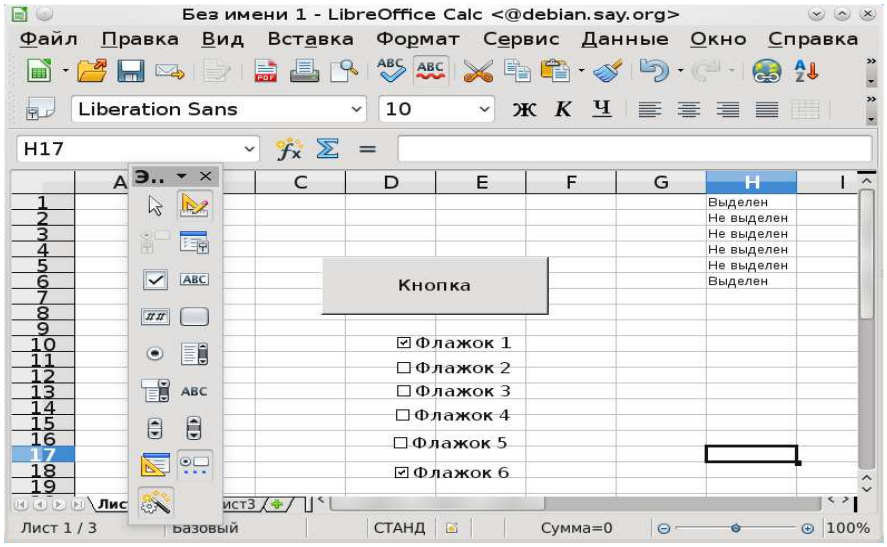


Рисунок 7.10 - Флажки и работа с Calc, обработка событий

В OOo Basic Вы можете использовать параметры объекта, чтобы предоставить дополнительную информацию о событии для процедуры, например:

Sub ProcessEvent(Event As Object)

End Sub

Подробность, с которой объект Event структурирован и его свойства, зависит от типа события, которое инициирует вызов процедуры. Независимо от типа события, все объекты обеспечивают доступ к соответствующему элементу управления и его модели. Элемент управления может быть достигнут с использованием

Event.Source

а его используемая модель

Event.Source.Model

Вы можете использовать эти свойства для инициирования события в пределах обработчика события.

Запишите код макроса. Затем свяжите данный макрос с реакцией на событие с каждым объектом флажок.

```
Sub Macro8(Event as Object)
Dim Doc As Object
Dim Sheet As Object
Dim num as integer
Doc = ThisComponent
'Показать имя объекта вызвавшего событие
msgbox Event.Source.Model.Name
'определяем какой именно объект создал событие и нумеруем
if Event.Source.Model.Name = "Флажок 1" then
num  = 1
end if
if Event.Source.Model.Name = "Флажок 2" then
num  = 2
end if
if Event.Source.Model.Name = "Флажок 3" then
num  = 3
end if
if Event.Source.Model.Name = "Флажок 4" then
num  = 4
end if
if Event.Source.Model.Name = "Флажок 5" then
num  = 5
```

```
end if
if Event.Source.Model.Name = "Флажок 6" then
num  = 6
end if
'Выбираем первый лист
Sheet = Doc.Sheets(0)
'Выбираем ячейку по ее координатам, первое число — столбец, второе номер строки
Cell = Sheet.getCellByPosition(7, num-1)
'если состояние флажка выделен, то указываем это в ячейке, в ином случае не выделен
if Event.Source.State = 1 then
Cell.String = "Выделен"
else
Cell.String = "Не выделен"
end if

End sub
```

Флажки предоставляют следующие свойства:

• State (Short) – состояние флажка (0: нет, 1: да, 2: промежуточное состояние);

• Label (String) – надпись для элемента управления;

• enableTriState (Boolean) – в дополнение к активированному и дезактивированному состояниям, Вы можете также использовать промежуточное состояние.

Модель объекта флажок обеспечивает следующие свойства:

• Model.FontDescriptor (struct) – структура, которая определяет детали шрифта, который используется (в соответствии со структурой com.sun.star.awt.FontDescriptor);

• Model.Label (String) – надпись, которая отображается на элементе управления;

• Model.Printable (Boolean) – элемент управления может быть напечатан;

• Model.State (Short) – состояние флажка (0: нет, 1: да, 2: промежуточное состояние);

• Model.Tabstop (Boolean) – элемент управления может быть достигнут при помощи клавиши Tab;

• Model.TextColor (Long) – цвет текста элемента управления;

• Model.HelpText (String) – текст всплывающей подсказки, которая появляется, когда Вы перемещаете курсор мыши над элементом управления;

• Model.HelpURL (String) – URL страницы онлайн справки для соответствующего элемента управления.

