Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Томский государственный университет
систем управления и радиоэлектроники (ТУСУР)**

А.Я. Суханов, А.С. Новикова, М.А. Беляева, П. А. Куминов

# Информатика

Учебно-методическое пособие по практическим и лабораторным занятиям, самостоятельной и индивидуальной работе студентов всех направлений

Томск
2025

**УДК** 004.4:004.62 (075.8)
**ББК** 16.23
С910

**Рецензент:**
**Исакова А.И.,** доцент кафедры Автоматизированных систем управления ТУСУР, канд. техн. наук

**Суханов, Александр Яковлевич**
С910 Суханов А.Я., Новикова А.С., Беляева М.А., Куминов П.А. Информатика: Учебно-методическое пособие по практическим и лабораторным занятиям, самостоятельной и индивидуальной работе студентов всех направлений [Электронный ресурс] / А. Я. Суханов, А.С. Новикова, М.А. Беляева, П.А. Куминов. — 2-е изд., перераб. и доп. - Томск: ТУСУР: 2025. — 151 с.

Настоящее издание представляет собой второе, переработанное и дополненное издание учебно-методического пособия. В него внесены изменения, учитывающие актуальные требования федеральных государственных образовательных стандартов. Ряд глав был существенно доработан, добавлены новые практические задания и обновлены иллюстративные материалы, в частности добавлено изучение командной строки Linux и скриптов Python. Учебное методическое пособие предназначено для студентов младших курсов бакалавриата всех направлений при изучении дисциплины «Информатика», данное пособие может быть полезно для начального ознакомления с основными методами и процессами представления и обработки информации. Пособие знакомит читателя с основами теоретической информатики, в частности кодированием числовой и текстовой информации, рассматривает равномерное и неравномерное кодирование, предлагает классические лабораторные направленные на изучение текстовых и табличных процессоров, а также работу в командной строке. Кроме того, даны азы программирования для написания макросов в табличных и текстовых процессорах.

**УДК** 004.4:004.62 (075.8)
**ББК** 16.23

**Оглавление**

## Введение

Информатика – это теоретическая и прикладная научная дисциплина, изучающая процессы сбора, обработки, хранения и передачи информации. Мы живем в век, когда информационные процессы играют основополагающую роль в жизни человеческого общества. Трудно себе представить современную жизнь без устройств обрабатывающих, предоставляющих и хранящих в том или ином виде разнообразную информацию, требующуюся человеку как для выполнения его рабочих обязанностей, так и для улучшения его качества жизни.

Данное пособие предназначено для выполнения студентами бакалаврами практических и лабораторных работ по информатике, изучения теоретической информатики, теории кодирования и способов представления информации, а также освоения практических навыков работы с программными продуктами, использующимися для реализации информационных процессов, в частности редактирования текстов и обработки табличных данных.

# 1 Лабораторная работа 1. Оформление документации в текстовых процессорах. Текстовый процессор LibreOffice Writer.

## 1.1 Текстовые процессоры

Обычно текстовые процессоры являются частью офисных комплексов, в которые также входят электронные таблицы, почтовые клиенты, инструменты для создания презентаций и файловые базы данных. На рынке существуют как зарубежные, так и отечественные варианты данного программного продукта. Некоторые из них доступны к бесплатному скачиванию, в то время как другие — только за плату.

Традиционно самым востребованным среди пользователей считается пакет Microsoft Office. Однако он предоставляется на коммерческой основе, и в текущий момент в ряде организаций, относящихся к критической инфраструктуре, указом президента РФ от 30 марта 2022 №166 [1] введён запрет на использование иностранного программного обеспечения; к тому же Министерство образования советует применять программы, включённые в Росреестр программного обеспечения [2].

В качестве альтернативных решений там указаны такие пакеты, как R7-Офис, МойОфис и AlterOffice; отмечается, что значительное число организаций уже перешли на их использование.

К примеру, в облачном сервисе ТУСУР установлен R7-офис, показанный на рисунке 1.1. AlterOffice представляет собой клон LibreOffice.

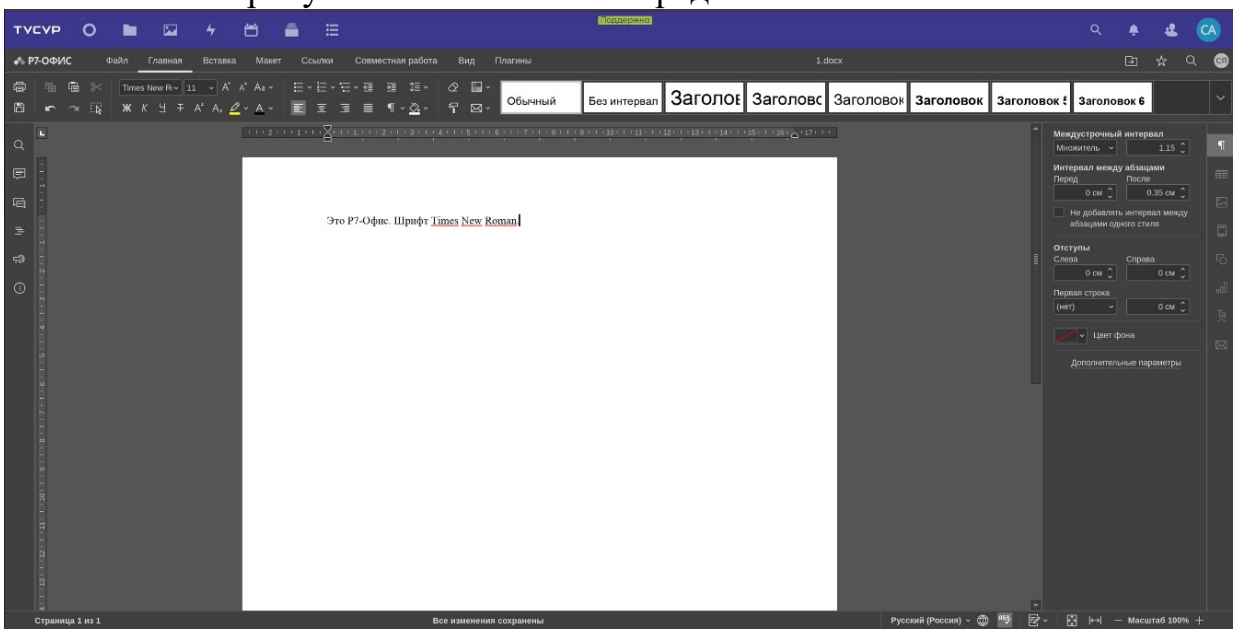



Рисунок 1.1 - Окно с документом R7-офис

Учитывая похожесть большинства пакетов в плане основных функций, мы сосредоточим свое внимание на изучении LibreOffice.

LibreOffice Writer — это приложение-текстовый процессор, позволяющее создавать, просматривать и изменять текстовые файлы, а также

использовать простейшие макросы как формы алгоритмов. LibreOffice представляет собой свободный, независимый офисный набор программ с открытым исходным кодом, поддерживаемый The Document Foundation; он возник как форк проекта OpenOffice.org и включает в себя редактор Writer. Подробные сведения о LibreOffice доступны на сайте [3].

Всякое приложение-текстовый процессор — это прикладное программное обеспечение, созданное для оформления документов, охватывающее набор, коррекцию, оформление и печать любой печатной информации. Периодически такие программы называют редакторами текста второго уровня.

В период 1970-х — 80-х годов под термином «текстовый процессор» обозначали специальные машины для набора и печати, предназначенные как для индивидуального, так и для офисного применения; они сочетали клавиатуру, встроенный микрокомпьютер для базового редактирования и электропринтер. Позднее это обозначение стало применяться к программам, выполняющим схожие функции. По сравнению с простыми текстовыми редакторами, такие процессоры предоставляют более широкие возможности: они позволяют форматировать текст, вставлять графику, формулы, таблицы и прочие элементы. Благодаря этому их используют не только для набора текста, но и для создания разнообразных документов, в том числе официальных. Программное обеспечение для работы с текстом условно делят на лёгкие процессоры, профессиональные редакторы и полноценные издательские системы.

## 1.2 Начало работы в LibreOffice Writer

### 1.2.1 Запуск и начало работы в LibreOffice

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, как показано на рисунке 1.2.

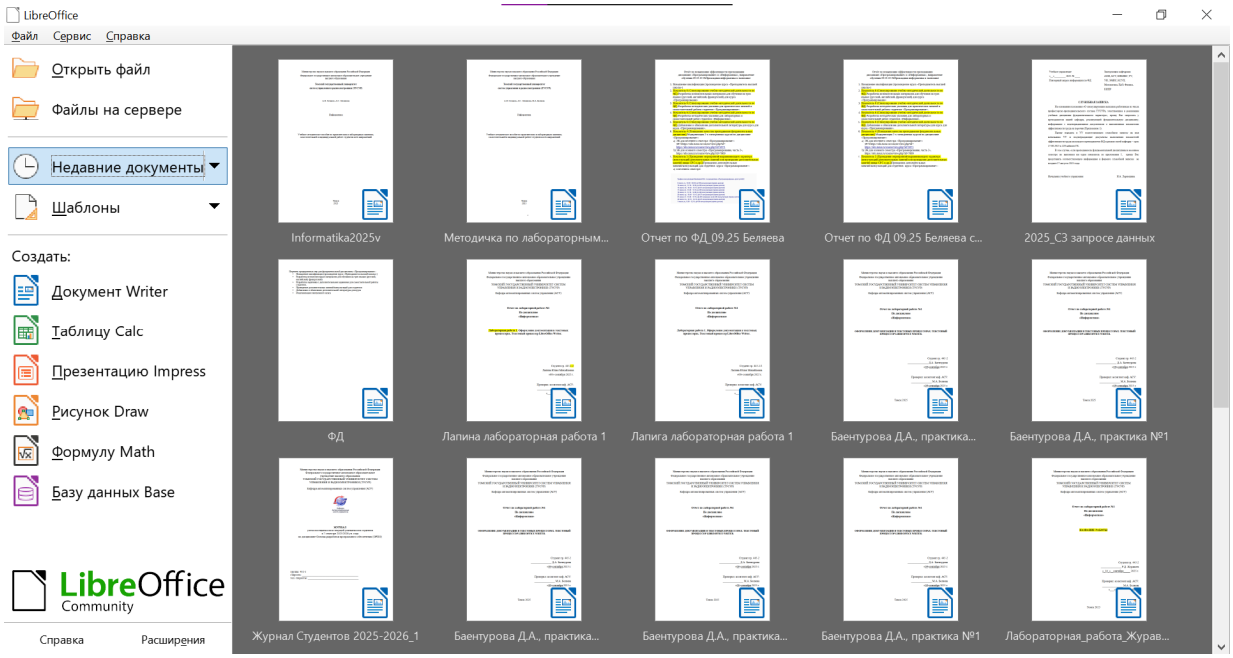



Рисунок 1.2 — Окно создания файлов в LibreOffice

Среди представленных вариантов документов есть «Документ Writer». При его выборе сразу же откроется окно с пустым бланком документа, как на рисунке 1.3.

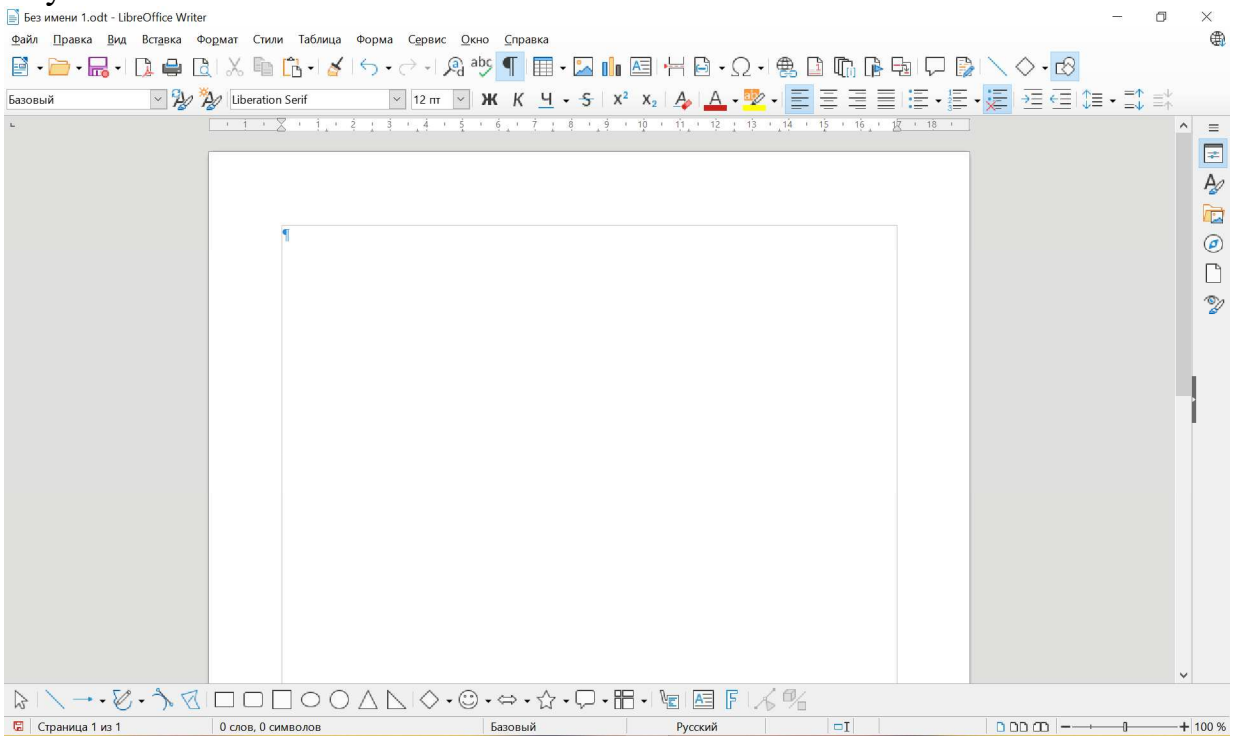



Рисунок 1.3 - Окно с документом LibreOffice Writer

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.

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.

1. Дважды щелкните на каком-нибудь слове.

2. Нажмите клавишу Delete.

Существующий текст сдвинется обратно и заполнит освободившееся место.

### 1.2.2 Сохранение документа в разных форматах

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 1.4.

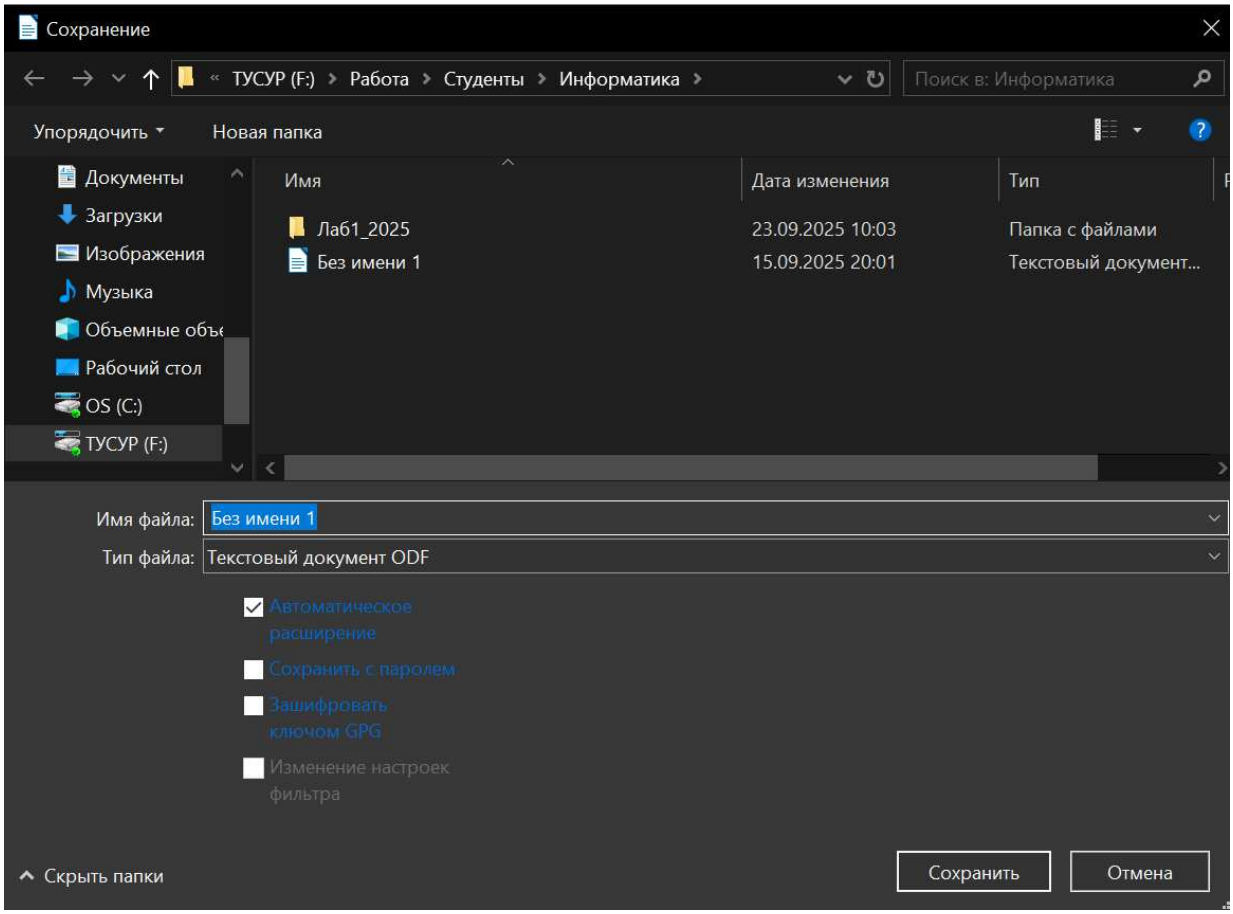


Рисунок 1.4 - Окно сохранения файла LibreOffice

В текстовом поле Имя файла (name) Writer автоматически предлагает имя для документа (обычно Без имени 1 или Untitled 1), но есть возможность ввести новое имя файла. Кроме того, для удаления текста здесь также можно использовать клавиши Backspace и Delete.

Также в поле Тип файла можно изменить тип документа, выбрав из предложенных в выпадающем списке, но по стандарту стоит odf.

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).

Также Writer предполагает возможность экспортировать текстовый файл в формат pdf, что позволит сохранить форматирование и корректность чтения. Есть два способа как это сделать:

1. На панели инструментов в верхней части рабочей области найти

значок pdf, как показано на рисунке 1.5.

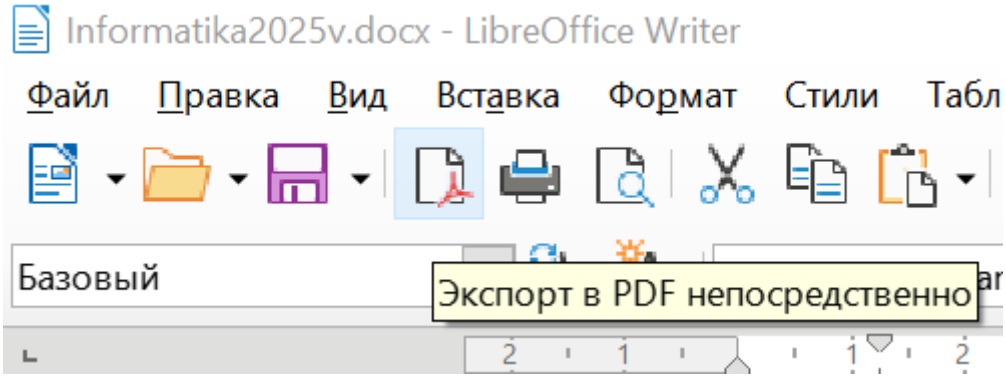



Рисунок 1.5 — Иконка Экспорта в PDF

2. Меню найти пункт Файл → Экспрот в → Экспорт в PDF, как показано на рисунке 1.6.

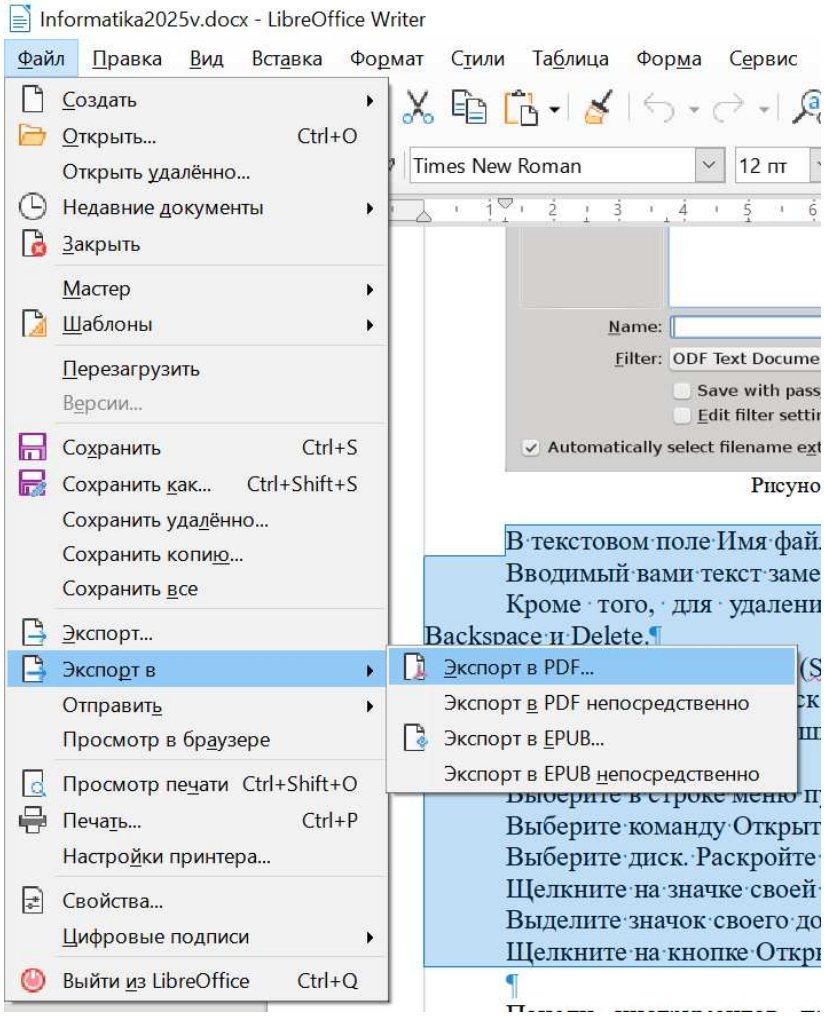



Рисунок 1.6 — Меню Экспорта в PDF

## 1.3 Параметры страницы

Меню Формат → Стиль страница позволяет задать такие свойства как отступы от краев листа, колонтитулы, поворот страницы – альбомная или

книжная, тип печатаемой страницы, например, А4 – стандартный лист для принтера, А5 – половина данного листа, А3 – два листа А4.