7.8 Изучение Переключателей.

Эти кнопки используются в группах и позволяют Вам выбирать один из нескольких вариантов. Когда Вы выбираете переключатель, все другие переключатели в группе дезактивируются. Это гарантирует, что в любой момент времени установлен только один переключатель.

Элемент управления переключатель предоставляет два свойства:

• State (Boolean) – состояние переключателя;

• Label (String) – надпись, которая отображается рядом с переключателем.

Вы можете также использовать следующие свойства из модели переключателей:

• Model.FontDescriptor (struct) – структура, которая определяет детали шрифта, который используется (в соответствии со структурой com.sun.star.awt.FontDescriptor);

• Model.Label (String) – надпись, которая отображается на элементе управления;

• Model.Printable (Boolean) – элемент управления может быть напечатан;

• Model.State (Short) – если это свойство равно 1, переключатель активирован, иначе он дезактивирован;

• Model.TextColor (Long) – цвет текста элемента управления;

• Model.HelpText (String) – текст всплывающей подсказки, которая появляется, когда Вы перемещаете курсор мыши над элементом управления;

• Model.HelpURL (String) – URL страницы онлайн справки для соответствующего элемента управления.

Для того, чтобы объединить переключатели в одну группу, необходимо выбрать каждый переключатель и в их свойстве Имя Группы, указать одно и то же название группы для всех переключателей.

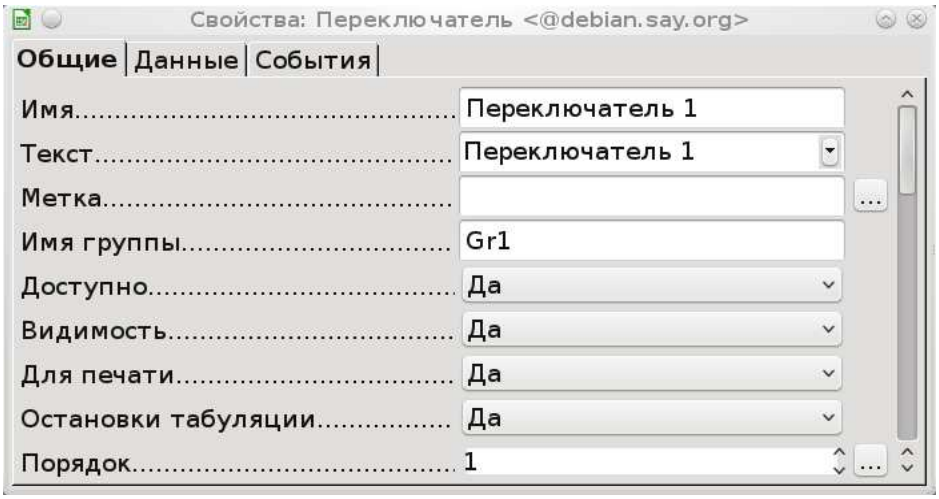


Рисунок 7.11 - Установка свойств переключателя

Добавьте на второй лист три переключателя и установите для них одну и ту же группу.

Добавьте в данных связанную с переключателем ячейку. В этом случае значение в поле Значение индекса (вкл/выкл) будет записано в указанную ячейку в зависимости от того в каком положении переключатель.