На рисунке 1.7 представлена стандартная настройка параметров страницы для любой работы в ТУСУРе.

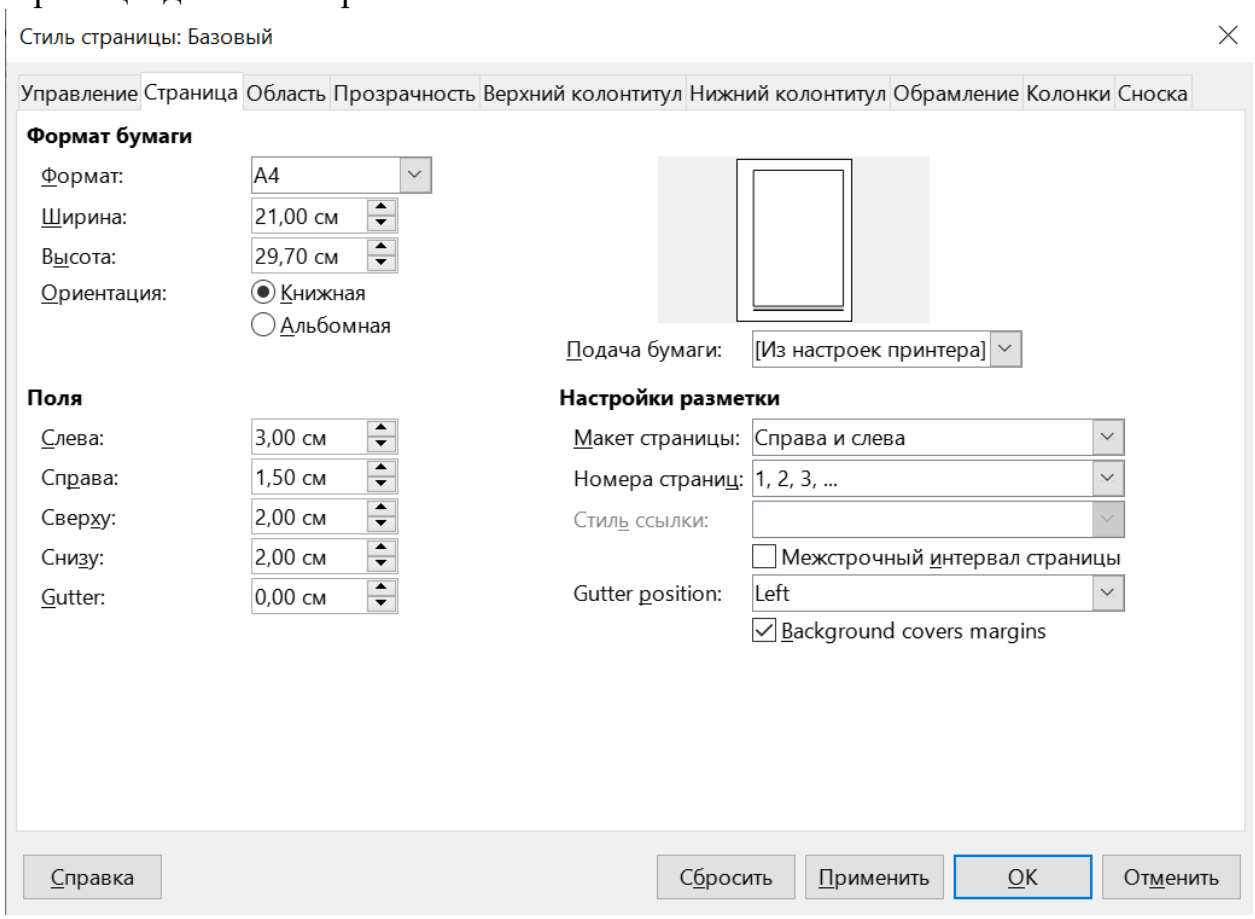



Рисунок 1.7 — Настройка параметров страницы

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

1. В меню выберите "Стиль" → "Стиль страницы" или нажмите клавишу F11, чтобы открыть панель стилей.
2. В открывшейся панели найдите вкладку "Стиль страницы".
3. Щёлкните правой кнопкой мыши по существующему стилю (например, "Обычный") и выберите "Создать".
4. В появившемся диалоговом окне задайте имя вашему стилю, например, "Мой стиль".

   Перейдите во вкладки:

   "Общий" — установите название, описание.

   "Границы" — настройте рамки страницы, если нужно.

   "Фон" — выберите цвет фона или изображение.

   "Разметка" — установите параметры полей, ориентацию страницы (портрет или альбомная).

   Вкладка "Колонки" — можно настроить колонный стиль, если

требуется.

Вкладка "Страницы" — настройте параметры разрывов, номера страниц и т. д.

5. После настройки нажмите "ОК".
6. Чтобы применить созданный стиль к текущей странице или разделу, выделите нужный текст или курсор поставьте на страницу, далее в панели стилей выберите созданный стиль и дважды щёлкните по нему.
7. Дополнительные настройки (по необходимости). Для более точной настройки используйте диалог "Настройка стиля страницы", который открывается при редактировании стиля (правый клик — "Изменить").

Чтобы стиль применялся ТОЛЬКО к одной странице, нужно разделить документ разрывами страниц со сменой стиля. Поместите курсор на начало страницы, где вы хотите применить одностраничный стиль.

Перейдите в меню Вставка → Еще разрыв → Разрыв.

В появившемся окне, которое показано на рисунке 1.8:

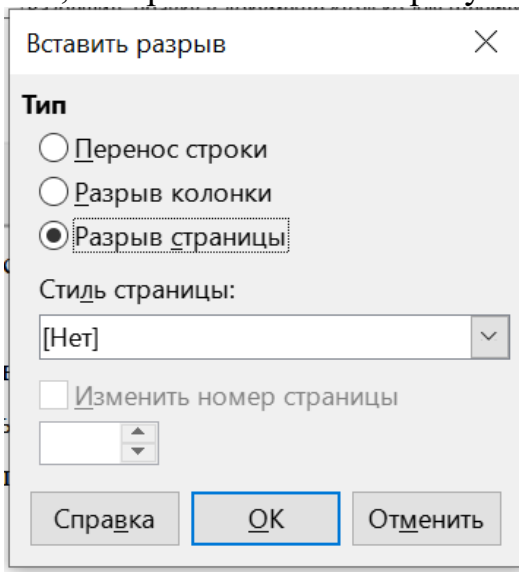



Рисунок 1.8 - Вставка разрыва на другую страницу

Выберите Тип разрыва — Разрыв страницы.

В пункте Стиль страницы выберите созданный вами.

Нажмите ОК.

Далее, чтобы вернуться к обычному стилю после этой страницы:

Поместите курсор в начало следующей страницы (после страницы с одностраничным стилем).

Снова вставьте разрыв страницы, выбрав в качестве стиля страницы оригинальный стиль, например Обычный.

В случаи если нужен обычный разрыв страницы без форматирования, то есть два варианта:

1. Вставить разрыв комбинацией Ctrl + Enter
2. Или через меню Вставка → Разрыв страницы.

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 и соответствующие параметры нового раздела, где он начинается, как показано на рисунке 1.9. 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).

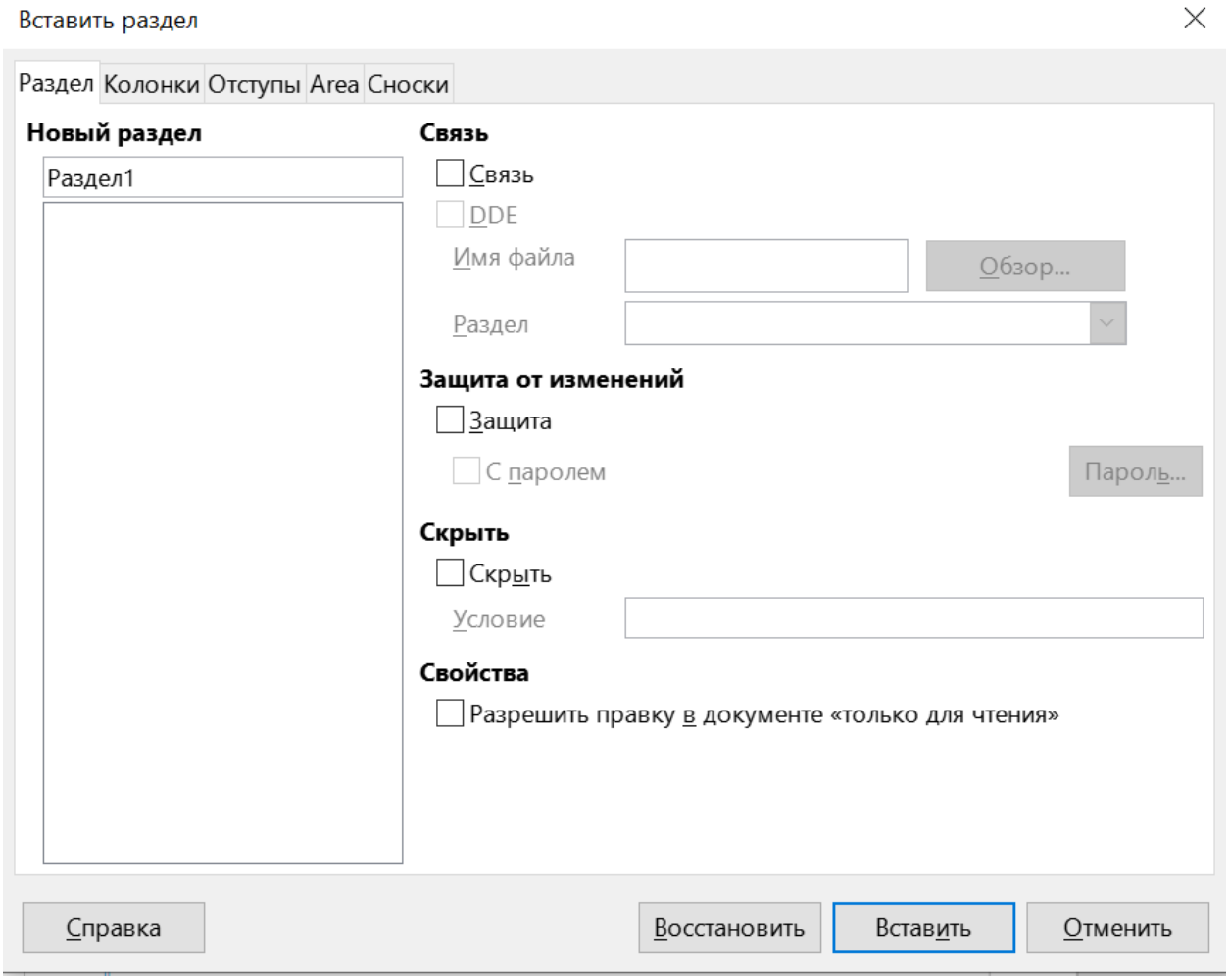



Рисунок 1.9 - Окно для вставки раздела и установка параметров раздела

Также страницы необходимо нумеровать, для этого необходимо:

– Перейдите в меню Вставка → Колонтитулы → выберите Нижний колонтитул. Если документ содержит несколько разделов, убедитесь, что вы вставляете нижний колонтитул в нужном разделе.

– После появления области нижнего колонтитула кликните в нее.

– Перейдите в меню Вставка → Номер страницы.

– Номер страницы появится в нижнем колонтитуле.

– Если нужно, вы можете выровнять номер страницы по центру

или по правому краю с помощью панели форматирования или сочетаний клавиш (например, Ctrl+E для центрирования).

– Для выхода из области колонтитула кликните в любой части документа вне колонтитула или нажмите клавишу Esc.

Также по нормативам первая страница не должна нумероваться, но если удалить номер страницы на ней, то он удалиться на всех остальных. Тогда как это сделать?

Можно прибегнуть к возможности создания титульной страницы, которая не будет иметь номер, но будет учитывается при нумерации остальных страниц. Для этого есть несколько способов.

**Способ 1: Использование стиля страницы**

– Создайте первую страницу документа и оформите её как титульную (заголовок, текст и др.).

– Перейдите в меню Формат → Стили и форматирование (или нажмите F11).

– В открывшейся панели выберите Стили страниц (значок страницы сверху).

– Щёлкните правой кнопкой мыши на стиль страницы, например, Первая страница или создайте новый стиль страницы для титульной страницы.

– Примените стиль страницы к первой странице документа (если нет – вставьте разрыв страницы с изменением стиля).

– Чтобы номера страниц не отображались на титульной странице, убедитесь, что в стиле этой страницы колонтитулы отключены.

– Следующая страница (начиная со второй) должна быть оформлена в другом стиле страницы с нижним колонтитулом и номерами страниц.

**Способ 2: Вставка разрыва с изменением стиля страницы**

– Поместите курсор в конце титульной страницы.

– Выберите меню Вставка → Разрыв…

– В открывшемся окне выберите Разрыв страницы, в разделе Стиль страницы – выберите стиль, в котором будут номера страниц (обычно "Стандартный").

– Убедитесь, что в титульной странице колонтитулы не включены, а во второй странице они есть.

– Так номера страниц начнутся со второй страницы.

Если нужен номер страницы, начиная со второй страницы и при этом не показывать номер на титульной, дополнительно:

Перейдите во второй странице в колонтитул → вставьте номер страницы → кликните правой кнопкой → выберите Параметры нумерации → установите стартовый номер (1 или 2 по необходимости).

## 1.4 Работа с текстовым содержимым документа

### 1.4.1 Панели инструментов и их настройка

Панель инструментов — это набор кнопок и элементов управления, расположенных обычно в верхней части окна LibreOffice Writer, которые обеспечивают быстрый доступ к часто используемым функциям.

Основные особенности панели инструментов:

– Шаблонная организация: панель состоит из кнопок для форматирования текста, вставки объектов, работы с файлами и др.

– Множественность панелей: LibreOffice поддерживает несколько панелей (Стандартная, Форматирование, Рисование и др.), каждая решает свои задачи.

– Настраиваемость: панели и отдельные кнопки можно изменять, добавлять или удалять по желанию пользователя.

В ходе работы с документом панели помогают быстро выполнять необходимые команды без перехода в меню. На рисунке 1.10 отображено стандартная панель инструментов.

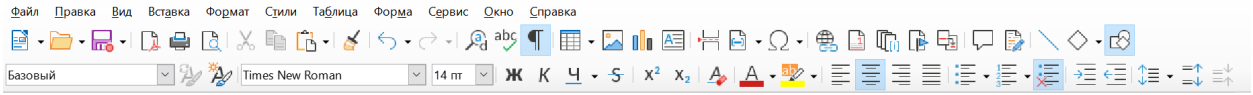



Рисунок 1.10 — Панель инструментов

Для добавления новых элементов на панель инструментов необходимо сделать следующее.

Найдите нужную панель инструментов (например, "Стандартная" или "Форматирование"). Щёлкните правой кнопкой по любой иконке на панели инструментов.

В выпадающем меню выберите Настроить панель инструментов…, показное на рисунке 1.11.

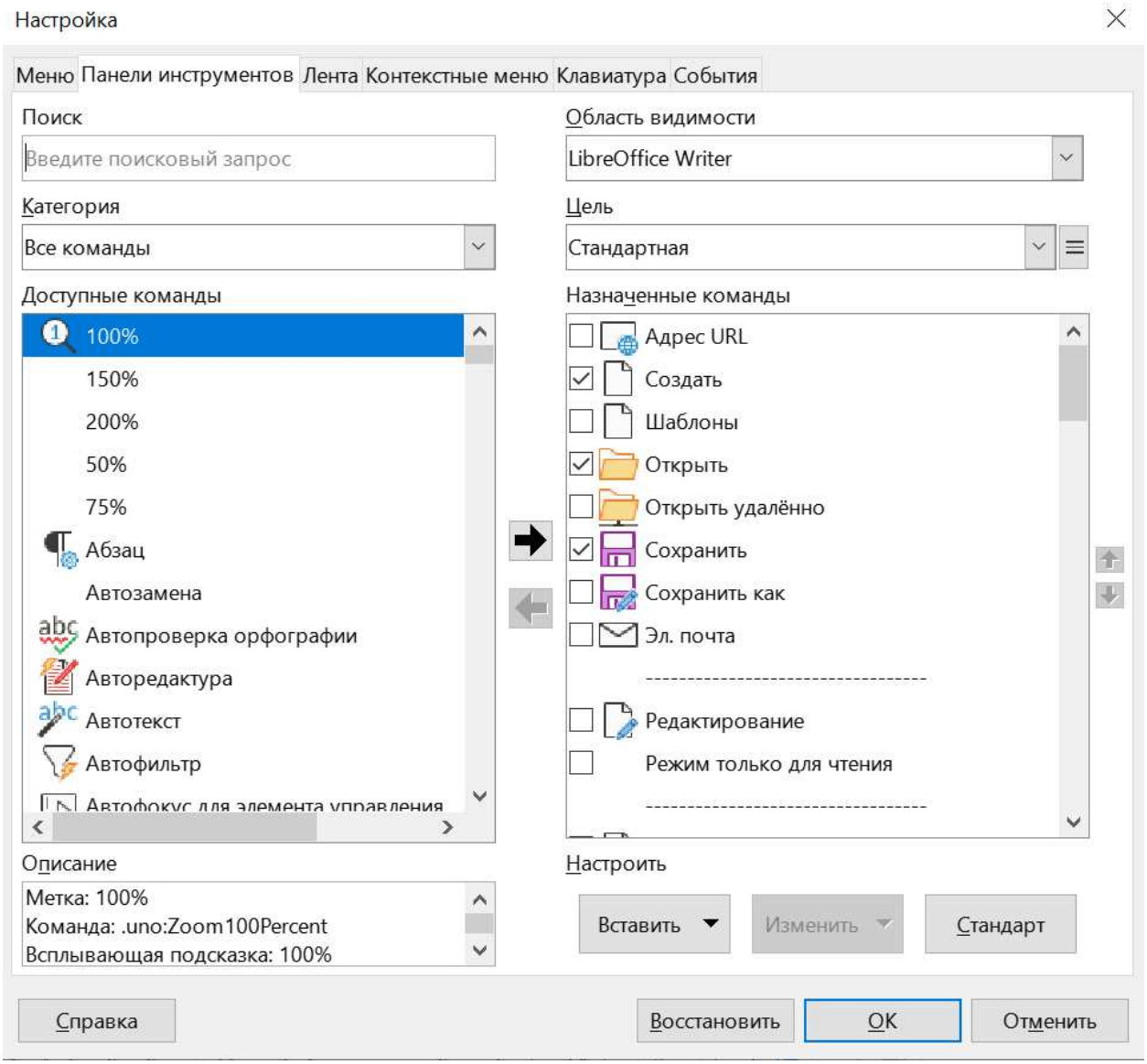



Рисунок 1.11 - Настройка Панели инструментов

В открывшемся окне Настройка в верхней части выберите нужную панель из списка (например, Стандартная).

В списке команд (нижняя часть окна) отображаются уже добавленные кнопки.

Нажмите кнопку Добавить… справа под списком элементов.

В появившемся окне выберите категорию команд слева (например, Файл, Редактирование, Вставка и др.).

Справа появится список команд из выбранной категории. Найдите нужную команду.

Выделите команду и нажмите Добавить.

После добавления она появится в списке текущих кнопок панели инструментов.

Чтобы изменить порядок кнопок, выделите команду в списке и используйте кнопки Вверх или Вниз справа.

Можно также выделить команду и нажать Изменить…, чтобы настроить имя, значок и подсказку.

После завершения нажмите ОК.

Новая кнопка или элемент появится на панели инструментов.

Дополнительная информация

### 1.4.2 Форматирование текста

Для форматирования текста необходимо выделить текст, которых хотим отформатировать или проделать настройку для начал набора текста.

Затем щелкните левой кнопкой мыши на поле страницы.

Щелкните на пункте Символы → Символы... в строке меню. После откроется меню, представленное на рисунке 1. 12.

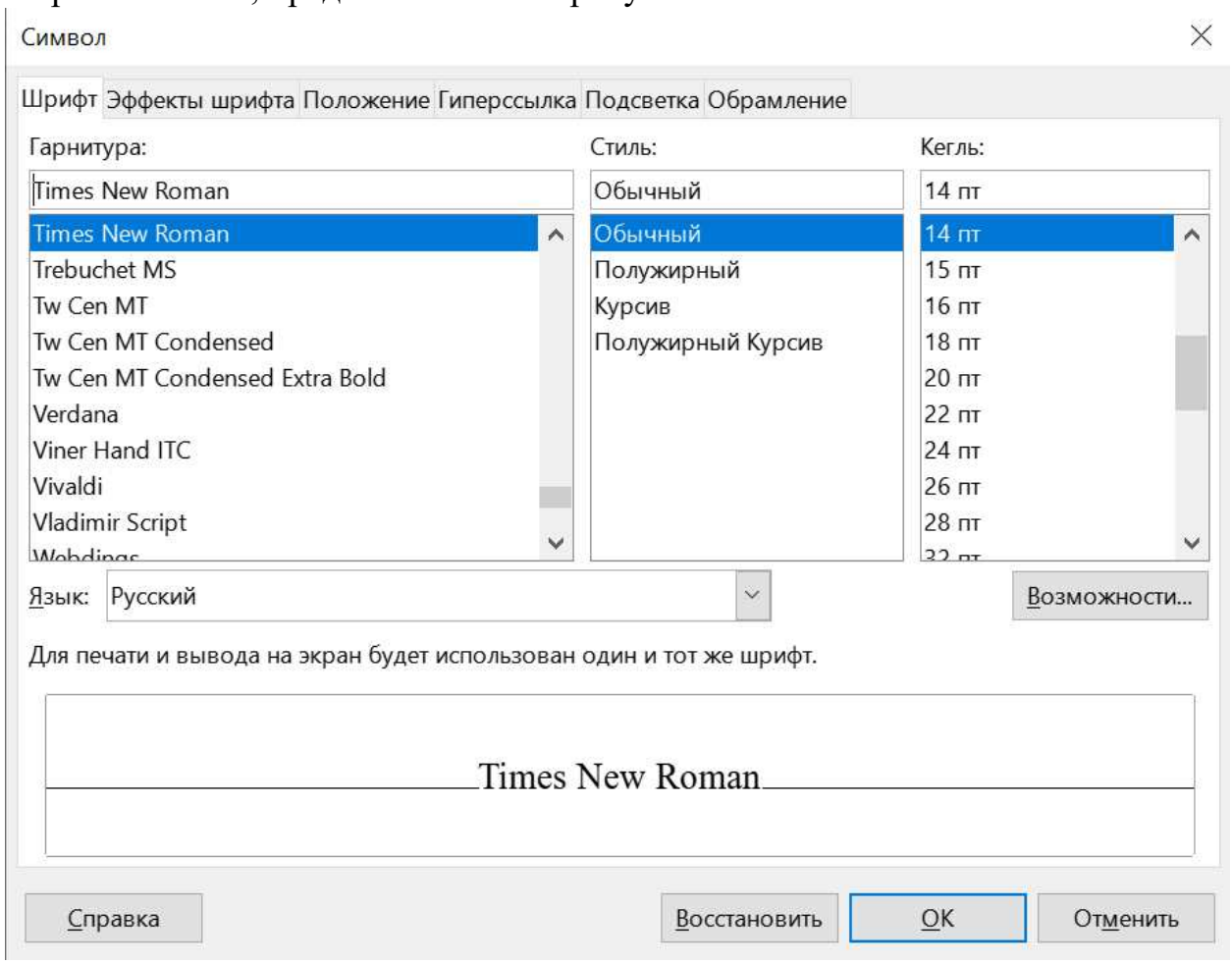



Рисунок 1.12 — Меню настроек символов

В этом меню можно настроит шрифт, стиль, кегль (размер шрифта), эффекты, положение на странице и т.д.