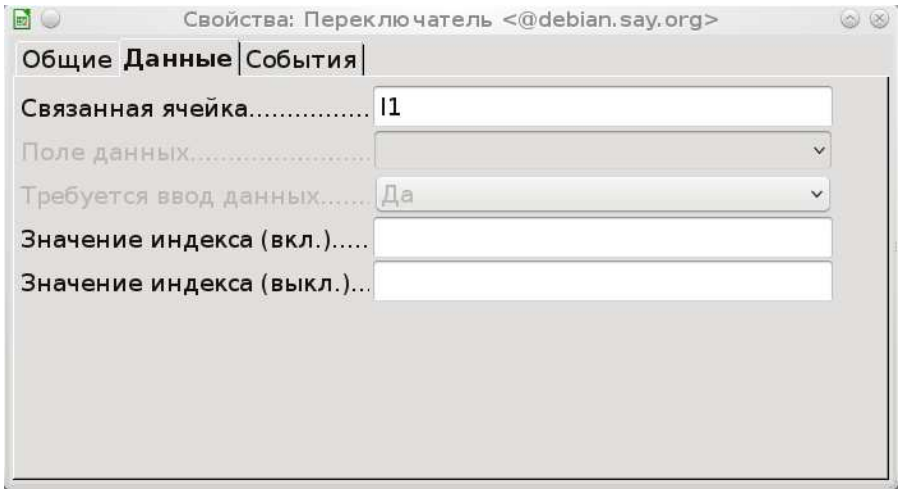


Рисунок 7.12 - Связанная с переключателем ячейка

Затем создайте макрос. И свяжите его с событием Выполнить действие.

```
Sub Macro9(Event as Object)
dim num as integer
Dim Shet, Doc,Cell as Object
'Проверка какой из переключателей сработал
if Event.Source.Model.Name = "Переключатель 1" then
num = 1
end if
if Event.Source.Model.Name = "Переключатель 2" then
num = 2
end if
if Event.Source.Model.Name = "Переключатель 3" then
num = 3
end if
'Обращение ко второму листу
Doc = ThisComponent
Sheet = Doc.Sheets(1)
'Ячейка первой строки восьмого столбца H
```

```
Cell = Sheet.getCellByPosition(7, 0)
'Установка значения ячейки номером сработавшего переключателя
Cell.Value = num
'Установка цвета символов ячейки и ее высоты в зависимости от переключателя
select case num
case 1
'цвет
Cell.CharColor = RGB(255, 0, 0)
'высота
Cell.CharHeight = 15
case 2
Cell.CharColor = RGB(0, 255, 0)
Cell.CharHeight = 10
case 3
Cell.CharColor = RGB(0, 0, 255)
Cell.CharHeight = 20
end select
end sub
```

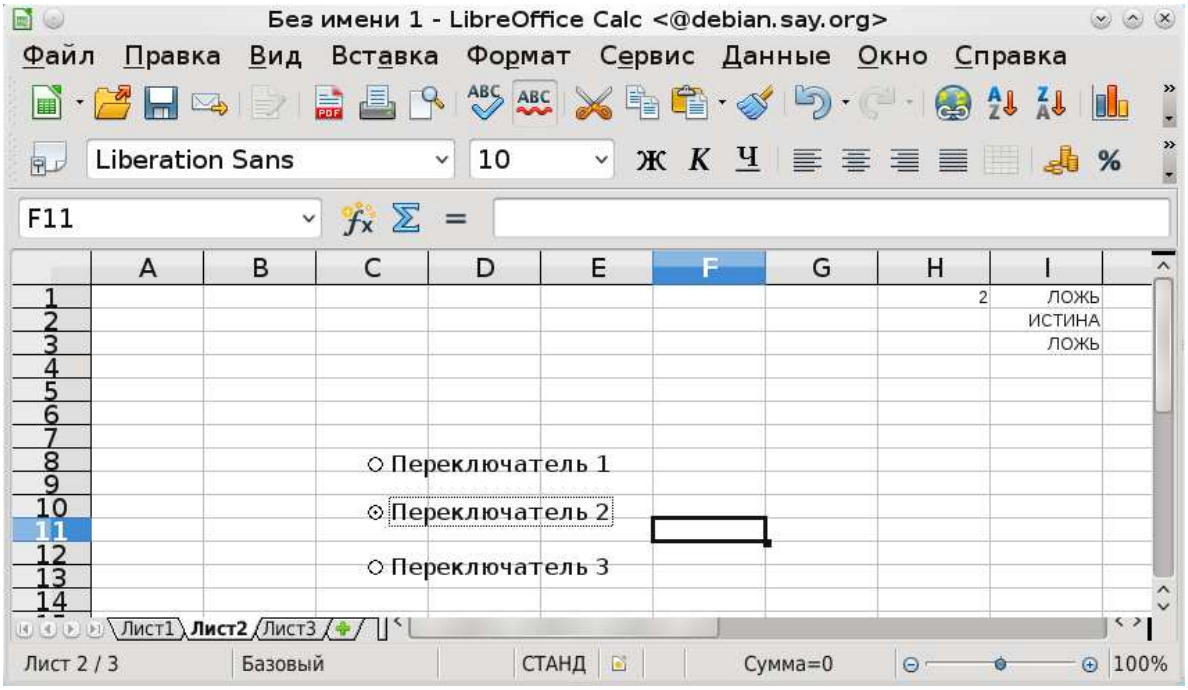


Рисунок 7.13 - Результат работы макроса

7.9 Текстовые поля

Текстовые поля позволяют пользователям вводить числа и текст. Текстовое поле может содержать одну или более строк и может редактироваться или быть заблокированным для пользовательского ввода. Текстовые поля могут также использоваться как поля ввода специальных денежных и числовых данных, а также как поля экрана для специальных задач. Поскольку эти элементы управления основаны на UNO сервисе UnoControlEdit, их программно-управляемая обработка подобна.

Текстовые поля предоставляют следующие свойства:

• Text (String) – текущий текст;

• SelectedText (String) – выделенный в настоящее время текст;

• Selection (Struct) – подробности выделения только для чтения (структура в соответствии с com.sun.star.awt.Selection, со свойствами Min и Max, определяющими начало и конец текущего выделения);

• MaxTextLen (short) – максимальное количество символов, которые Вы можете ввести в поле;

• Editable (Boolean) – True активизирует возможность ввода текста, False блокирует возможность ввода (свойство нельзя вызвать непосредственно, а только через IsEditable);

• IsEditable (Boolean) – содержимое элемента управления может быть изменено, только для чтения.

Текстовые поля предоставляют следующие методы:

• insertText (Sel, Text) – вставляет текст, заданный строковой переменной Text, в положение, определяемое структурой Sel;

• getSelectedText – возвращает строку, содержащую выделенный в настоящее время текст;

• setSelection (Selection) – устанавливает пользовательское выделение в соответствии с информацией, содержащейся в структуре Selection;

• setEditable (Editable) – делает текст редактируемым для пользователя или только для чтения в зависимости от логической переменной Editable.

Кроме того, следующие свойства предоставляются через связанный объект модели:

• Model.Align (short) – выравнивание текста (0: выравнивание по левому краю, 1: выравнивание по центру, 2: выравнивание по правому краю);

• Model.BackgroundColor (long) – цвет фона элемента управления;

• Model.Border (short) – тип границы (0: нет границы, 1: трехмерная граница, 2: простая граница);

• Model.EchoChar (short) – код эхо-символа для полей ввода пароля;

• Model.FontDescriptor (struct) – структура с подробностями используемого шрифта (в соответствии со структурой com.sun.star.awt.FontDescriptor);

• Model.HardLineBreaks (Boolean) – автоматические разрывы строки постоянно вставляются в текст элемента управления;

Элементы управления диалогов подробно

• Model.HScroll (Boolean) – текст имеет горизонтальную полосу прокрутки;

• Model.MaxTextLen (Short) – максимальная длина текста, где 0 соответствует отсутствию ограничения длины;

• Model.MultiLine (Boolean) – разрешает ввод в объеме нескольких строк;

• Model.Printable (Boolean) – элемент управления может быть напечатан;

• Model.ReadOnly (Boolean) – содержимое элемента управления только для чтения;

• Model.Tabstop (Boolean) – элемент управления может быть достигнут при помощи клавиши Tab;

• Model.Text (String) – текст связанный с элементом управления;

• Model.TextColor (Long) – цвет текста элемента управления;

• Model.VScroll (Boolean) – текст имеет вертикальную полосу прокрутки;

• Model.HelpText (String) – текст всплывающей подсказки, которая появляется, когда Вы перемещаете курсор мыши над элементом управления;

• Model.HelpURL (String) – URL страницы онлайн справки для соответствующего элемента управления.

7.10 Список

Элемент управления список (сервис com.sun.star.awt.UnoControlListBox) отображает список элементов, из которых пользователь может выбрать один или несколько. Если число элементов превышает то количество, которое может быть отображено в поле списка, в элементе управления автоматически появляются полосы прокрутки. Если свойство Dropdown установлено в True, список элементов отображается в раскрывающемся списке. В этом случае, максимальное количество строк в раскрывающемся списке определяется свойством LineCount. Фактическим списком элементов управляет свойство StringItemList. Всеми выбранными свойство пунктами управляет свойство SelectedItems.