После завершения редактирования 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.

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) , результат вызова меню отображено на рисунке 1.13. Если необходимо переформатировать более одного абзаца, необходимо их предварительно выделить.

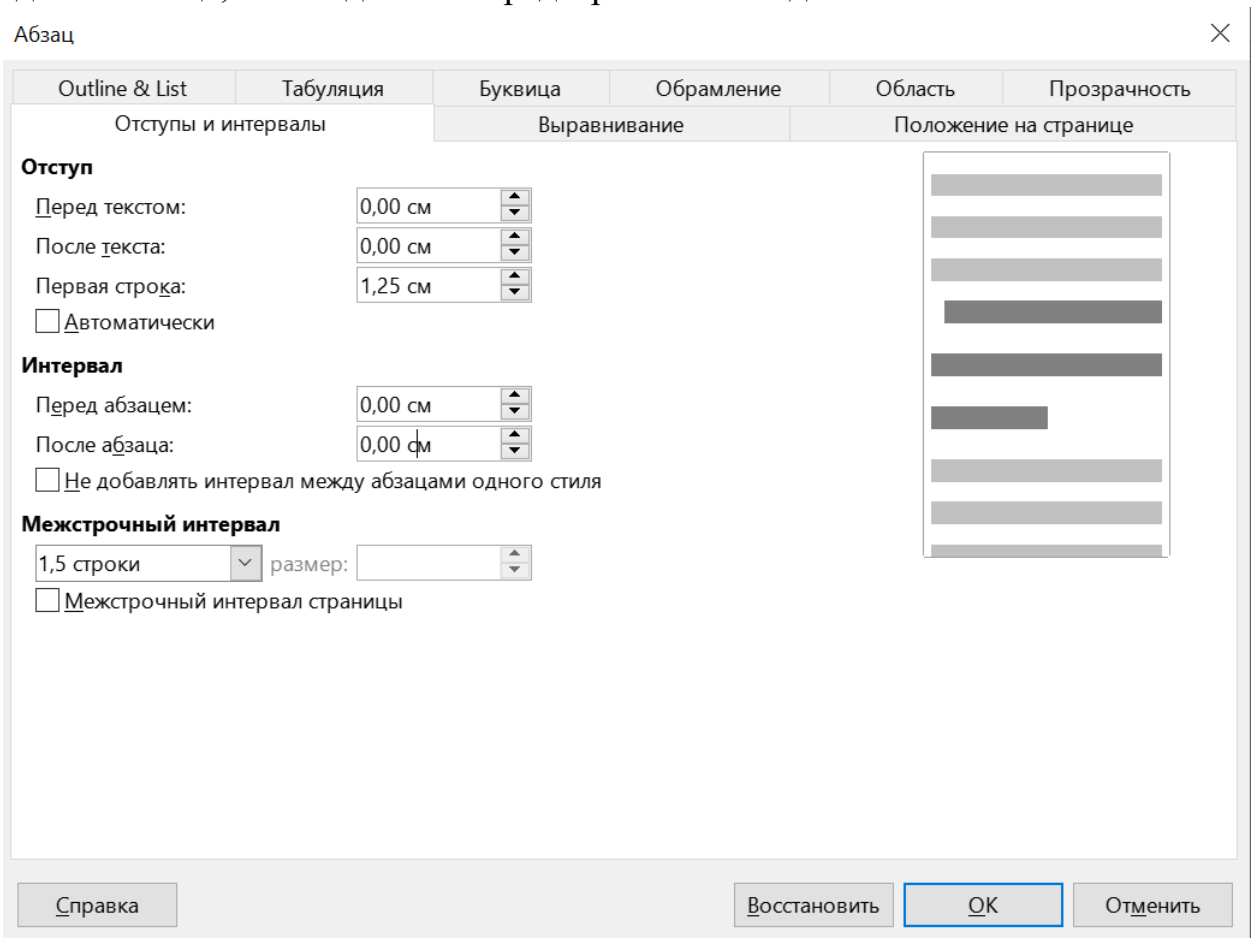



Рисунок 1.13 — Настройка параметров абзаца

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.

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.

**1. Создание стиля для текста (абзаца):**

– Напишите немного текста и отформатируйте его так, как хотите видеть в будущем стиле (шрифт, размер, цвет, межстрочный интервал и т.д.).

– Выделите отформатированный текст.

– На панели справа или через меню Стили → Управление стилями (или нажмите F11) откройте окно стилей.

– В верхней части окна выберите вкладку «Стили абзацев».

– Нажмите правой кнопкой внутри списка стилей и выберите Новый.

– В открывшемся окне задайте имя стиля (например, "Мой основной текст").

– Перейдите по вкладкам и при необходимости настройте дополнительные параметры (выравнивание, отступы, шрифты и т.д.).

– Нажмите OK. Теперь у вас есть собственный стиль для абзацев.

**2. Создание стиля для заголовков:**

– Выделите заголовок и отформатируйте его (например, крупный шрифт, жирный).

– Откройте окно стилей (F11), вкладку «Стили абзацев».

– Нажмите правой кнопкой мыши и выберите Новый.

– Дайте стилю имя (например, "Мой заголовок 1").

– В разделе «Основные» выберите стиль на основе или сделайте его независимым.

– В разделе «Текст» настройте шрифт, размер, цвет.

– Дополнительно в разделе «Положение» настройте отступы, выравнивание.

– Чтобы заголовок автоматически назначался при использовании Навигации по документу и Содержания, нужно установить уровень в разделе «Организация»: выберите «Уровень списка» (например, 1 для заголовка 1).

– Нажмите OK.

**3. Создание стиля списка:**

– Создайте список с нужным форматированием (маркированный или нумерованный).

– Откройте окно стилей (F11).

– Перейдите во вкладку Стили списков (она находится рядом со стилями абзацев).

– Нажмите правой кнопкой мыши → Новый.

– Задайте имя стиля (например, "Мой список").

– Во вкладке «Организация» выберите желаемый тип списка (маркированный или нумерованный).

– Настройте форматирование (отступы, знак списка, формат номера и т.д.).

– Нажмите OK.

**4. Создание стиля таблицы:**

– Создайте таблицу и отформатируйте ее так, как вам нужно (границы, фон, выравнивание текста).

– Поместите курсор в таблицу.

– Откройте окно стилей (F11).

– Перейдите во вкладку Стили таблиц.

– Нажмите правой кнопкой → Новый.

– Задайте имя (например, "Моя таблица").

– Перейдите по вкладкам и настройте рамки, фон, отступы ячеек.

– Нажмите OK.

**5. Применение созданных стилей:**

– Для применения стиля выделите текст, список или таблицу.

– Откройте окно стилей (F11).

– Выберите нужный стиль двойным кликом.

**Советы:**

– Можно создавать стили на основе существующих, чтобы быстро менять оформление.

– Используйте стили для удобства форматирования и автоматической генерации оглавлений и структур.

– Стили можно экспортировать и импортировать через шаблоны для использования в разных документах.

Также в тексте могут понадобится символы, которых нет на клавиатуре для этого используются спецсимволы.

Специальные символы — это символы, которые не имеют отдельной клавиши на клавиатуре, например: ©, ™, ∞, €, различные математические знаки, латинские буквы с диакритическими знаками и т.д.

Для вставка специальных символов через меню необходимо:

– Поставьте курсор в то место, где нужно вставить символ.

– Перейдите в меню Вставка → Специальный символ…

– Вставка специального символа

– В открывшемся окне выберите нужный символ: в верхней части выберите нужный набор символов (например, «Основной латинский», «Математические операторы», «Другие символы» и т.д.).

– Найдите символ в таблице.

– Двойным кликом на символ или нажатием кнопки Вставить вставьте его в документ.

– Закройте окно.

Так же можно применить использование шорткатов (горячих клавиш). Некоторые часто используемые специальные символы имеют стандартные сочетания клавиш. Например:

– Знак евро (€): Ctrl + Alt + E (зависит от раскладки и ОС)

– копирайт ©: Ctrl + Shift + C (в некоторых версиях)

Для других символов можно назначить собственные горячие клавиши:

– Идите в меню Инструменты → Настройка → вкладка Клавиатура.
– В списке функций найдите нужный символ или команду.
– Назначьте клавишу и сохраните.

Альтернативой горячим клавишам может быть ввод Unicode-символов вручную. Если вы знаете Unicode код символа, можно ввести его так:

– В документе введите код символа в шестнадцатеричном формате, например: 221E — это код символа бесконечности (∞).
– Затем нажмите Alt + X.
– Код преобразуется в соответствующий символ.

LibreOffice Writer поддерживает автозамену для специальных символов. Например, при вводе (c) он автоматически заменяет на ©.

Проверить и добавить автозамену можно:

– Перейдите в меню Сервис → Параметры автозамены → вкладка Автозамена.
– Здесь можно настроить списки автозамены, добавлять свои сочетания.

**Другие советы:**

– Для быстрого повторного использования символа, добавьте его в закладку или в документ-шаблон.
– Используйте панель Навигатор или Таблица символов для удобного поиска.

Также текст может идти как одной колонкой (стандартный вид набора текста), так и в несколько.

Колонки — это способ разбить текст на несколько вертикальных частей на одной странице, как в газетах или журналах. Текст автоматически распределяется по колонкам.

**Создание колонок в документе**

Способ А: Для всего документа

– Перейдите в меню Формат → Колонки…
– В открывшемся окне выберите количество колонок (от 1 до 10).
– Можно настроить ширину колонок и расстояние между ними вручную или оставить автоматическое.
– Нажмите OK — весь текст документа будет распределён по выбранному количеству колонок.

Способ В: Для части документа (выделенного текста)

– Выделите текст, который хотите разместить в колонках.
– Перейдите в меню Формат → Абзац…
– В появившемся окне перейдите на вкладку Текстовый поток.
– Отметьте опцию Колонки, выберите нужное количество колонок.
– Нажмите OK.

Или:

– Вставьте в нужное место разрыв раздела (меню Вставка → Разрыв вручную… → выберите Тип: «Раздел», и установите, что этот раздел будет в колонках).

– После создания раздела настройте колонки только в этом разделе (через Формат → Колонки).

**Настройка колонок**

Количество колонок — сколько вертикальных частей будет на странице.

Ширина колонок и промежуток — можно указать фиксированную ширину для каждой колонки или установить равные колонки с промежутком.

Разрывы колонок — чтобы вручную переходить на следующую колонку, используйте ввод разрыва колонки:

Меню Вставка → Еще разрывы → Вставить разрыв колонки.

**Применение колонок к разделам документа**

Чтобы применить колонки только к части документа, сначала вставьте разрывы разделов (меню Вставка → Разрыв вручную… → Выберите «Раздел»).

После этого настройте колонки для этого раздела через Формат → Колонки….

Таким образом можно смешивать части документа с разным количеством колонок.

**Важные моменты**

– При работе с колонками могут изменяться параметры нумерации страниц и колонтитулов в разных разделах.

– Колонки хорошо использовать для оформления бюллетеней, газетных статей и флаеров.

– Колонки не применяются к таблицам, изображениям и другим объектам — они остаются в обычном формате.

### 1.4.3 Автозамена и ее параметры

Автозамена — это функция, которая автоматически заменяет определённые вводимые вами тексты (например, опечатки, сокращения или специальные обозначения) на заданные варианты. Это помогает быстро исправлять ошибки и упростить набор текста.

**1. Открытие настроек автозамены**

Перейдите в меню Сервис → Автозамена → Параметры автозамены.

Альтернативно: Инструменты → Автозамена → Параметры автозамены.

Откроется окно с несколькими вкладками и опциями, как показано на рисунке 1.14.

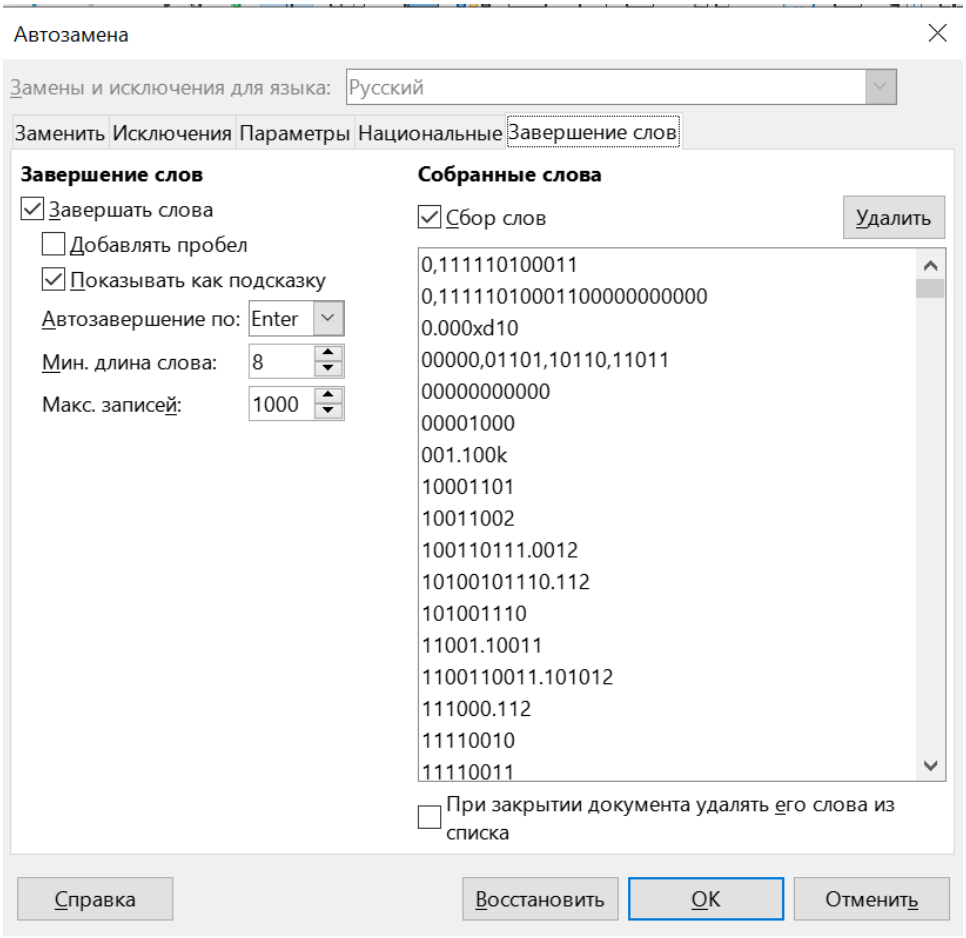



Рисунок 1.14 — Настройки автозамены

**2. Основные вкладки и параметры**

Автозамена — главная вкладка, где вы включаете или отключаете функцию автозамены и настраиваете основные параметры:

– Использовать автозамену — активирует/деактивирует всю функцию.

– Автоматическая замена при вводе — включает замену текста в процессе печати.

– Использовать автокоррекцию (например, заглавные буквы) — исправляет заглавные или малые буквы, например при начале предложения.

– Заменять дефисы, квадратные кавычки, тройные точки (на многоточие) и прочее — автоматически заменяет соответствующие символы.

Таблица автозамены — список всех текущих правил автозамены (пар «исходный текст» → «заменить на»). Здесь вы можете:

– Добавлять новые правила: в полях внизу окна вводите слово или символ для замены и его замену, затем нажмите кнопку Новая запись.

– Редактировать существующие записи, выделив их и изменив значения.

– Удалять ненужные правила кнопкой Удалить.

Другая вкладки (например, параметры форматирования, специфические

настройки для языка) могут быть доступны в зависимости от версии LibreOffice.

**3. Пример добавления своего правила автозамены**

– В открытом окне Параметры автозамены перейдите на вкладку Таблица автозамены.

– В поле Заменить введите, например, brb.

– В поле На введите, например, be right back.

– Нажмите Новая запись.

– Теперь при вводе brb и пробела или знака препинания, текст автоматически заменится на be right back.

**4. Дополнительные настройки**

– Автозамена учитывает разделение слов пробелами и знаками препинания.

– Можно включать или отключать замену при разных сценариях, чтобы избежать ошибок.

– Для некоторых языков доступны специфические правила автокоррекции, например, кавычки или тире.

**Итог**

Функция автозамены в LibreOffice Writer помогает ускорить ввод текста, автоматически исправляет ошибки и заменяет шаблонные фразы. Её настройки гибко настраиваются в меню Параметры автозамены.

Если нужно, могу помочь с конкретными примерами или другими функциями автозамены.

## 1.5 Оглавление и указатели

Перед созданием оглавления убедитесь, что в документе правильно использованы стили заголовков. Основные стили заголовков:

– Заголовок 1 (Для главных разделов)

– Заголовок 2 (Для подразделов)

– Заголовок 3 и др. (для подуровней)

Эти стили присваиваются тексту, который должен войти в оглавление.

Как применить стиль заголовка:

– Выделите заголовок.

– В панели стилей (F11) выберите нужный стиль заголовка или в меню Формат → Стиль выберите Заголовок 1, Заголовок 2 и т. д.

**1. Создание оглавления**

1. Поместите курсор в место, где хотите вставить оглавление (обычно в начале документа).

2. Перейдите в меню Вставка → Оглавление и указатели → выберите Оглавление, указатель или библиография.

3. В открывшемся окне настройте параметры:

– Вкладка Общее: Название (например, "Оглавление"), Формат оформления, Количество уровней (обычно 3 или 4)

– Вкладка Типы:

– Убедитесь, что выбран Таблица содержания.

– Вкладка Образцы:

– Можно выбрать оформление внешнего вида файла.

4. Нажмите ОК.

Теперь оглавление автоматически подтянет все заголовки, помеченные стилями Заголовок 1, Заголовок 2 и т. д.

**2. Обновление оглавления**

Если вы вносите изменения в документ (добавляете, удаляете или изменяете заголовки):

– Щёлкните по оглавлению правой кнопкой мыши.

– Выберите Обновить оглавление.

Это обновит номера страниц и структуру.

Примером для правильного оформления оглавления по ОС ТУСУРа может служить оглавление данного пособия.

## 1.6 Работа с изображениями

**1. Вставка изображения**

1. Поместите курсор в то место документа, куда нужно вставить изображение.
2. Перейдите в меню «Вставка» → «Изображение» → «Из файла...».
3. В диалоговом окне выберите нужное изображение на компьютере и нажмите «Открыть».
4. Изображение вставится в документ.

**2. Форматирование изображения**

После вставки можно выполнить различные операции для редактирования и форматирования изображения:

- Изменение размера:
  - Перетащите за угловые маркеры рамки изображения.
  - Для пропорционального изменения зажмите клавишу «Shift» при перетаскивании.
- Обтягивание текста:
  - На панели «Объект изображения» или в панели «Свойства» выберите режим обтягивания текста, например:
    - Обтекание текстом.
    - Внутри рамки.
    - Нет(изображение будет отделено от текста).
- Обрезка:

- Выберите изображение.
- В панели «Свойства» нажмите кнопку «Обрезка» и перетащите области обрезки.

- Настройки яркости, контраста, прозрачности и других параметров:
  - В панели «Свойства» выберите вкладки «Цвет» или «Графика» и настройте параметры.

**3. Вставка подписи к рисунку**

1. Выберите изображение.
2. Перейдите в меню «Вставка» → «Название».
3. В открывшемся окне выберите позицию подписи:
   1. Под рисунком.
   2. Над рисунком.
4. Введите текст подписи и нажмите «ОК».

**4. Форматирование подписи**

Выделите подпись, чтобы изменить шрифт, размер, цвет, или применить другие стили через панель инструментов.

**5. Автоматическая нумерация и создание списка рисунков**

1. Чтобы автоматически нумеровать все подписи, используйте «Стили».
2. Для этого:
   1. Создайте новый стиль подписи: «Правый клик» на панели «Стили» → «Создать стиль».
   2. Назовите его, например, «Рисунок_подпись».
   3. В свойствах стиля укажите нужные параметры.
3. В каждую подпись вставляйте номер автоматически:
   1. В меню «Вставка» → «Таблица» → «Поле» → «Область».
   2. В окне «Область» выберите «Вставить поле» → «Код» или используйте «Область текста» для автоматического номера.
   3. Или используйте «Автоподписи».

**Использование автоподписей**

1. Перейдите в меню «Вставка» → «Объекты» → «Автоподписи».
2. Выберите «Рисунок», чтобы автоматически создавать подписи и нумерацию к рисункам.
3. После вставки вы сможете обновлять нумерацию автоматически.

**6. Обновление номеров**

При добавлении новых изображений и подписей нажмите «Правый клик» по нумерации и выберите «Обновить поле».

Примером правильного оформления по ОС ТУСУРа рисунка может служить любое изображение в данном пособии.