Если MultiSelection установлено в True, может быть выбран более чем один элемент.

Списки поддерживают следующие свойства:

• ItemCount (Short) – число элементов, только для чтения;

• SelectedItem (String) – текст выделенного элемента, только для чтения;

• SelectedItems (Array Of Strings) – массив данных с выделенными записями (для списков, которые поддерживают множественный выбор), только для чтения;

• SelectItemPos (Short) – номер элемента, выделенного в настоящее время, только для чтения;

• SelectItemsPos (Array of Short) – массив данных с номерами выделенных записей (для списков, которые поддерживают множественный выбор), только для чтения;

• MultipleMode (Boolean) – True активизирует возможность для множественного выбора записей, False блокирует множественный выбор (свойство нельзя вызвать непосредственно, но только через IsMultipleMode);

• IsMultipleMode (Boolean) – разрешает множественный выбор в пределах списка, только для чтения.

Списки предоставляют следующие методы:

• addItem (Item, Pos) – вводит строку, определенную в Item в список в позиции Pos;

• addItems (ItemArray, Pos) – вводит записи, перечисленные в строковом массиве данных ItemArray в список в позиции Pos;

• selectItem (Item, SelectMode) – активизирует или дезактивирует выделение для элемента, определенного в строке Item в зависимости от логической переменной SelectMode;

• makeVisible (Pos) – прокручивает область списка так, чтобы элемент, определенный параметром Pos был видим.

Объект модели списка предоставляет следующие свойства:

• Model.BackgroundColor (long) – цвет фона элемента управления;

• Model.Border (short) – тип границы (0: нет границы, 1: трехмерная граница, 2: простая граница);

• Model.FontDescriptor (struct) – структура с подробностями используемого шрифта (в соответствии со структурой com.sun.star.awt.FontDescriptor);

• Model.LineCount (Short) – число строк в элементе управления;

• Model.MultiSelection (Boolean) – разрешает множественный выбор записей;

• Model.SelectedItems (Array of Strings) – список выделенных записей;

• Model.StringItemList (Array of Strings) – список всех записей;

• Model.Printable (Boolean) – элемент управления может быть напечатан;

• Model.ReadOnly (Boolean) – содержимое элемента управления только для чтения;

• Model.Tabstop (Boolean) – элемент управления может быть достигнут при помощи клавиши Tab;

• Model.TextColor (Long) – цвет текста элемента управления;

• Model.HelpText (String) – текст всплывающей подсказки, которая появляется, когда Вы перемещаете курсор мыши над элементом управления;

• Model.HelpURL (String) – URL страницы онлайн справки для соответствующего элемента управления.

7.11 Поле со списком

Элемент управления поле со списком (сервис com.sun.star.awt.UnoControlComboBox) представляет для пользователя список выбора. Дополнительно, он содержит текстовое поле, позволяя пользователю ввести вариант, который отсутствует в списке. Поле со списком используется, когда имеется только список предложенных выборов, тогда как список используется, когда пользовательский ввод ограничен только списком.

Возможности и свойства поля со списком и списка подобны. Также в поле со списком список элементов может быть показан в раскрывающемся списке, если свойство Dropdown установлено в True. Фактический список элементов доступен через свойство StringItemList.

Текстом, отображаемым в текстовом поле поля со списком управляет свойство Text.

Поля со списком поддерживают следующие свойства:

• ItemCount (Short) – число элементов, только для чтения;

• Text (String) – текущий текст;

• MaxTextLen (short) – максимальное количество символов, которые Вы можете ввести в поле;

Поля со списком предоставляют следующие методы:

• addItem (Item, Pos) – вводит строку, определенную в Item в список в позиции Pos;

• addItems (ItemArray, Pos) – вводит записи, перечисленные в строковом массиве данных ItemArray в список в позиции Pos;

• getDropDownLineCount () –возвращает число видимых линий в выпадающем списке поля со списком;

• getItemCount () – возвращает число элементов в поле со списком;

• getItem (Pos) – возвращает элемент в указанной позиции Pos;

• getItems () – возвращает все элементы поля со списком;

• removeItems (Pos, Count) – удаляет Count элементов в позиции Pos;