## 1.7 Работа с формулами

**1. Создание формул**

1. Вставка формулы
    1. Поместите курсор в нужное место документа.
    2. Перейдите в меню «Вставка» → «Объект» → «Формула».
    3. Или используйте сочетание клавиш «Ctrl + F2».
    4. В появившемся окне редактора формул появится placeholder для набора формул. Как показано на рисунке 1.15.

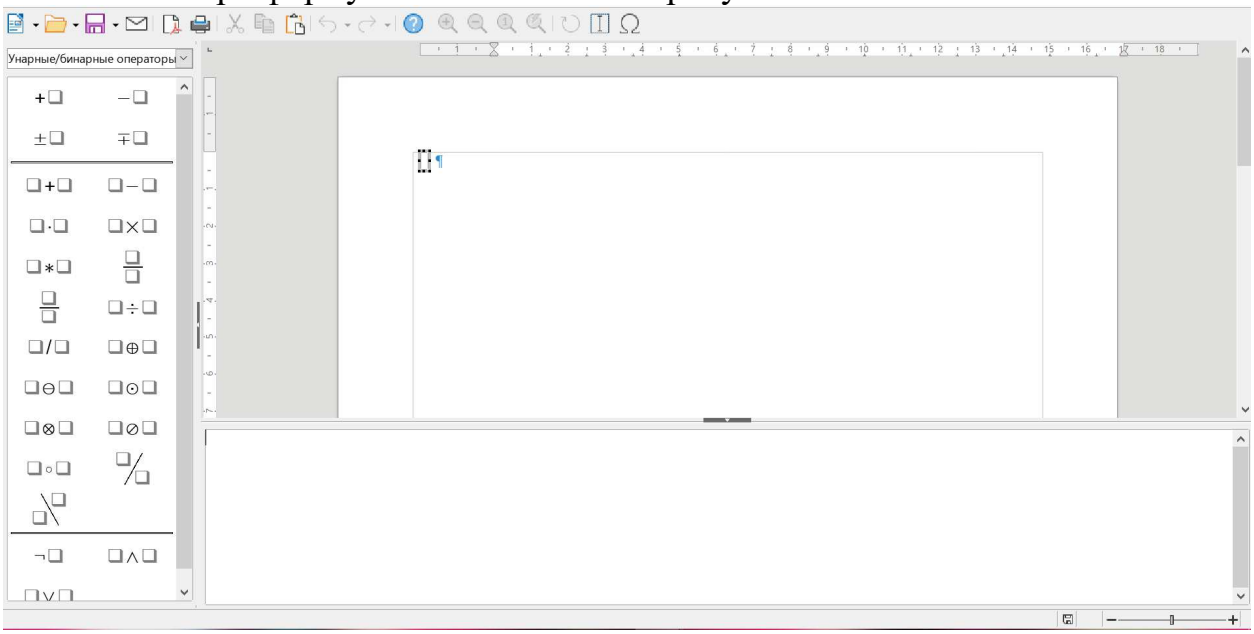



Рисунок 1.15 — Окно редактора формул

2. Автоматическое создание формул
    1. После вставки появится область редактирования формулы, где можно вводить математические выражения.

**2. Инструменты для записи формул**

1. Панель редактора формул

После вставки формулы в верхней части окна редактора появляется панель с инструментами, включающая:

1. Стандартные символы (сумма, произведение, интеграл, границы).
2. Стиль и размеры элементов.
3. Выравнивание и группировка элементов.
4. Специальные операторы и структуры (подстрочные, надстрочные, дроби, корни, скобки).

2. Основные команды для набора формул

Формулы записываются с помощью большинства стандартных математических символов и команд:

1. `+`, `-`, `*`, `/` — арифметические операции.
2. `^` — степень.
3. `_` — индекс.

4. `\frac{a}{b}` — дробь.
5. `\sqrt{}` — квадратный корень.
6. `\sum`, `\prod`, `\int` — суммы, произведения, интегралы.
7. `\left(` и `\right)` — автоматическая подгонка скобок.
8. `\alpha`, `\beta`, `\gamma` — греческие буквы.
9. и др.

3. Использование шаблонов

Можно использовать команды из LaTeX-стиля, подставляя их в редактор формул для быстрого набора сложных выражений.

**3. Редактирование формул**

Для редактирования уже вставленной формулы:

1. Кликните по ней или дважды.
2. В открывшемся редакторе изменяйте команду или добавляйте символы по мере необходимости.
3. После завершения нажмите **Esc** или кликните вне области редактора.

**4. Оформление формул**

Формулы по умолчанию располагаются по центру строки.

Можно менять размер и стиль текста, выделяя формулу и применяя стандартные средства форматирования (жирный, курсив, размер шрифта).

Для выделения формулы используйте комбинацию «Ctrl + Масштаб» или соответствующие инструменты.

**5. Нумерация формул**

1. Вставка номера формулы
    1. После вставки формулы, выделите нее.
    2. В меню «Вставка» → «Названия» .
    3. В настройках области укажите отображение нумерации, например, номер в виде `(1)`, `(2)` и т. п. Как показано на рисунке 1.16.

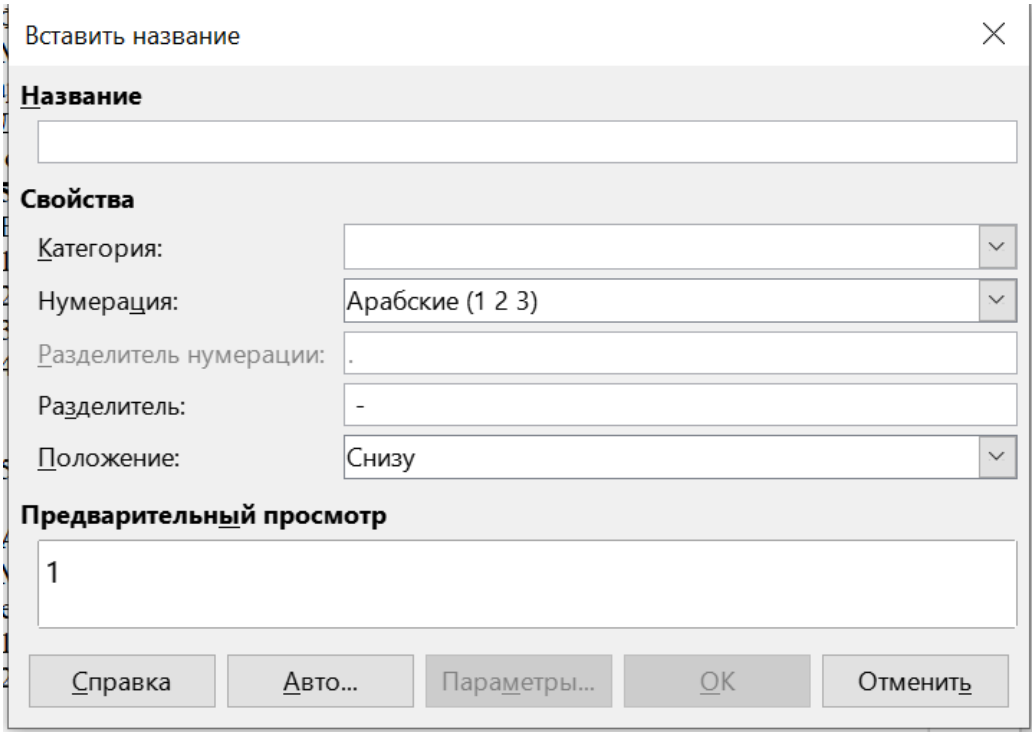


Рисунок 1.16 — Окно настроек нумерации формул

2. Автоматическая нумерация

Можно настроить автоматическую нумерацию для всех формул в документе:

1. Используйте стили или поля.
2. Например, вставляйте номера вручную или через таблицы/области, обновляя их при необходимости.

Для примера, как должна выглядит формула по правилам ОС ТУСУРа запишем формулу для расчета информационной энтропии:

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

где $p_i$ - вероятность встречи символа в сообщении.

**6. Дополнительные советы**

- Для более сложных формул воспользуйтесь экспортом из LaTeX с помощью специальных конвертеров или расширений.
- Сохраняйте шаблоны и чаще используемые формулы для ускорения работы.

## 1.8 Работа с таблицами

**1. Создание таблицы**

1. Вставка новой таблицы
   1. Поместите курсор в нужное место документа.
   2. Перейдите в меню «Вставка» → «Таблица».

3. В появившемся диалоговом окне укажите:
    1. Количество строк.
    2. Количество столбцов.
4. Нажмите «ОК», таблица вставится в документ.

2. Быстрая вставка таблицы
   Можно также воспользоваться инструментальной панелью:
    1. Кликните по значку «Вставить таблицу» или используйте сочетание клавиш «Ctrl + F12».
    2. А затем выберите размеры таблицы.

**2. Работа с таблицами (инструменты)**

1. Выделение таблицы и ячеек
    1. Кликните в любую ячейку, чтобы активировать таблицу.
    2. Для выделения одной ячейки — кликните по ней.
    3. Для выделения всей таблицы — кликните на маркер, появляющийся в левом верхнем углу таблицы при наведении.
2. Изменение размеров
    1. Перетаскивайте границы строк и столбцов вручную.
    2. Или используйте контекстное меню:
        1. правый клик по таблице → «Свойства таблицы».
        2. Во вкладке «Строки» и «Столбцы» задавайте точные размеры.
3. Добавление и удаление строк и столбцов
   Правый клик по выбранной строке или столбцу → выбирайте:
    1. «Вставить строку выше/ниже».
    2. «Вставить столбец слева/справа».
    3. «Удалить».
4. Форматирование таблицы
   Правый клик по таблице → «Свойства таблицы»:
    1. Вкладка «Общие»: установка границ.
    2. Вкладка «Цвет»: заливка ячеек.
    3. Вкладка «Выравнивание»: выравнивание текста.
    4. Вкладка «Шрифты»: изменение шрифта и его стиля.
5. Объединение и разделение ячеек
    1. Выделите несколько ячеек, правый клик → «Объединить ячейки».
    2. Для разделения — выберите объединённую ячейку, правый клик → «Разделить ячейки».

**3. Работа с содержимым таблицы**

- Вводите текст, числа или формулы прямо в ячейку.
- Для вставки изображений или других объектов используйте вставку как в обычном документе.

**4. Нумерация таблиц и их оформление**

1. Вставка подписи (названия) таблицы
2. Вставьте таблицу.
3. Над или под таблицей вставьте текст или поле, например:

«Таблица 1.1 — Название таблицы».

4. Можно автоматизировать нумерацию:
    1. Создайте стиль «Таблица».
    2. Используйте «Объекты» → «Область» для вставки номера.
    3. Связывайте подписи с номером таблицы, чтобы при добавлении новые нумерации обновлялись автоматически.
5. Настройка автоматической нумерации
    1. В меню «Вставка» → «Область» или «Области» создайте область для нумерации.
    2. Настройте шаблон подписи на автоматический вывод номера:
        1. Например, «Таблица <Num.Table#> — описание».
6. Обновление номеров

При добавлении новых таблиц или их перемещении, обновляйте нумерацию, выбирая меню «Правка» → «Обновить все поля» или нажав «F9» в некоторых случаях.

**5. Дополнительные советы**

- Используйте стили ячеек для единообразия оформления.
- Для более сложных таблиц используйте условия форматирования.
- Для удобства сохраняйте шаблоны таблиц с предустановленным оформлением.

Примером правильного оформления таблицы по правилам ОС ТУСУРа может служить таблица 1.1.

Таблица 1.1 — Моя таблица

| № | Название лабораторной работы | Зачтено / не зачтено |
|---|---|---|
| 1 | Текстовый документ Word | Зачтено |
| 2 | Таблицы Excel | Не зачтено |

## 1.9 Задания по лабораторной работе

Выполнение первой лабораторной работы подразумевает создание отчета, отвечающий стандартам ОС ТУСУР. Для необходимо выполнить следующие пункты:

1. Внимательно изучите возможности Libre Writer описанные в методическом пособии. При выполнении можно повторно обращаться к разделам по мере необходимости.

2. 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.

3. Работа выполняется в формате А4 . Текст работы следует писать, соблюдая следующие размеры полей (как это сделать описано в пункте 1.3 этого пособия):

1. левое – не менее 30 мм;
2. правое – не менее 15 мм;
3. верхнее – не менее 20 мм;
4. нижнее – не менее 20 мм.

Настройки самого текста должны быть следующие:

– шрифт Times New Roman;
– размер шрифта 14 пт;
– междустрочный интервал 1,5;
– отступ красной строки 1,25;
– выравнивание по ширине;
– отступов до и после абзаца не должно быть.

Установите шрифты, параметры страницы, абзацев, создайте свои стили (как это сделать написано в пункте 1.4.2 этого пособия).

4. Создайте титульный лист, согласно шаблону из приложения А.

5. Затем записать пункты будущей работы. В их перечень должны входить следующие пункты в данном порядке:

1. Оглавление;
2. Введение;
3. 1. Теоретические основы текстовых процессоров;
4. 1.1 Libre office;
5. 1.2 Microsoft Word;
6. 1.3 Сравнение текстовых процессоров;
7. 2. Ход работы;
8. 2.1 Задание 1. Найти и заменить;
9. 2.2 Задание 2. Альбомная страница;
10. 2.3 Задание 3. Двухколонный текст;
11. 2.4 Задание 4. Таблица с данными;
12. 2.5 Задание 5. Формула;
13. Вывод;
14. Список использованный источников.

Те пункты что описаны одной цифрой (например, 1) или не имею нумерации должны быть стиля «Заголовок 1» и начинаться с новой страницы. А те, что в две цифры (например, 1.2) - «Заголовок 2» и могу не начинаться с новой страницы. Выравнивание по центру, остальные настройки, как у всего остального текста. А также пере и после каждого заголовка должна быть пустая строка.

6. Вставить нумерацию страниц внизу страницы с выравнивание по центру, титульная страница не должна нумероваться (как это сделать описано в пункте 1.3 этого пособия).

7. После титульного листа вставить оглавление (как это сделать описано в пункте 1.5 этого пособия)

8. В разделе «Введение» описать цель работы.

Целью работы является ознакомить студентов с требованиями к

оформлению лабораторных работ. Дать основные навыки по оформлению автоматического содержания, расстановки номеров страниц, альбомного листа внутри текста с книжными страницами, дать навыки выделения и копирования текста, оформления формул, таблиц, рисунков, использованных источников, поиска и замены текста и специальных символов, изменения стиля текста, абзаца и страницы. Задание по лабораторной работе дано в конце главы.

9. Заполнить разделы 1.1, 1.2 и 1.3 информацией из интернет ресурсов. Ссылки на эти ресурсы необходимо внести в «Список использованный источников». Примеры правильного оформления источников можно посмотреть в документе ОС ТУСУР в Приложении Д или в данном пособии в приложении Б.

Помимо текста в каждый раздел необходимо вставить по изображению рабочей области каждого из редакторов. При этом в тексте должны быть ссылки на рисунки. Например:

– как показано на рисунке 1.1;
– (рисунок 1.1) и т.д.

Сам же рисунок должен располагаться сразу после его упоминания в тексте. Оформить рисунки нужно согласно ОС ТУСУР (как это сделать описано в пункте 1.6 этого пособия).

10. Далее сохраните документ и начните выполнение заданий. Каждое выполнение задания должно быть под соответствующим заголовком.

11. **Задание 1. Найти и заменить.** Удалите ненужные абзацы, отобразите невидимые символы. 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. Сделайте скриншот результата замены и отмените действие нажав на кнопку «Отменить» или комбинацией Ctrl + Z. Вставьте скриншот, оформив его как рисунок 2.1, в раздел «Задание 1. Найти и заменить» и опишите свои действия по выполнения данного задания с упоминание рисунка 2.1, как результата работы.

12. **Задание 2. Альбомная страница.** Далее при помощи разрывов вставьте альбомную станицу, проверив нумерацию страниц, при этом остальные должны оставаться книжными. Вставьте заголовок раздела и опишите свои действия в отчете с соответствующими скриншотами по его выполнению. Нумерация рисунков будет продолжаться с последнего рисунка из предыдущего задания (как это сделать описано в пункте 1.6 этого пособия).

13. **Задание 3. Двухколонный текст.** На следующей книжной странице необходимо вписать текст под заголовком Задания 3. Затем

отформатировать его на двухколонный формат (как это сделать описано в пункте 1.4.2). Добавить скриншот настроек колонок. После необходимо сделать первую букву первого абзаца «Буквицей». Для этого нужно выделить первый абзац и вызвать меню настроек абзаца (Правая кнопка мышы → Абзац → Абзац → Вкладка Буквица). Отметит пункт «Добавить буквицу». Количество символов -1, Строк — 3. Так же сделать скриншот и добавить отчет с подписями.

14. **Задание 4. Таблицы с данными.** В этом задании необходимо вставить таблицу с данными о текстовом документе. В этой таблице должны быть следующие пункты: Количество слов в документе, количество символов с пробелами в документе, количество символов без пробелов в документе и средняя длинна слова. Первые три пункта можно узнать, нажав на поле с количеством слов и символов внизу рабочей программы. А последний необходимо рассчитать, самостоятельно поделив количество символов без пробелов на количество слов. Шапка таблицы должна быть жирным начертание и выравнивание по центру. Подпись должна быть оформлена согласно ОС ТУСУР (как описано в пункте 1.8). Также, как и в других задания необходимо описать процесс создания данной таблицы.

15. **Задание 5. Формулы.** В этом пункте необходимо будет описать процесс создания формулы и вставить сами формулы. В этой работе необходимо вставить три следующие формулы: $E = mc^2$ , $c=\sqrt{a^2+b^2}$ и $\frac{a}{b}+\frac{c}{d}=\frac{ad+bc}{bd}$. Оформить их нужно согласно ОС ТУСУР (как описано в пункте 1.7).

16. После выполнения всех пяти заданий необходимо в соответствующем разделе описать выводы по проделанной работе. Любые выводы по любой работе следуют из целей этой работы.

17. В завершении необходимо дополнить «Списки источников» ссылкой на методические указания по лабораторной работе. Правильное оформление выглядит следующим образом:

Информатика: Учебное методическое пособие по практическим и лабораторным занятиям, самостоятельной и индивидуальной работе студентов всех направлений / А. Я. Суханов, А.С. Новикова, М.А. Беляева, П. А. Куминов – Томск: ТУСУР, 2025. – 151 с.

## 2 Лабораторная работа 2. Командная строка и командные файлы.

Целью работы является освоить основные возможности командной строки в одной из операционных систем, в частности, операционных систем семейства Windows (cmd) и Linux (shell - bash).

Необходимо выполнить основные задания лабораторной работы, а также индивидуальное задание. Рассматриваются и изучаются стандартные операции копирования файлов, создания файлов, просмотр содержимого каталогов, удаление каталогов и файлов, выбор логических дисков, задание маски файлов, перенаправление вывода результатов команд в файл, использование относительного пути. Также приводится пример написания командного файла.

### 2.1 Изучение основных консольных команд Windows

Рассмотрим для начала работу в командной строке Windows (cmd), на примере операционной системы Windows 10.

Для того, чтобы запустить командную строку необходимо выбрать «Поиск», в качестве запускаемой программе указать «cmd» и запустить, нажав кнопку «Открыть» (Рисунок 2.1).

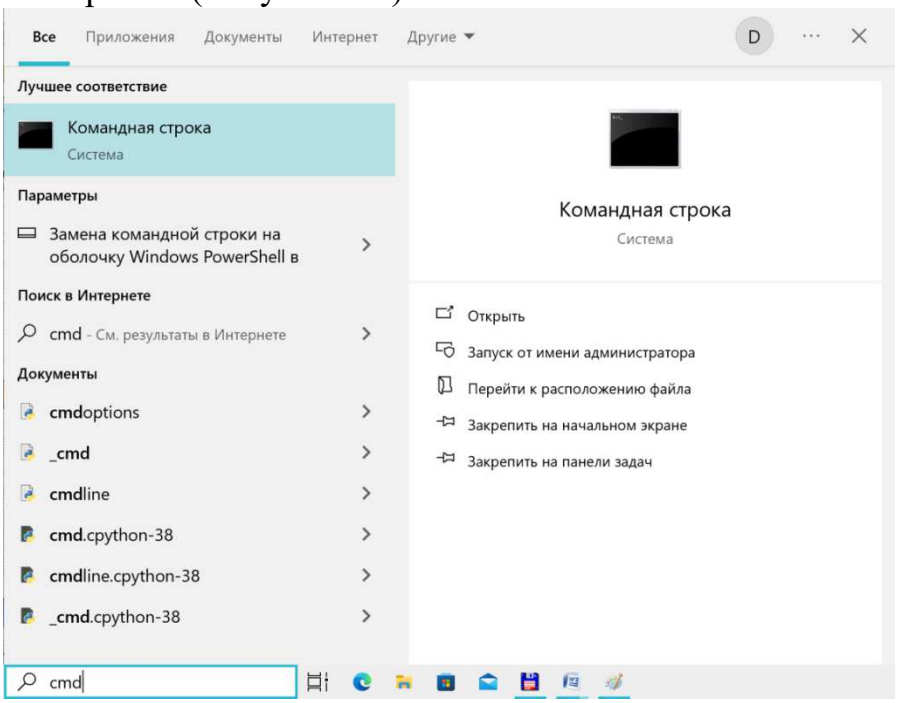



Рисунок 2.1 – Вызов командной строки

После запуска отобразится командная строка, где ввод команд осуществляется с клавиатуры (Рисунок 2.2). Подробное описание команд можно найти, пользуясь командой help, то есть необходимо ввести в строку «help» и нажать Enter. Более подробное описание конкретной команды можно прочитать, введя help и далее имя интересующей команды, например,

>help help

>help dir

Кроме того, часть описаний команд можно найти в **приложении Б** данного пособия.


Рисунок 2.2 – Окно командной строки

На представленном рисунке 2.2 виден черный экран, на котором указан путь к текущему каталогу. Первая буква и двоеточие «C:» - означают логический диск или том. Далее через слэш указано имя папки или директории (каталога) «Users» на диске C.

Рассмотрим общий формат команды в командной строке CMD. Формат команды CMD:

имя_команды [параметр]... [ключ]...

где имя_команды – зарезервированное ключевое слово; параметр – сведения об объектах, с которыми работает команда (спецификации файлов, каталогов и т.д.). Может быть несколько параметров; ключ – дополнительная информация для уточнения функций команды. В команде может быть несколько ключей. В квадратных скобочках указан необязательный параметр, который можно опустить. Пример команды показан на рисунке 2.3.