• setDropDownLineCount (Lines) – устанавливает число отображаемых линий в выпадающем списке поля со списком;

Объект модели списка предоставляет следующие свойства:

• Model.Align (short) – определяет горизонтальное выравнивание текста в элементе управления;

• Model.Autocomplete (boolean) определяет, разрешено ли автоматическое завершение текста;

• Model.BackgroundColor (long) – цвет фона элемента управления;

• Model.Border (short) – тип границы (0: нет границы, 1: трехмерная граница, 2: простая граница);

• Model.Dropdown (boolean) – определяет, имеет ли элемент управления кнопку раскрывающегося списка;

• Model.FontDescriptor (struct) – структура с подробностями используемого шрифта (в соответствии со структурой com.sun.star.awt.FontDescriptor);

• Model.HelpText (string) – текст всплывающей подсказки, которая появляется, когда Вы перемещаете курсор мыши над элементом управления;

• Model.HelpURL (string) – URL страницы онлайн справки для соответствующего элемента управления;

• Model.HideInactiveSelection (boolean) – определяет, должен ли вариант выбора в элементе управления быть скрыт, когда элемент управления не активен (не в фокусе);

• Model.LineCount (short) – число строк в элементе управления;

• Model.MaxTextLen (short) – определяет максимальное количество символов;

• Model.StringItemList (Array of Strings) – список всех записей;

• Model.Printable (Boolean) – элемент управления может быть напечатан;

• Model.ReadOnly (Boolean) – содержимое элемента управления только для чтения;

• Model.Tabstop (Boolean) – элемент управления может быть достигнут при помощи клавиши Tab;

• Model.Text (String) – текст связанный с элементом управления;

• Model.TextColor (Long) – цвет текста элемента управления;

7.12 Макрос реализующий использование текстового поля и списков

```
Sub Macro10(Event as Object)
Dim Doc,Sheet,oControl, oForm,Docctl,objtext as Object
Dim objsp1,objsp2 as Object
Doc = ThisComponent
'получение контроллера текущего документа, для работы с Control элементами
Docctl = Doc.getCurrentController()
'получение ссылки на форму третьего листа
oForm = Doc.DrawPages(2).Forms.getByName("Форма")
```

```
'получение ссылку на текстовое поле, при этом мы не получаем все свойства объекта
oControl = oForm.getByName("Текстовое поле 1")
'получение ссылки на объект контроллер текстового поля
'работа с объектом как с типичным объектом в диалоге
objtext = Docctl.getControl(oControl)
'задание цвета текста текстового поля
objtext.Model.TextColor = rgb(255,0,0)
'если добавлена пустая строка то указывается слово Строка
'попробуйте учесть если в строке только одни пробелы
if objtext.Text = "" then
objtext.Text = "Строка"
end if
'получение ссылки на поле со списком
oControl = oForm.getByName("Поле со списком 1")
'получение объекта контроллера выпадающего списка
objsp1 = Docctl.getControl(oControl)
oControl = oForm.getByName("Список 1")
'получение объекта контроллера списка
objsp2 = Docctl.getControl(oControl)
'добавление в конец списка введенной строки
objsp1.addItem(objtext.text,objsp1.itemCount)
'установка выбранной видимой строки в выпадающем списке
objsp1.Text = objtext.text
'добавление в конец списка введенной строки
objsp2.addItem(objtext.text,objsp2.itemCount)
end sub
```