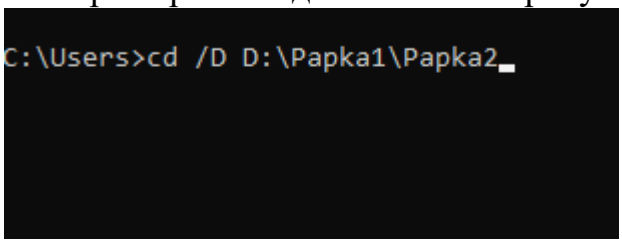



Рисунок 2.3 – Пример команды

Параметры отделяются хотя бы одним пробелом, каждый ключ начинается с символа «/» (slash). Допускается водить составные части команды большими и малыми латинскими буквами.

### 2.1.1 Задание на выполнение основных команд CMD

Выполнить все нижеследующие команды и задания, запоминая используемые команды и их назначение, после чего приступить к

индивидуальному заданию. При сдаче преподавателю уметь выполнять предложенные в командной строке команды, объяснять их назначение и команды в созданном командном файле. Оформить краткий отчет с учетом требований образовательного стандарта и представить в виде pdf.

**1. Проверить дату и время, команды DATE и TIME.**

Просмотреть сначала параметры команд с помощью

>help date

>help time

При вызове help <имя команды> указываются все ключи данной команды и режимы ее работы.

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), сменить каталог или диск.**

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.

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.

Изменение команды CHDIR при включении расширенной обработки команд: 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.

Описание ключей можно найти в приложении Б данного пособия. 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: или любом другом диске.** 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 будут 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, если данных много, и их не удаётся отобразить сразу, то часть из них может остаться скрытой от пользователя из-за ограничения количества буфера строк.

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.

Описание ключей можно найти в приложении Б данного пособия.

**7. Скопировать все файлы с диска (или какой-либо папки, содержащей файлы) C: с 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.

Описание ключей можно найти в приложении Б данного пособия. 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.** Скопировать файл в другой подкаталог.

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). Осуществить ввод данных с помощью EDIT.

**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 FILE2.TXT >> RESULT.TXT

добавит к файлу RESULT данные файла FILE2.

PROG < MYFILE.DAT

обеспечит ввод данных из файла в программу PROG.

Общий формат.

Команда [>|< |>>|<<] файл

Один знак > создает новый файл или переписывает старый, два 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 названия одного или нескольких файлов. Для удаления сразу нескольких файлов используются подстановочные знаки. Если указан каталог, из него будут удалены все файлы. Префикс "-" имеет значение НЕ

Описание ключей можно найти в приложении Б данного пособия.

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

Описание ключей можно найти в приложении Б данного пособия.

**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 cmd, а также специальные конструкции присущие языкам программирования, что позволяет создавать некие системные программы, выполняющие определенную последовательность действий, упрощая работу пользователя. Все нижеперечисленные 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. Изучить команды работы с сетью и сетевыми интерфейсами.**

В окне командной строки выполните команду ipconfig. Скриншот окна (Alt+PrintScr) поместите в отчет (Ctrl+V). Запишите в отчет информацию об IP адресе сетевого адаптера, маске сети и шлюзе по умолчанию.

Для получения более подробной информации о настройках адаптера запустите в окне командной строки утилиту ipconfig с ключом /all. Скриншот окна поместите в отчет.

Повторите команду ipconfig /all с выводом в текстовый файл и запишите в отчет информацию о физическом адресе сетевой платы.

Применив команду ping, проверьте доступность основного шлюза и доступность удаленного узла. Адреса удаленных узлов выбирайте по своему варианту (см. Варианты). Скриншоты и выводы поместите в отчет.

Используя опцию -i команды ping определите адреса первых трех маршрутизаторов, находящихся между вашим компьютером и удаленным узлом.

Применив команду tracert, получите список роутеров на маршруте от вашего компьютера до удаленного узла. Команда, как и ping, использует протокол ICMP, только вначале задается время жизни 1, что приводит к возвращению ошибки в виде дейтаграммы ICMP время жизни истекло, затем время жизни увеличивается до 2 и так далее.

Используя pathping, изучите состояние линков на маршруте от вашего компьютера до удаленного узла и определите самые «узкие места» (т.е. самые медленные участки).

Получите таблицу ARP вашего компьютера с помощью утилиты arp, arp -a. Выпишите в отчет MAC адрес основного шлюза.

Командой netstat, выполненной с ключами –a, –n и –o, получите список соединений, установленных на вашем компьютере.

Определите имя любого приложения, установившей соединение с удаленной программой. Свой вывод обоснуйте, используя скриншот.

### 2.1.2 Установка Python и запуск примера

Для выполнения индивидуального задания необходимо воспользоваться языком программирования Python.

Для его установки необходимо перейти на официальный сайт и скачать установщик [4] (Рисунок 2.4)

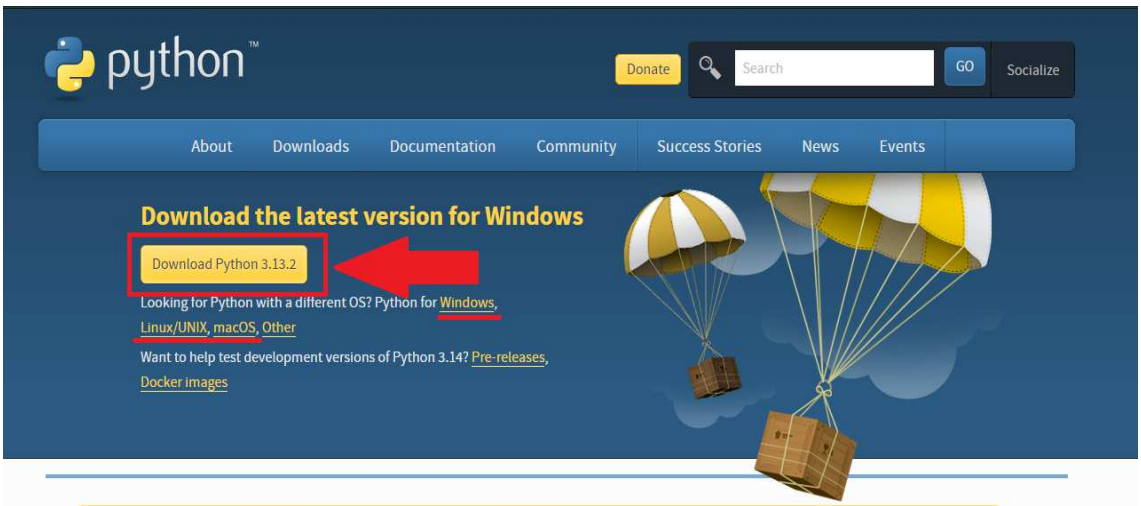



Рисунок 2.4 – Официальный сайт Python

После скачивания установщика, необходимо его открыть (Рисунок 2.5). Если необходимо изменить какие-то настройки перед установкой, необходимо нажать кнопку «Customize installation». Для установки нужно нажать на кнопку «Install Now».


Рисунок 2.5 – Установщик Python

После установки (Рисунок 2.6) можно переходить к выполнению индивидуального задания.

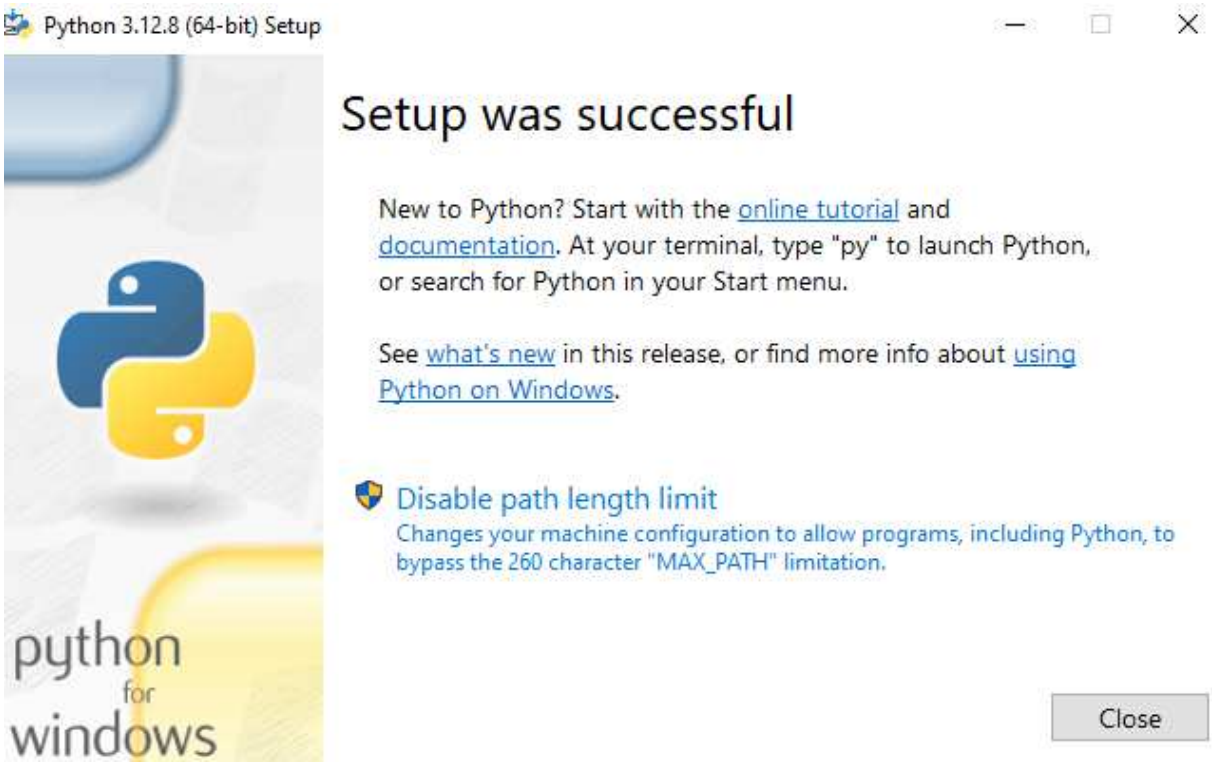



Рисунок 2.6 – Успешная установка Python

В Python входные аргументы (аргументы командной строки) передаются в скрипт через модуль sys или более удобный модуль argparse. Модуль sys предоставляет список sys.argv, который содержит все аргументы, переданные скрипту при запуске. Первый элемент (sys.argv[0]) — это имя скрипта, а остальные элементы — аргументы.

Запустим следующий пример:

1. Создайте файл «script.py» через IDLE (Python) либо через переименование созданного текстового файла.
2. Скопируйте код примера в данный файл и сохраните его.
3. Откройте командную строку и перейдите в ней в каталог, где храниться файл «script.py».
4. Используйте команду «Python script.py Николай 20» для запуска.

**Пример кода на языке Python:**

```
import sys
def main():
        if len(sys.argv) != 4:
                sys.exit(1)
        p1 = float(sys.argv[1])
        p2 = float(sys.argv[2])
        p3 = float(sys.argv[3])
        result = (p1 + p2) * p3
        print(result)
if __name__ == "__main__":
        main()
```

### 2.1.3 Индивидуальные задания по консоли CMD

Сделать задание в соответствии со своим вариантом. Необходимо реализовать по вариантам задачи на Python, используя команды cmd.

Если необходимо произвести расчёты с плавающей запятой, лучше всего использовать Python. В этом случае в программу можно передавать аргументы командной строки через stdin и stdout. Скрипт можно вызвать из cmd файла, а ввод осуществлять с помощью cmd. Непосредственно расчёт следует выполнять в скрипте Python.

Если используются целые значения, то реализовать отдельно скрипт Python и файл cmd.

Для запуска программы на языке Python через консоль необходимо в командной строке перейти в папку, в которой расположен файл с кодом и вызвать его следующей командой:

*python название_файла.py 3 1 8 > res.txt*

С помощью python вызывается интерпретатор языка. Происходит запуск скрипта название_файла.py. sys.argv[] – это список аргументов командной строки. sys.argv[0] – название файла. Значения 3, 1 и 8 будут записаны в sys.argv[1], sys.argv[2] и sys.argv[3] соответственно. Символ «>» это оператор перенаправления. В результате его выполнения значения будут записаны в файл res.txt. Если такого файла не существует, то он будет создан автоматически.

Для ознакомления с Python можно обратиться к параграфу 4.9 пособия Суханов, А. Я. Разработка веб-сервисов для научных и прикладных задач [5] или [6, 7].

**Написать cmd файл и скрипт Python:**

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. Ввести параметры квадратичного уравнения и записать корни уравнения в файл.
5. Ввести два числа и минимальное записать в файл.
6. Ввести две координаты в двумерном пространстве и по ним построить уравнение прямой, коэффициенты прямой вывести в файл.
7. Ввести два трехмерных вектора, найти их скалярное произведение и вывести в файл.
8. Пронормировать трехмерный вектор, результат вывести в файл.
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 их сумму. Длину вектора ввести с клавиатуры.

13. По двум сторонам треугольника и углу между ними найти третью сторону 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.

### 2.1.4 Полезные примеры для создания cmd файла

Вот несколько полезных примеров для создания `.cmd` файлов (Windows командные сценарии). Эти скрипты позволяют автоматизировать задачи, запускать программы, копировать файлы, управлять системными настройками и многое другое.

**1. Пример: Простой запуск программы**

```
@echo off
echo Запуск блокнота
start notepad.exe
pause
```

Описание: Открывает блокнот и ожидает нажатия клавиши (`pause`).

**2. Пример: Копирование файла**

```
@echo off
echo Копирование файла
copy "C:\Исходный\файл.txt" "D:\Резервные копии\файл_копия.txt"
pause
```

Описание: Копирует указанный файл в другую папку.

**3. Пример: Множество команд подряд**

```
@echo off
echo Создаем папку
mkdir "C:\МоиПроекты\Проект1"
echo Папка создана
pause
```

**4. Пример: Проверка наличия файла**

```
@echo off
if exist "C:\Файлы\важный_файл.txt" (
    echo Файл найден!
) else (
    echo Файл не найден.
)
pause
```

**5. Пример: Запуск программы с аргументами**

```
@echo off
echo Запуск программы с аргументами
start "" "C:\Program Files\SomeApp\app.exe" /arg1 /arg2
pause
```

**6. Пример: Цикл for для обработки файлов**

```
@echo off
for %%F in (*.txt) do (
   echo Обработка файла %%F
   rem добавить сюда команды, например, архивировать или копировать
)
pause
```

**7. Пример: Загрузка нескольких команд через файл**

Создайте файл `скрипт.cmd` с содержимым:

```
@echo off
echo Начинаем автоматическую задачу
rem - вставьте ваши команды
pause
```

Запускаете его двойным кликом или через командную строку.

**8. Пример: Задержка выполнения**

```
@echo off
echo Ждем 5 секунд
timeout /t 5
echo Продолжаем
pause
```

**9. Пример: Вызов другого `.cmd` файла**

```
@echo off
call "C:\МоиСкрипты\другой_скрипт.cmd"
pause
```

Итог

- `@echo off` — отключает отображение команд в выводе
- `pause` — ждет нажатия клавиши, чтобы пользователь мог увидеть результат
- `start` — запускает программу
- `if`, `for` — управляющие конструкции
- `rem` или `::` — комментарии

## 2.2 Изучение терминала Linux (shell — bash)

Задание — Выполнить аналогичные команды и задания для данного терминала как для cmd.

Shell — это оболочка операционной системы для взаимодействия

пользователя с системой, она может быть в виде командной строки или графического интерфейса. Bash (Bourne Again Shell) — это одна из самых популярных реализаций командной оболочки (shell) для Unix-подобных операционных систем, таких как Linux и macOS. Это улучшенная версия оригинального sh (Bourne Shell), созданная в рамках проекта GNU.

Данная оболочка позволяет вводить команды в интерактивном режиме, запускать написанные скрипты, содержание циклы, условия и функции, поддерживает переменные.

Пример:

```
name="User"
echo "Hello, $name!"
```

Так же, как и в CMD, можно перенаправлять результаты команды не на экран, а в файл:

```
ls > file.txt  # Запись вывода команды ls в файл
```

Можно запускать процессы, в том числе в фоновом режиме:

```
# Запуск процесса в фоновом режиме, приостановка на 10 секунд
sleep 10 &
```

Так же можно передавать вывод одной команды на вход другой:

```
ls | grep "txt" # Поиск файлов с расширением .txt
```

Здесь в результате запуска команды ls получается список файлов текущего каталога. Команда grep используется для поиска текста. В данном случае она ищет строки, содержащие подстроку "txt", в выводе команды ls. grep позволяет искать даже по содержимому файлов, используя сложные шаблоны для поиска.

Ниже приведены несколько полезных примеров Bash-скриптов, которые помогут автоматизировать задачи и управлять системой.

**1. Простой скрипт: выводит текущую дату и каталог**

```
#!/bin/bash
echo "Текущая дата и время: $(date)"
echo "Текущий каталог: $(pwd)"
```

**2. Создание резервной копии папки**

```
#!/bin/bash
SOURCE_DIR="$HOME/Документы"
BACKUP_DIR="$HOME/Бэкапы"
DATE=$(date +%Y-%m-%d)
tar -czf "$BACKUP_DIR/backup-$DATE.tar.gz" "$SOURCE_DIR"
echo "Резервная копия создана: $BACKUP_DIR/backup-$DATE.tar.gz"
```

**3. Проверка наличия файла и его обработка**

```
#!/bin/bash
FILE="$HOME/important.txt"
if [ -f "$FILE" ]; then
   echo "Файл существует."
   # Можно добавить команды для обработки файла
```

```
    cat "$FILE"
else
    echo "Файл не найден."
fi
```

**4. Цикл для обработки списка файлов**

```
#!/bin/bash
for filename in "$HOME/Downloads"/*.jpg; do
    echo "Обработка файла: $filename"
    # Например, сжать изображение или переименовать
    # mv "$filename" "${filename%.jpg}.jpeg"
done
```

**5. Мониторинг использования диска и отправка уведомления**

```
#!/bin/bash
USAGE=$(df / | grep / | awk '{ print $5 }' | sed 's/%//')
THRESHOLD=80
if [ "$USAGE" -ge "$THRESHOLD" ]; then
    echo "Использование диска превышает $THRESHOLD% ($USAGE%)"
    # например, отправить email или выполнить другие действия
fi
```

**6. Автоматизация запуска нескольких команд с задержками**

```
#!/bin/bash
echo "Начинаем выполнение задач..."
sleep 2
echo "Задача 1 завершена."
sleep 3
echo "Задача 2 завершена."
```

**7. Взаимодействие с пользователем (подсказка и ввод)**

```
#!/bin/bash
read -p "Введите ваше имя: " NAME
echo "Привет, $NAME!"
```

**8. Обновление системы и автоматическая перезагрузка**

```
#!/bin/bash
sudo apt update && sudo apt upgrade -y
echo "Обновление завершено. Перезагрузка системы..."
sudo reboot
```

**9. Скрипт запуска нескольких команд с обработкой ошибок**

```
#!/bin/bash
command1
if [ $? -ne 0 ]; then
    echo "Ошибка выполнения command1"
    exit 1
fi
command2
```

```
if [ $? -ne 0 ]; then
    echo "Ошибка выполнения command2"
    exit 1
fi
echo "Все команды выполнены успешно."
```

**Общие рекомендации**

– Сделайте файл исполняемым:

*chmod +x ваш_скрипт.sh*

– Запускайте так:

*./ваш_скрипт.sh*

или:

*bash ваш_скрипт.sh*

## 3. Лабораторная работа 3. Изучение электронных таблиц Calc

Изучить основные возможности электронных таблиц по обработке данных, включая расчеты по данным в ячейках, оформление таблиц, использование закрепления ячеек, условного оператора в расчетах, построения графиков, использование агрегированных функций и расчетов по условию, сводных таблиц, итоговых таблиц и фильтрации.

### 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 3.1 приведено главное окно Calc.

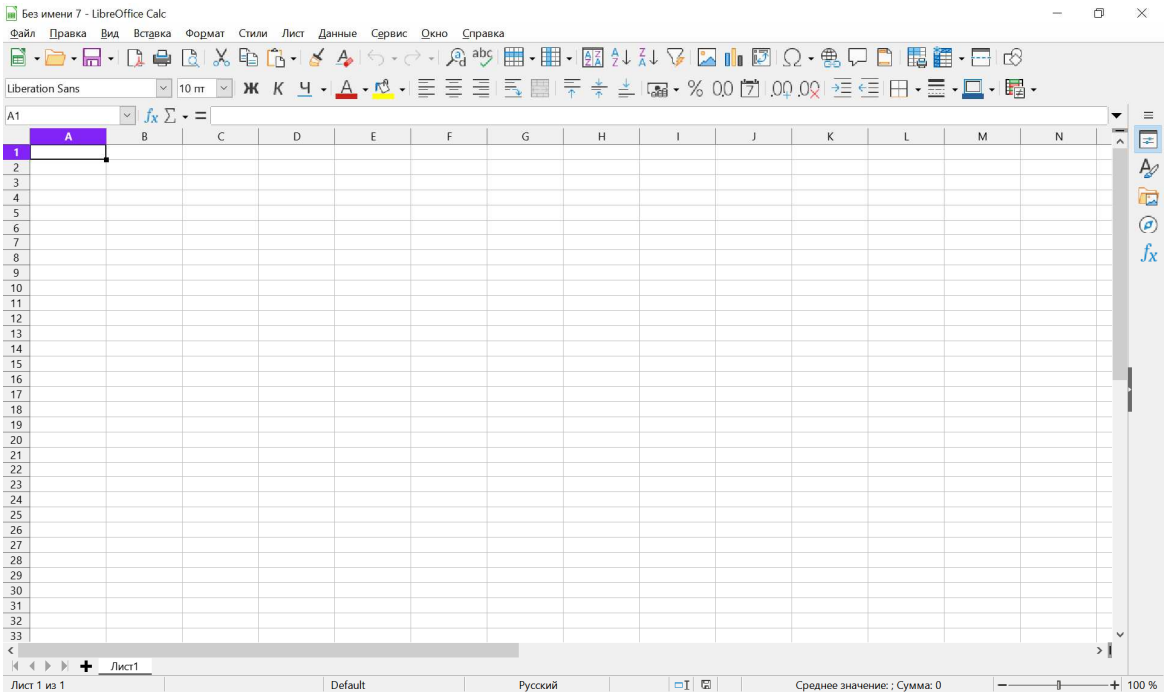



Рисунок 3.1 - Главное рабочее окно LibreOffice Calc

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 неизменными, 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.

## 3.3 Формулы и ячейки Calc

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 оставляет 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(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(E2:P7)*
– Функция: *=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(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:

- Щелкните 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 и 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.

## 3.4 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 ту же 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.

### 3.5 Выполнение агрегированных операций и операций фильтрации

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). Также 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.

### 3.6 Задание 1.

На листе 1 создать удобочитаемую таблицу (таблицы) в соответствие с 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. 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**.

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.

### 3.7 Задание 2. График функции.

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.

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 равное - 2, затем ниже ввести

формулу =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.

Таблица 3.1 — Варианты для задания 2

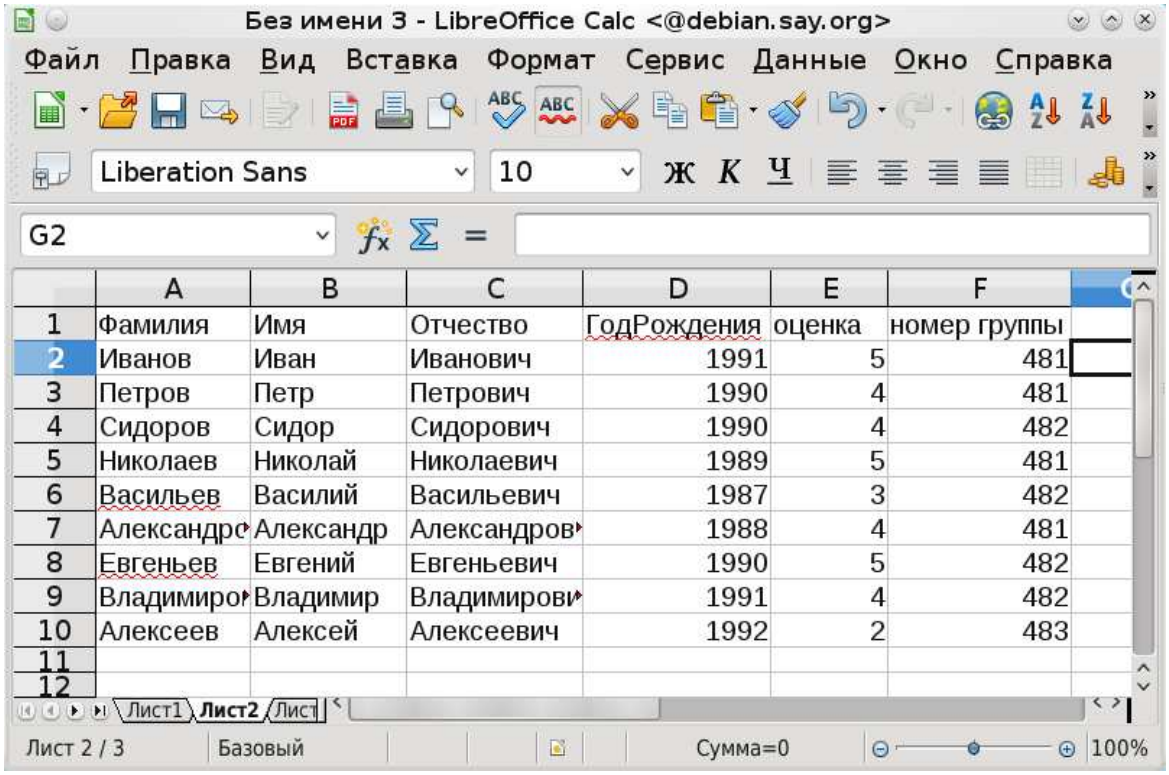
| | |
|---|---|
| 1) $y=\begin{cases}\dfrac{1+x^2}{\sqrt{1+x^4}}, x\le 0\\ 2x+\dfrac{\sin^2(x)}{3+x}, x>0\end{cases}$ | 2) $y=\begin{cases}\dfrac{3+\sin^2(2x)}{1+\cos^2(x)}, x\le 0\\ 2x+\dfrac{\sin^2(x)}{3+x}, x>0\end{cases}$ |
| 3) $y=\begin{cases}\dfrac{3+\sin^2(x)}{1+x^4}, x\le 0\\ 2x^2\cos^2(x),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

# 4 Лабораторная работа 4. Использование Calc как базы данных, изучение макросов

Цель работы. Освоить фильтрацию данных, группировку данных, подведение итоговых результатов. 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

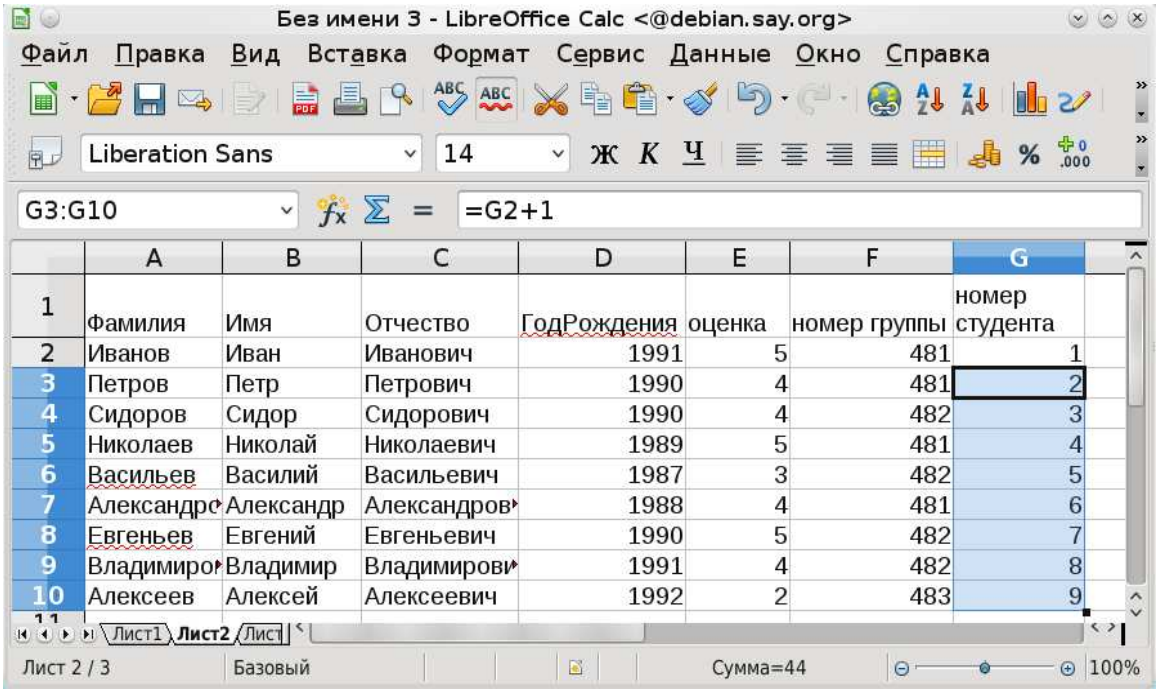
Дана таблица с шапкой как в примере, представленном на рисунке 4.1, необходимо дополнить ее до 15-20 записей:




| | 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.




| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Фамилия | Имя | Отчество | ГодРождения | оценка | номер группы | 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 4.2 - 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.

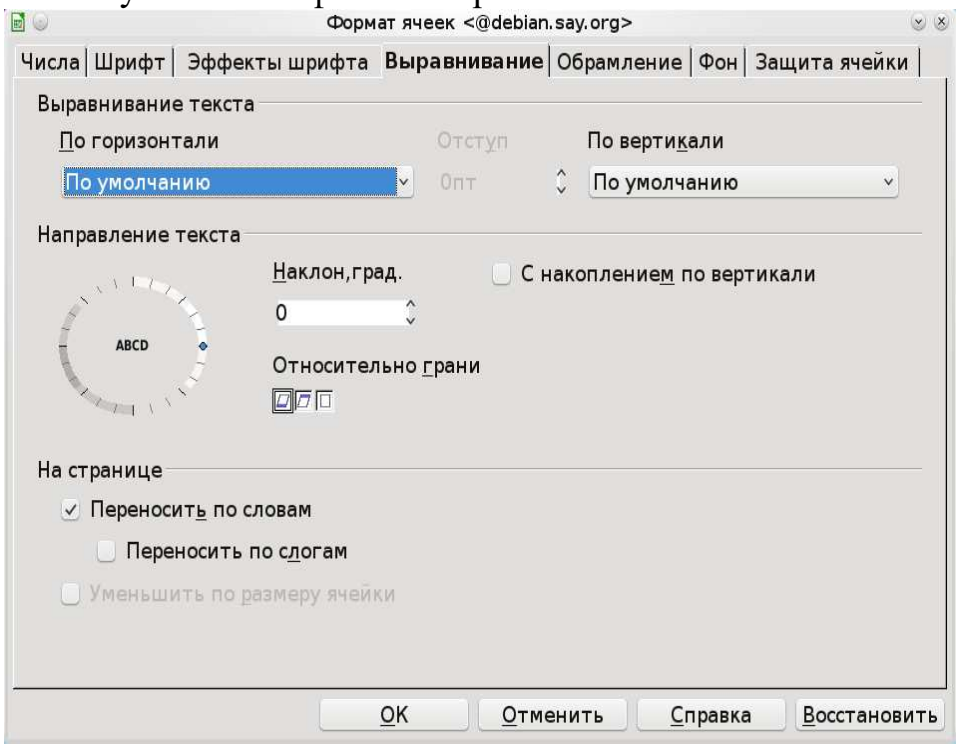



Рисунок 4.3 - Форматирование ячейки

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».

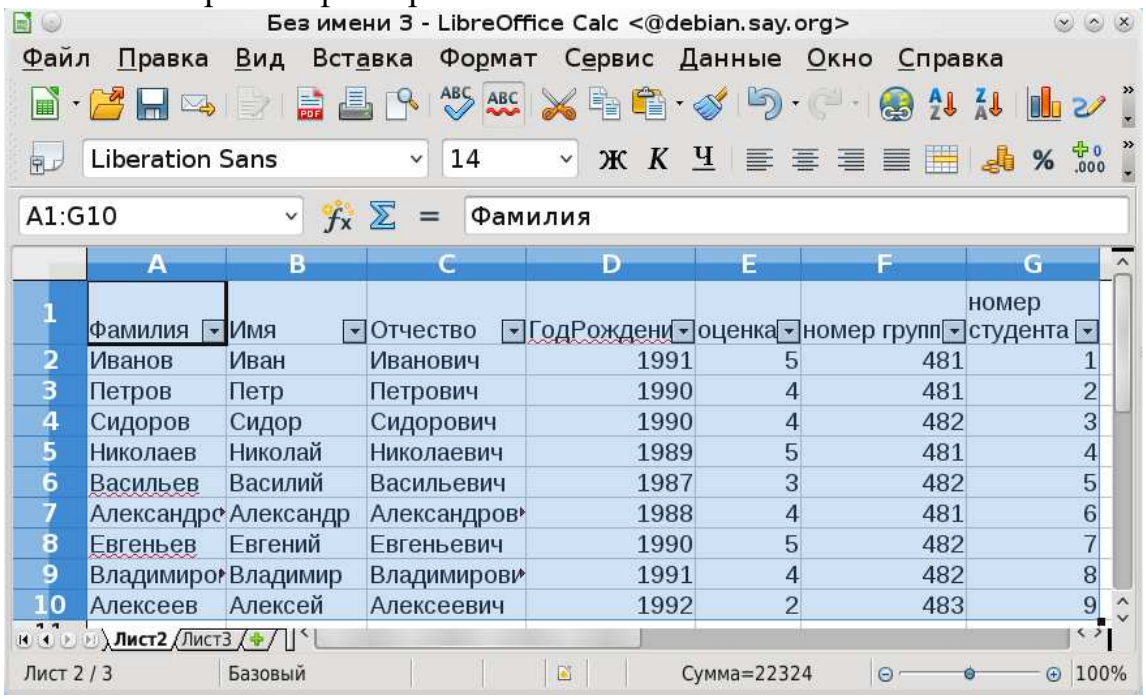



Рисунок 4.4 - Автофильтр

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.

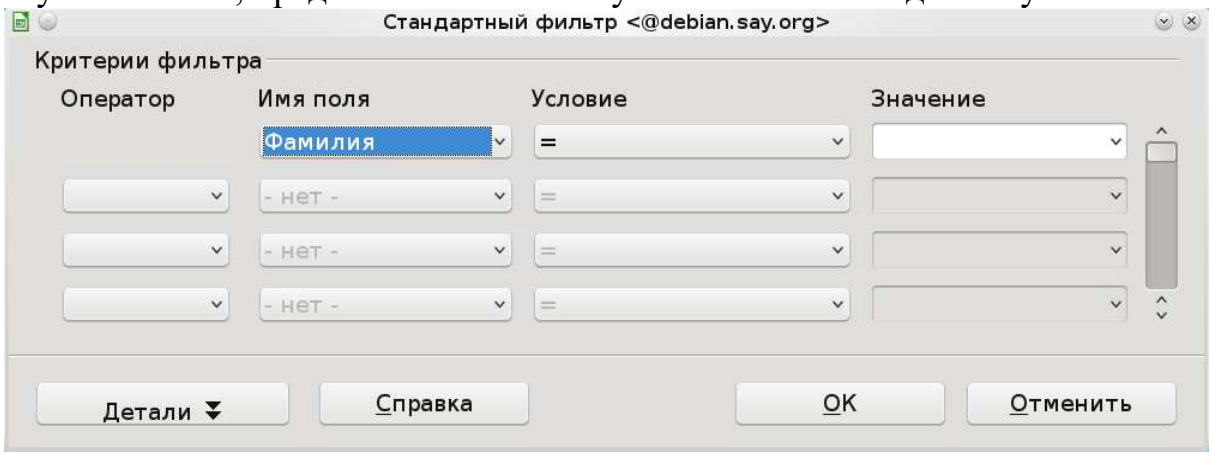



Рисунок 4.5 - Условие на поле таблицы

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.

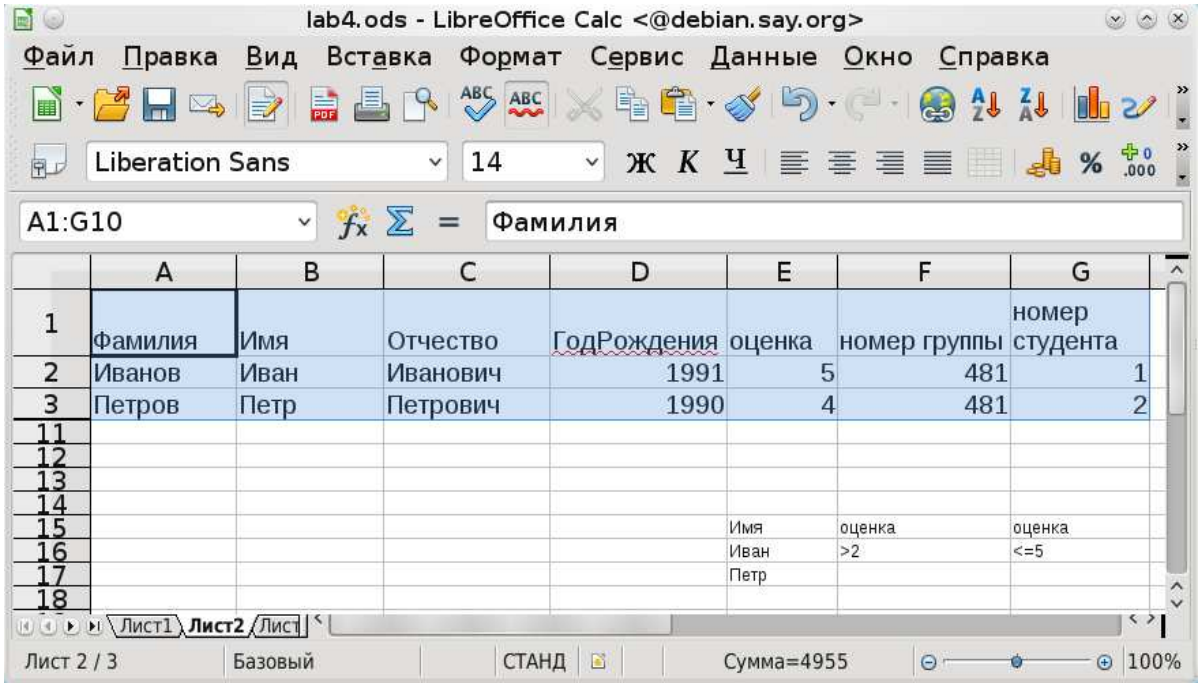



Рисунок 4.6 - Использование расширенного фильтра

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 4.1 - Рабочая таблица о продажах телефонов

| Дата | Магазин | Марка | Серия | Продажа (штуки) | Цена (руб) | 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 таблицу». 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.

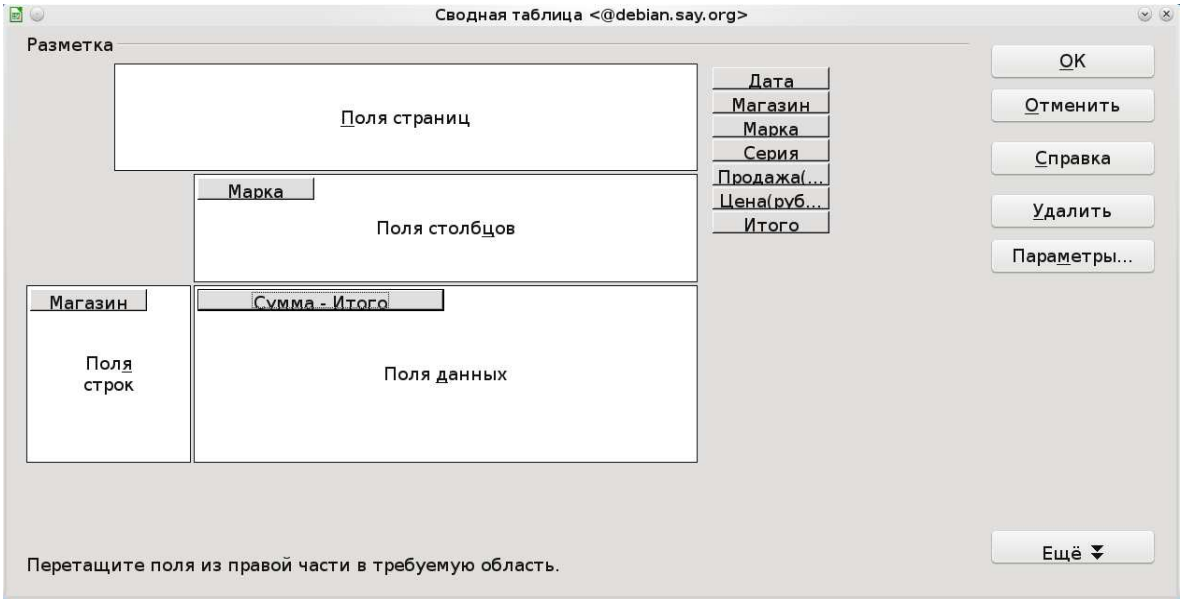



Рисунок 4.7 - Установка параметров сводной таблицы

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

Таблица 4.2 - 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).

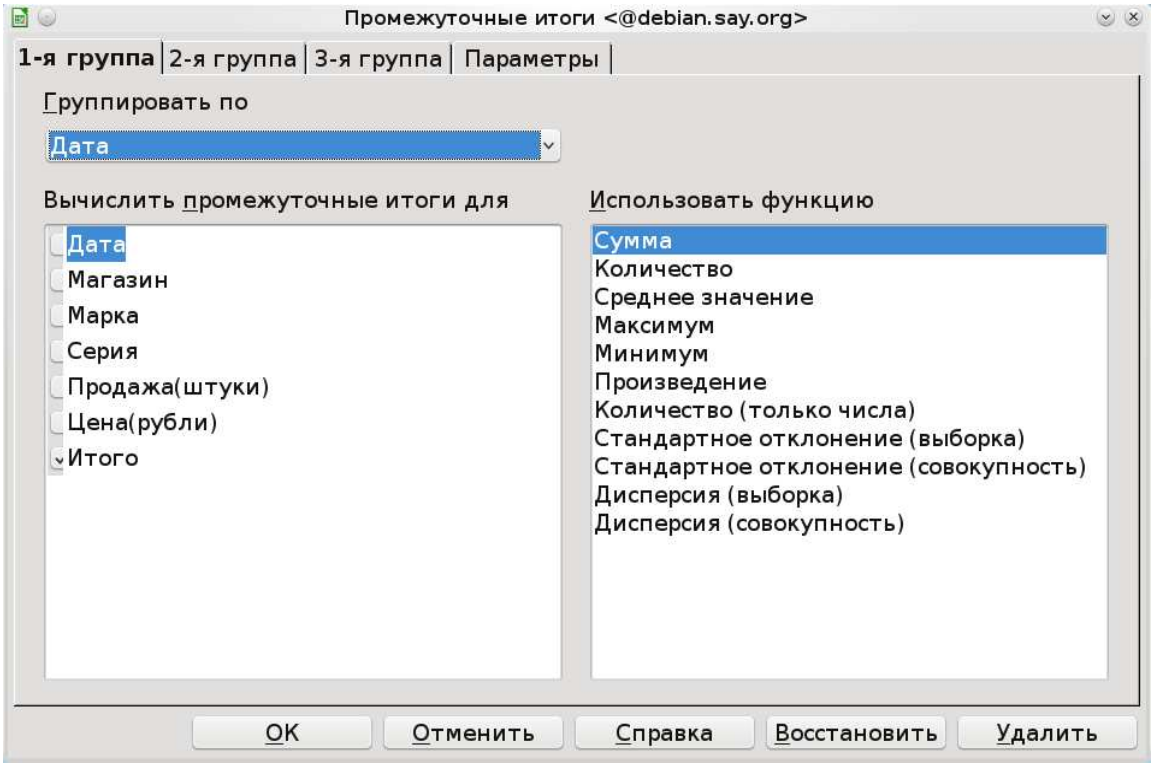



Рисунок 4.8 - Группировка и итоги

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 4.3 - Результирующая таблица

| Дата | Магазин | Марка | Серия | Продажа (штуки) | Цена (руб) | 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 группировки по серии.

Таблица 4.4 - 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.