7.13 Элемент Счетчик

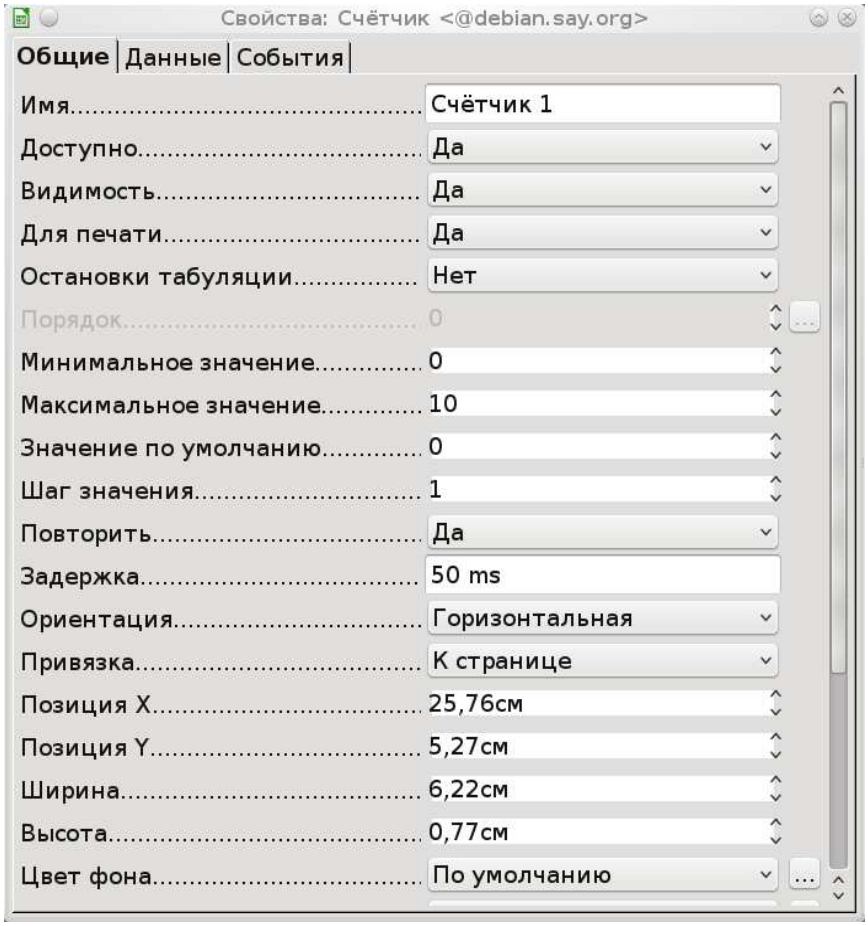


Рисунок 7.14 - Свойсвта объекта счетчик

Объект счетчик предназначен для изменения числового значения величины с каким-то шагом от минимального до максимального значения, с помощью управляющих элементов визуально представленных в виде стрелочек, соответственно одна из которых уменьшает, другая увеличивает текущее контролируемое значение.

7.14 Задание

При выполнении задания использовать диалоги, списки, выпадающие списки, текстовые поля, кнопки и другие элементы управления, которые могут пригодиться для реализации удобного интерфейса пользователя.

Вариант 1.

С помощью форм и диалогов реализовать мастер позволяющий выбирать рейсы из одного города в другой на различных видах транспортных средств (самолетах, автобусах, поездах, пароходах) в различные страны и формировать билет или бронь на рейс для человека. Все данные о рейсах хранить в виде таблиц, эти данные считывать с листа Calc и формировать

интерфейс взаимодействия с пользователем. Число элементов можно хранить в отдельной ячейке или считывать пока не появится пустое поле. Например, хранить данные о видах транспорта в отдельной таблице и считывать в список Диалога эти данные. После того, как человек ввел данные с помощью мастера, данные сохраняются в отдельной таблице, обеспечить возможность навигации по людям в мастере и редактирование введенных данных. Обеспечить фильтрацию и поиск по заказанным рейсам. Например, сделать возможным выводить информацию о самом популярном городе, или виде транспорта. Искать наиболее дешевый маршрут.

Вариант 2.

Обеспечить с помощью форм и диалогов возможность ввода данных о продаваемом на рынке жилье. Обеспечить ввод телефона и имени продавца, параметров жилья в различных городах. Данные сохраняются на листе. Обеспечить редактирование уже введенной записи, также с помощью мастера. Реализовать форму для фильтрации данных для клиента по различным параметрам, отфильтрованные данные выводить на отдельном листе calc или в списке.

Вариант 3.

Продажа компьютерных комплектующих, обеспечить ввод данных о комплектующих (типе, цене, названии, фирме и т.д.). Данные о типах комплектующих хранить в отдельной таблице, затем при работе мастера обеспечить автоматическую возможность выбора типа комплектующего в списке. Реализовать фильтрацию или поиск данных по параметрам комплектующих.

Вариант 4.

Продажа одежды.

Вариант 5.

Продажа спортивных товаров.

Вариант 6.

Продажа продуктов питания.

Вариант 7.

Продажа напитков.

Вариант 8.

Регистрация данных о физических лицах.

Вариант 9.

Регистрация данных о фирмах.

Вариант 10.

Продажа автомобилей.