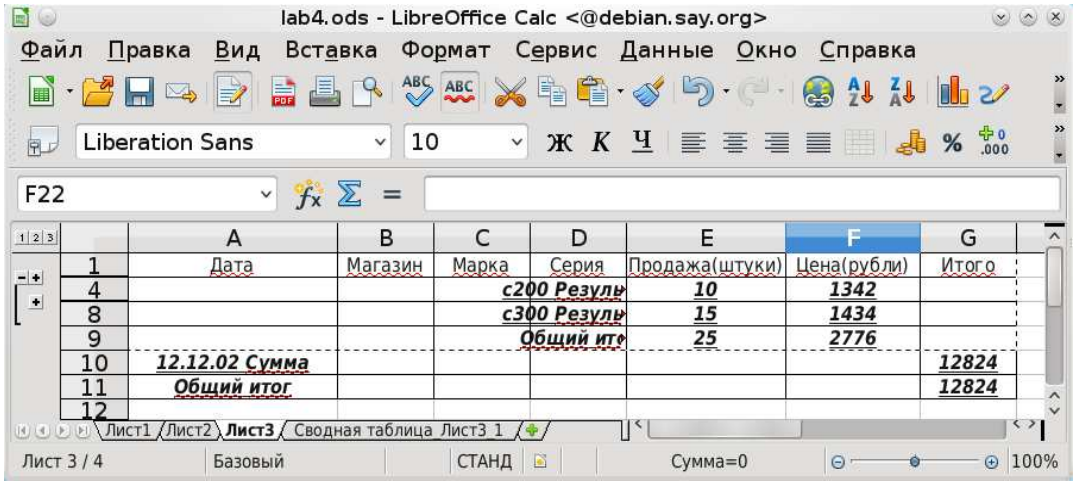



Рисунок 4.9 - Пример итоговой таблицы без исходных данных

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 с200.

Таблица 4.5 - Группировка по двум полям

| Дата | Магазин | Марка | Серия | Продажа(штуки) | Цена(рубли) | Итого |
|---|---|---|---|---|---|---|
| 12.12.02 | 1 | Нокия | с200 | 4 | 1221 | 4884 |
| 12.12.02 | 2 | Нокия | с200 | 6 | 121 | 726 |
| | | | ***с200 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 по Calc

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

## 5 Лабораторная работа 5. Изучение макросов Calc Basic

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

Для дополнительной помощи по функциям макросов можно обращаться по ссылке [8]

### 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.

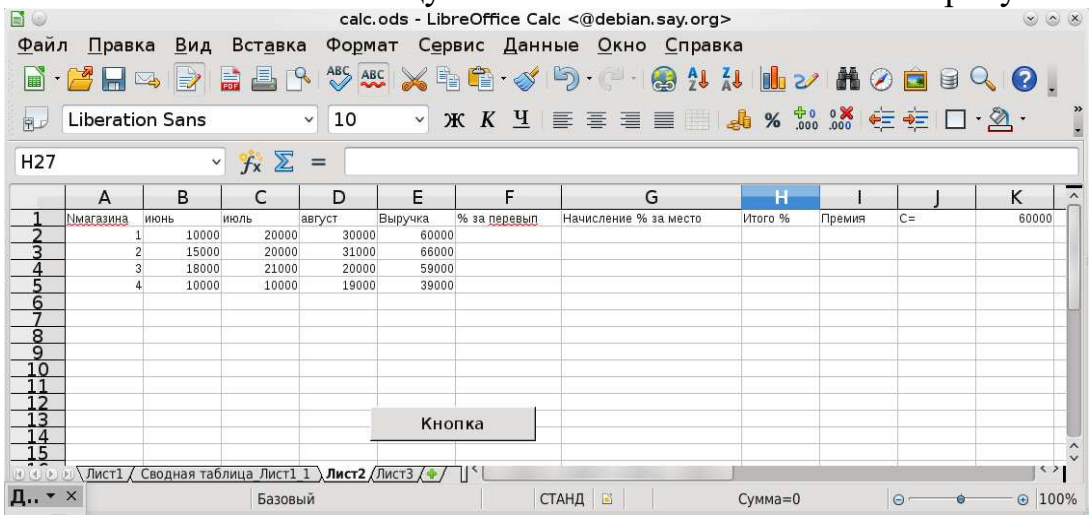



Рисунок 5.1 - Расчетная таблица по магазинам

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 j
Next I
`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
    Endif
    If Sheet.getCellByPosition(4, i).Value = box(1) Then
        Sheet.getCellByPosition(6, i).Value=2
    Endif
    If Sheet.getCellByPosition(4, i).Value = box(2) Then
        Sheet.getCellByPosition(6, i).Value=1
    Endif
    If Sheet.getCellByPosition(4, i).Value = box(3) Then
        Sheet.getCellByPosition(6, i).Value=0
    Endif
Next i
```

То же 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 i

      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 i

      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.

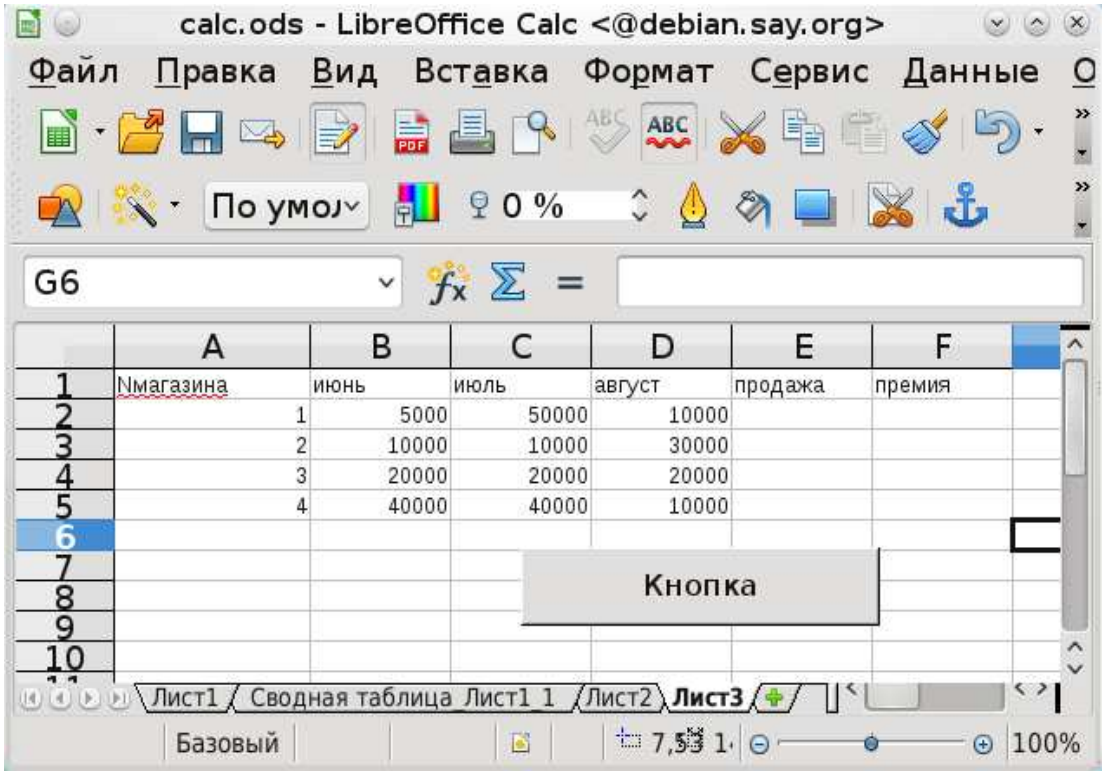



Рисунок 5.2 - Исходная таблица с данными о магазинах

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 посередине матрицы и строки посередине. 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 Лабораторная работа 6. Изучение макросов OOO Basic Libre Office Writer

Целью работы является изучение основных возможностей работы с макросами и документами.

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. Для дополнительной помощи по функциям можно обращаться по ссылке [6]

## 6.1 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.

## 6.2 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.

## 6.3 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*

## 6.4 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)*

## 6.5 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 6.1). 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 6.2).

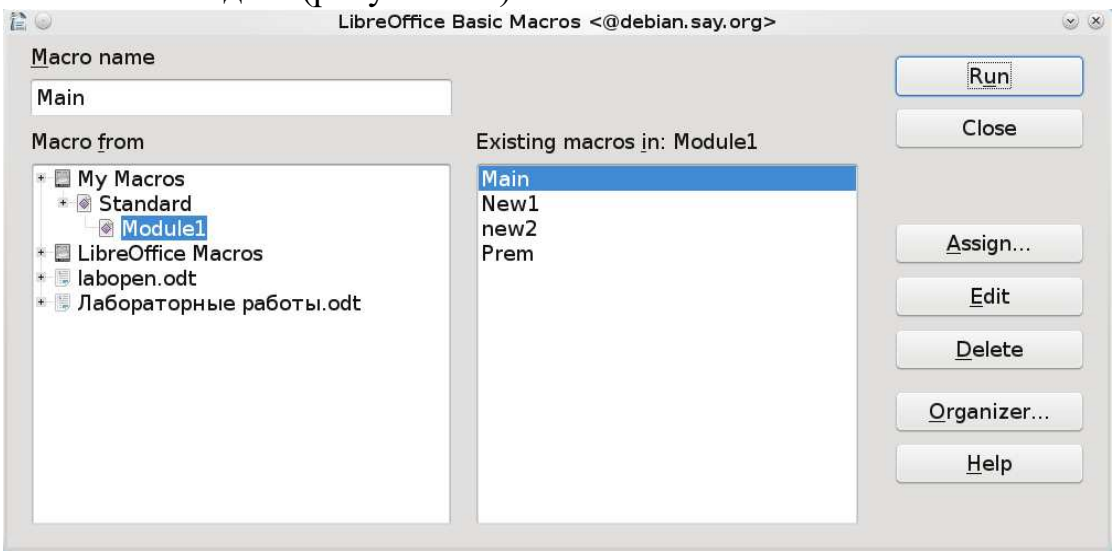



Рисунок 6.1 - Окно создания и редактирования макросов

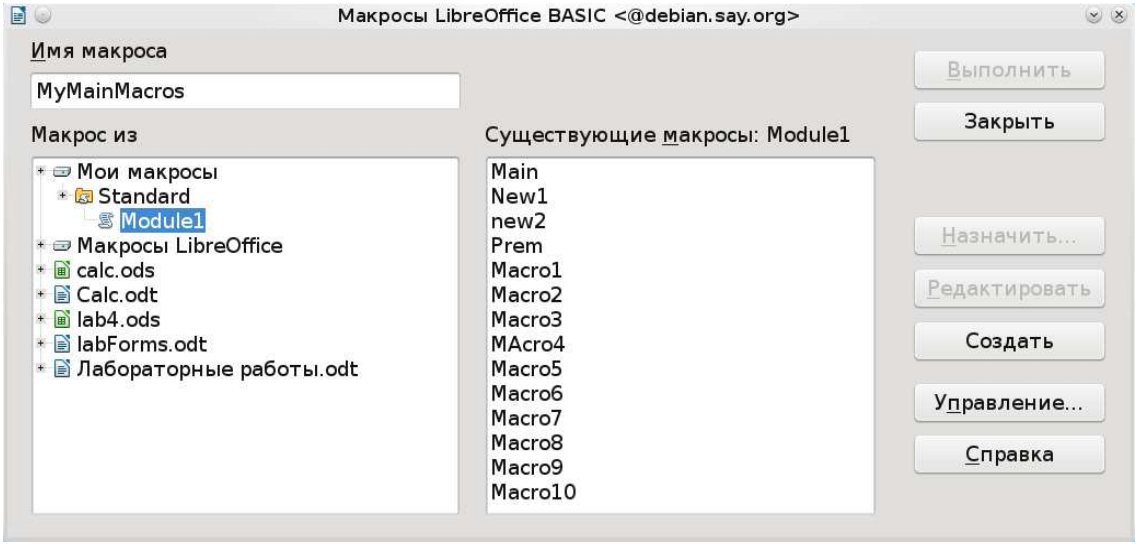



Рисунок 6.2 - Пример создания нового макроса MyMainMacros

В результате создания и редактирования макроса появляется окно редактора Бэйсика, на рисунке 6.3 приведен пример с двумя макросами, естественно их может быть больше и они могут иметь входные параметры.

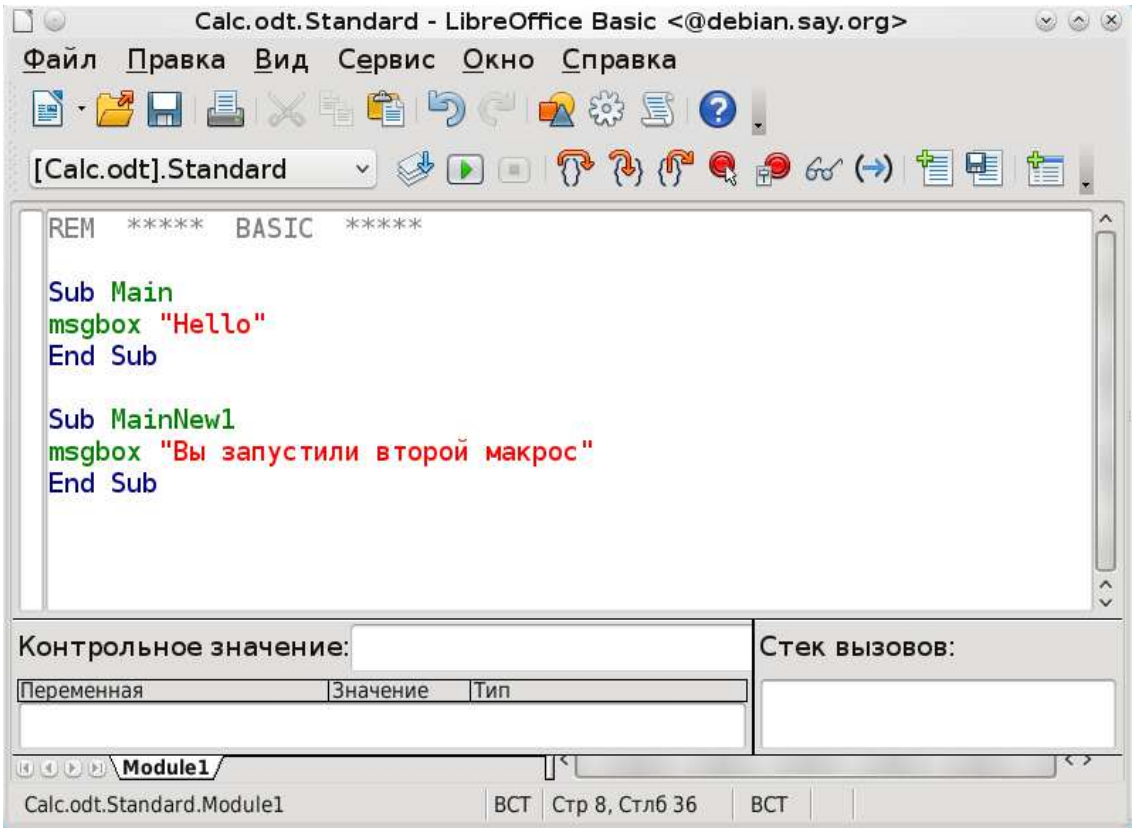



Рисунок 6.3 - Редактор Basic и два макроса

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
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.

## 6.6 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.

### 6.7 Индивидуальные задания по вариантам. 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 посередине 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.

# 7 Лабораторная работа 7. Изучение Форм и визуальных элементов управления в LibreOffice

Для дополнительной помощи при выполнении лабораторной можно обращаться по ссылке [6].

## 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 «ОК» и значка-иконки «Сведения».

Для создания 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*

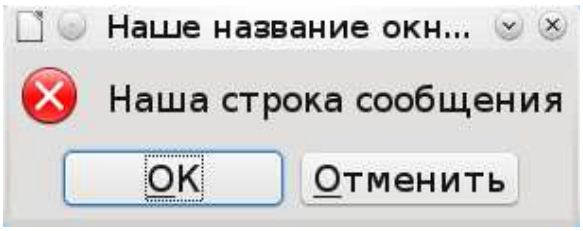



Рисунок 7.1 - пример Диалогового окна для обработки ошибок

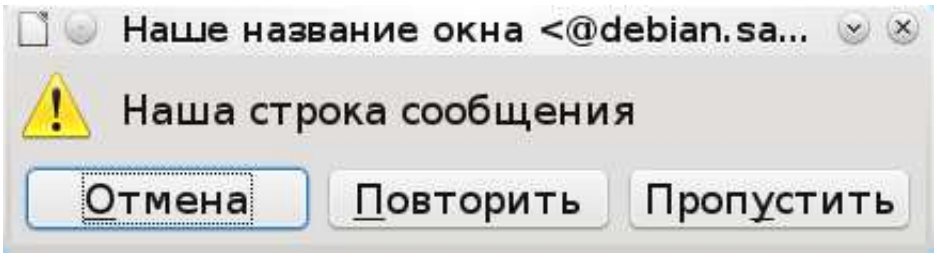



Рисунок 7.2 - Пример окна с восклицательным знаком

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 OK — 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.

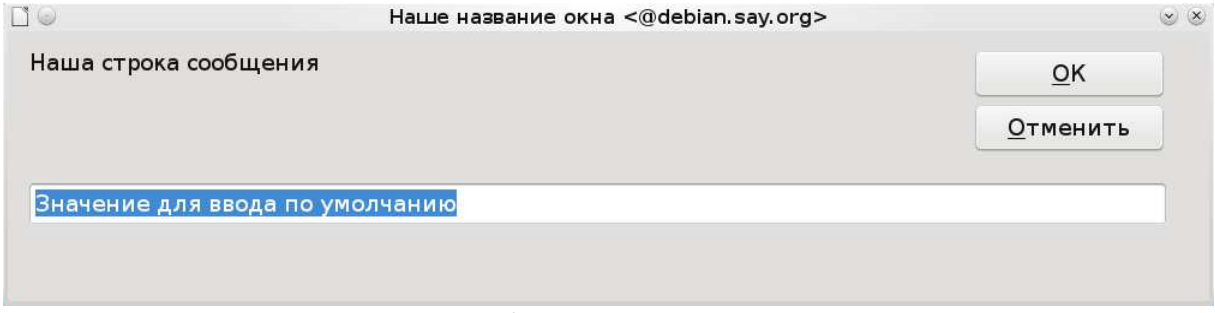



Рисунок 7.3 - Ввод текста с помощью диалога

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).

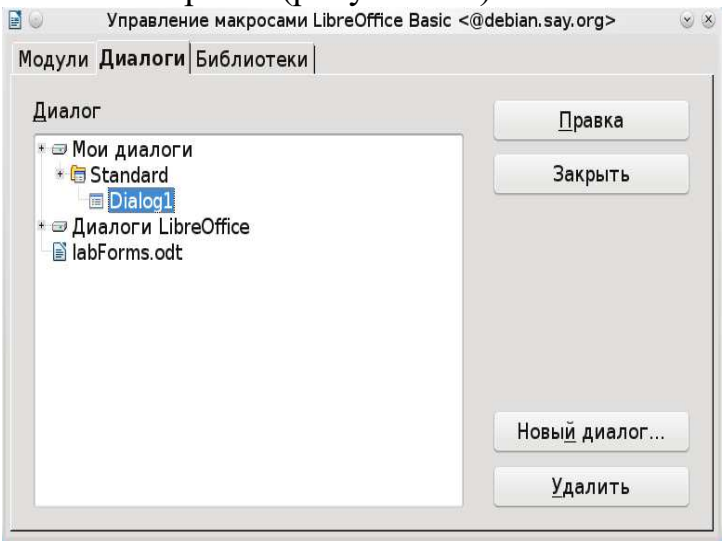



Рисунок 7.4 - Создание Диалога

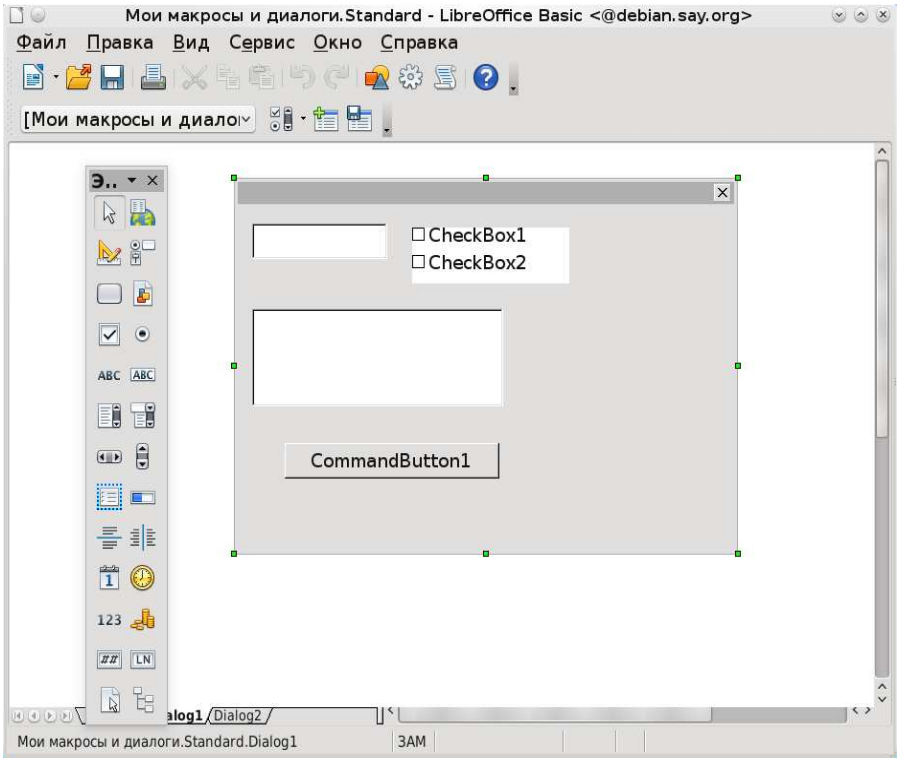



Рисунок 7.5 - Редактирование окна диалога, установка элементов управления

На рисунке 7.5 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 представлеяет 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).

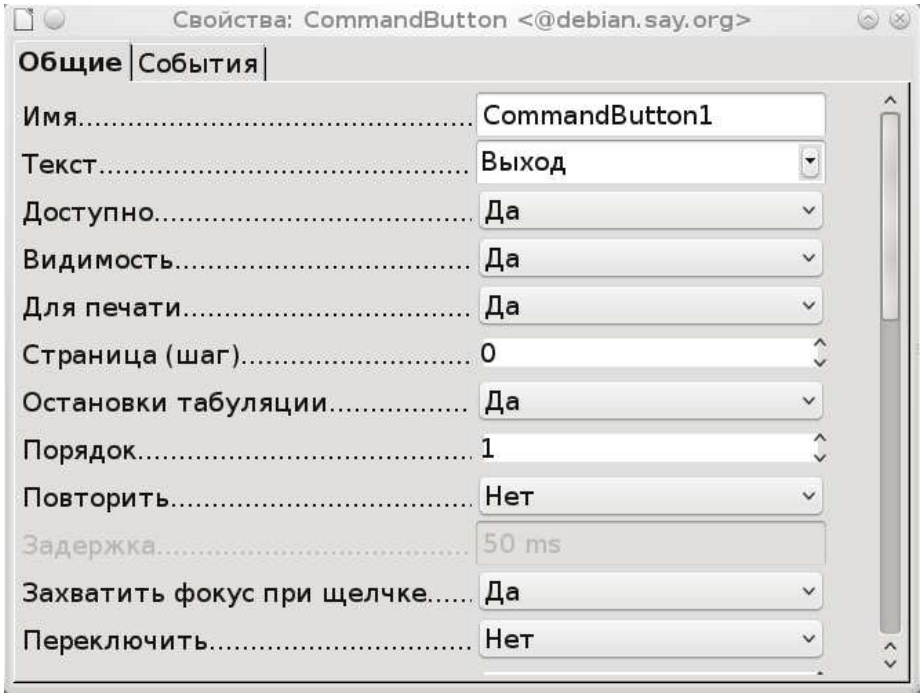


Рисунок 7.6 - Свойства объекта кнопка

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.

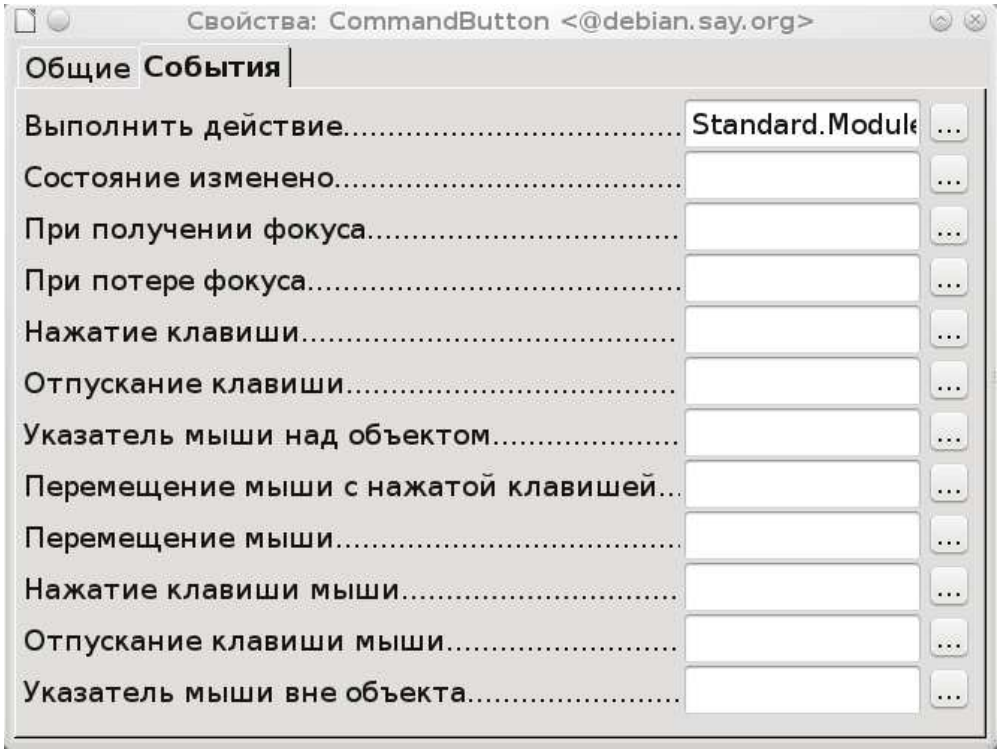



Рисунок 7.7 - События объекта кнопка

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

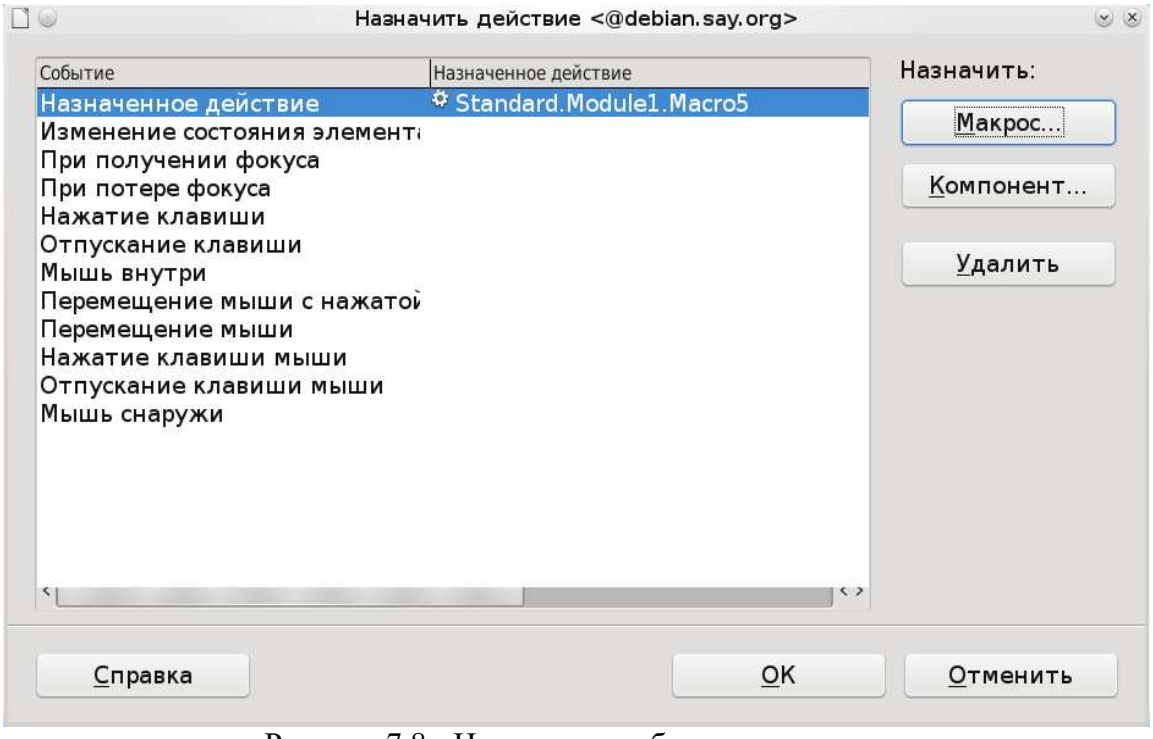



Рисунок 7.8 - Назначение события для кнопки

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 следующиесвойства 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 левоговнутреннего 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 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.

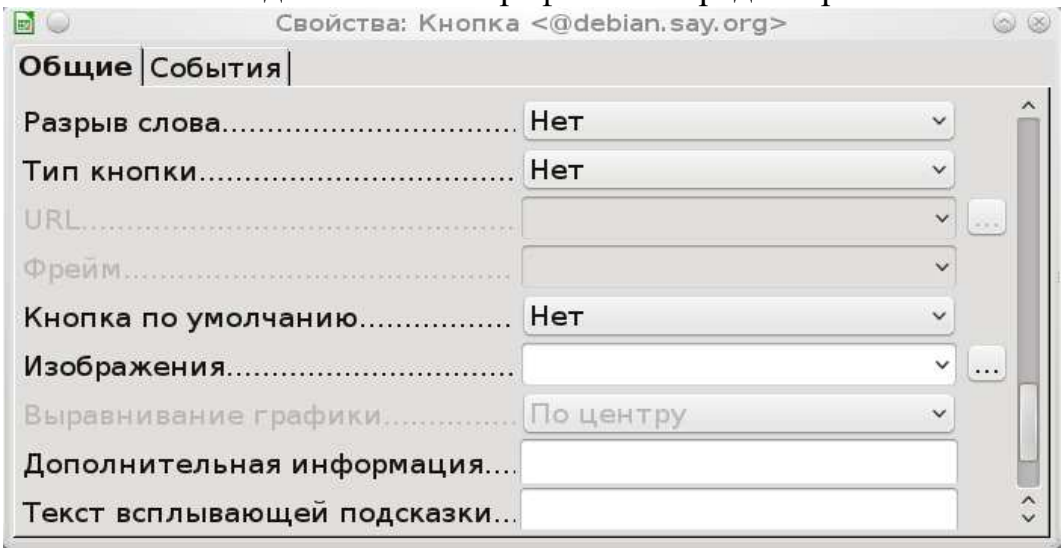



Рисунок 7.9 - Установка изображения на кнопке

## 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.

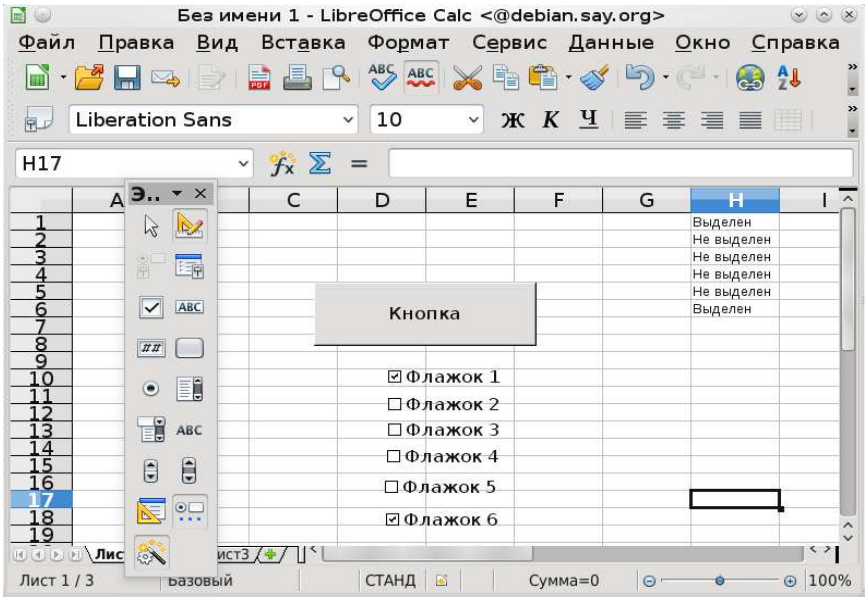



Рисунок 7.10 - Флажки и работа с Calc, обработка событий

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.

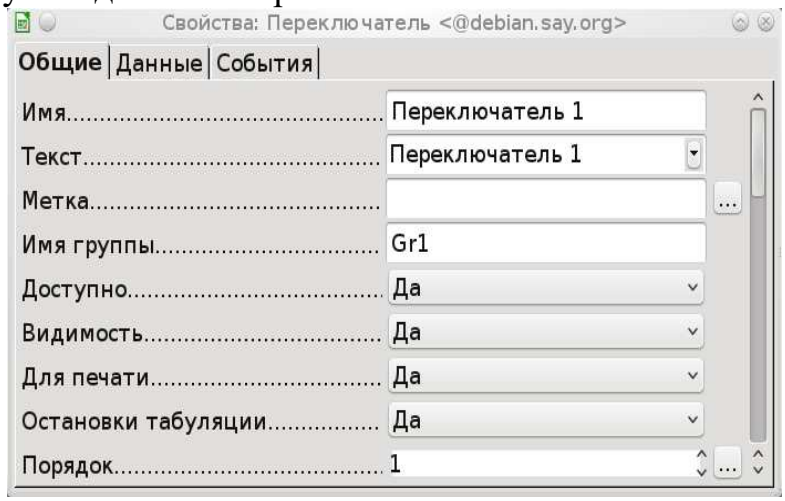



Рисунок 7.11 - Установка свойств переключателя

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.

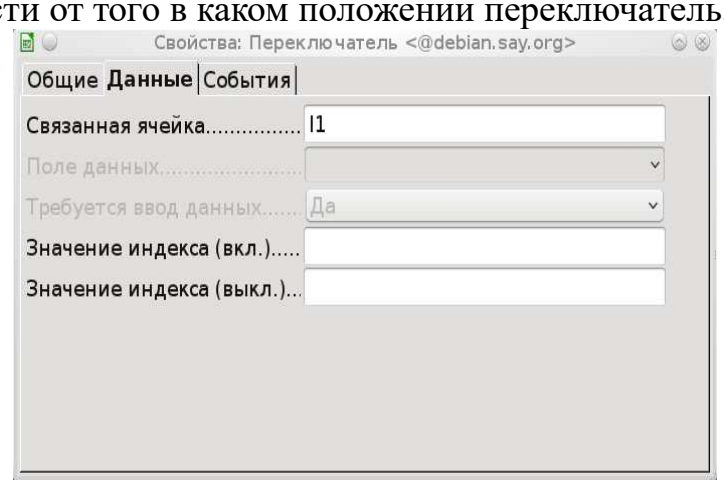



Рисунок 7.12 - Связанная с переключателем ячейка

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
```

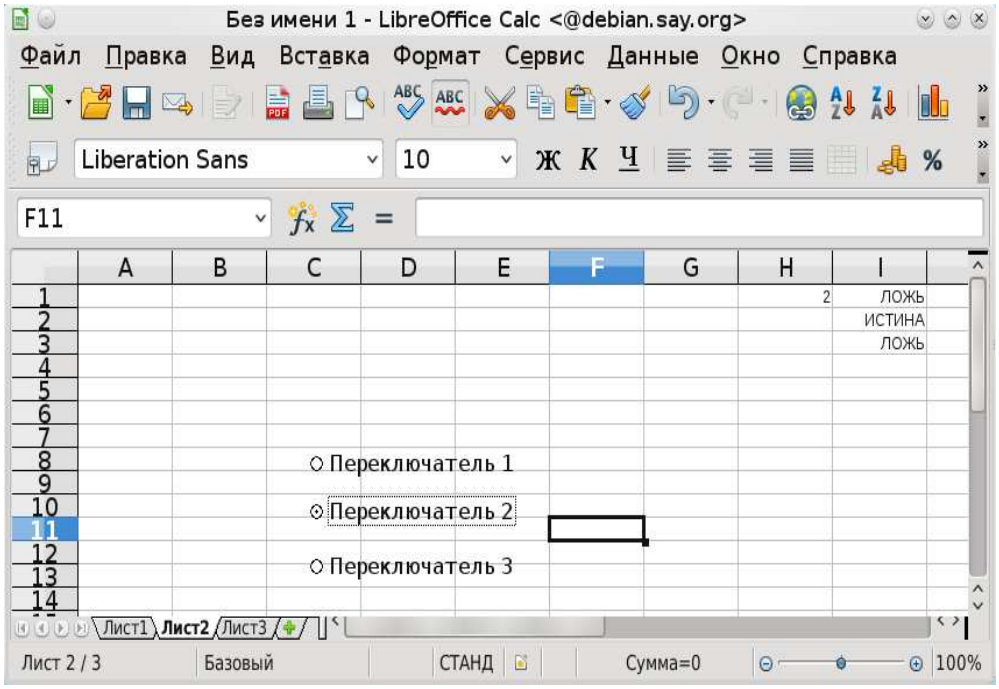


Рисунок 7.13 - Результат работы макроса

## 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
```

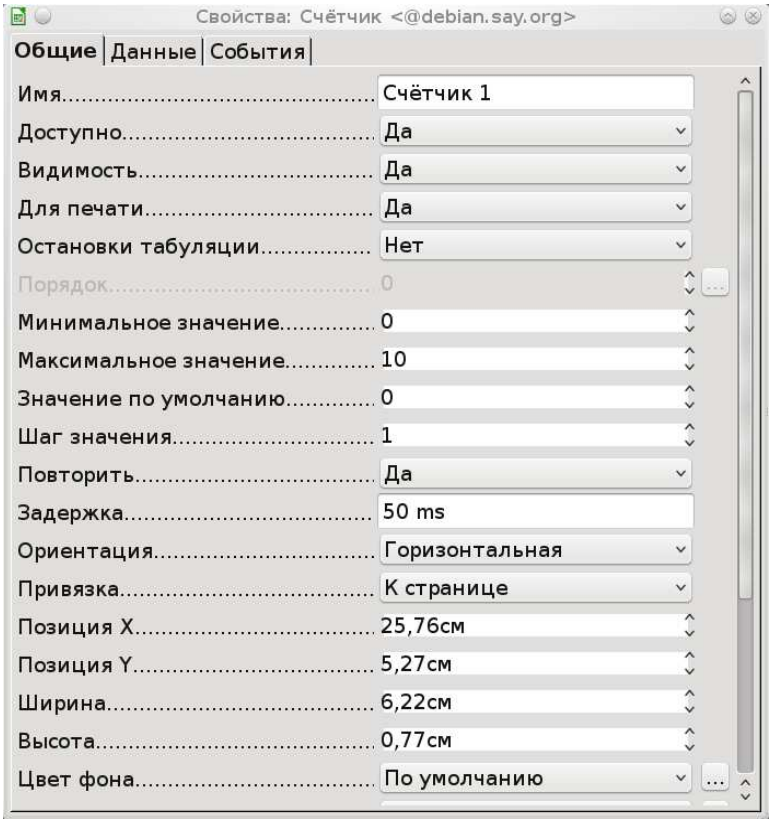


Рисунок 7.14 - Свойсвта объекта счетчик

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.

## 7.13 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.**

Задание:

Обеспечить с помощью форм и диалогов ввод данных о продаваемой одежде. Необходимо реализовать возможность ввода следующей информации: Название товара, Размер (например, S, M, L, XL), Цвет, Цена, Количество в наличии, Контактные данные продавца (имя, телефон), Город продажи. Все введённые данные должны сохраняться на листе calc. Также обеспечить возможность редактирования уже введённых записей через мастера или форму. Дополнительно реализовать форму для фильтрации данных по различным параметрам (например: по размеру, цвету, цене или городу). Отфильтрованные данные выводить на отдельном листе (например, "Фильтрованные данные") или в отдельном списке для удобства клиента.

**Вариант 5.**

Обеспечить с помощью форм и диалогов ввод данных о продаваемых спортивных товарах. Оформить возможность ввода следующей информации: Название товара, Категория товара (например, фитнес, командные виды, бега, футбол и т. д.), Цена, Количество на складе, Имя продавца, Контактный телефон продавца, Город продажи. Все введённые данные должны сохраняться на отдельном листе calc. Реализовать возможность редактирования уже внесённых записей через мастер-форму. Дополнительно необходимо создать форму или инструмент для фильтрации товаров по различным параметрам: Категории, Городам, Ценовым диапазонам, Наличию на складе (например, только товары с количеством >0). Отфильтрованные данные выводить на отдельный лист или в отдельный список для удобства

анализа.

**Вариант 6.**

Обеспечить с помощью форм и диалогов возможность ввода данных о продуктах питания, продаваемых на рынке. Необходимо реализовать возможность ввода следующей информации: название продукта, категория (например, овощи, фрукты, мясо, молочные продукты и т.д.), цена за единицу, количество в наличии, а также контактные данные продавца (имя и телефон). Данные сохранять на листе calc. Обеспечить возможность редактирования уже введённых записей с помощью мастера или диалоговой формы. Реализовать форму для фильтрации данных по различным параметрам (например, категория, цена, наличие), а отфильтрованные результаты выводить на отдельном листе (например, "Отфильтрованные" или в отдельном списке).

**Вариант 7.**

Обеспечить с помощью форм и диалогов возможность ввода данных о продаваемых напитках. Обеспечить ввод названия напитка, ценовой стоимости, количества в наличии, а также контактных данных продавца (имя и телефон). Данные должны сохраняться на листе calc. Реализовать возможность редактирования уже введенной записи с помощью мастера. Также необходимо создать форму для фильтрации данных по различным параметрам: названию напитка, ценовому диапазону, наличию и контактным данным. Отфильтрованные данные выводить на отдельный лист (например, «Отфильтрованные» или в отдельном списке).

**Вариант 8.**

Обеспечить с помощью форм и диалогов возможность ввода данных о физических лицах для регистрации в системе. Обеспечить ввод следующих данных: имя, фамилия, дата рождения, номер телефона, адрес проживания и электронная почта. Все введённые данные сохранять на отдельном листе. Реализовать возможность редактирования уже зарегистрированных данных через специального мастера или диалоговое окно. Также необходимо создать форму для фильтрации зарегистрированных лиц по различным параметрам (например, по возрасту, городу, наличию электронного ящика), а отфильтрованные данные выводить на отдельный лист или в список, чтобы удобно анализировать информацию.

**Вариант 9.**

Обеспечить с помощью форм и диалогов возможность ввода данных о зарегистрированных фирмах. При этом необходимо реализовать следующие функции:

– Ввод информации о фирме, включая название, юридический адрес, контактный номер телефона, имя руководителя и сферу деятельности.

– Сохранять введённые данные на отдельном листе.

– Обеспечить возможность редактирования уже внесённых данных с помощью мастера или формы для редактирования.

– Реализовать форму для фильтрации данных о фирмах по различным параметрам (например, по сфере деятельности, региону, руководителю).

– Отфильтрованные данные выводить на отдельном листе (например, "Отфильтрованные" или в отдельном списке).

**Вариант 10.**

Обеспечить с помощью форм и диалогов возможность ввода данных о продаваемых автомобилях. Обеспечить ввод информации о бренде, модели, году выпуска, цене, контактных данных продавца (имя и телефон), а также параметров автомобиля (пробег, тип кузова, тип двигателя и т.д.). Данные должны сохраняться на листе Excel. Реализовать возможность редактирования уже введенных записей с помощью мастера или формы. Также необходимо реализовать форму для фильтрации данных по различным параметрам (например, по бренду, цене, году выпуска), а отфильтрованные данные выводить на отдельный лист или в список для удобства просмотра клиентом.

# 8 Практическое занятие: Кодирование, представление числовой информации

## 8.1 Равномерное и неравномерное кодирование

Результат одного отдельного альтернативного выбора может быть представлен как 0 или 1. Тогда выбору всякого сообщения (события, символа т.п.) в массиве сообщений соответствует некоторая последовательность двоичных знаков 0 или 1, то есть двоичное слово. Это двоичное слово называют кодировкой, а множество кодировок источника сообщений – кодом сообщений.

Кодирование – замена информационного слова на кодовое.

Таблица 8.1 - Пример кодирования

| Информационное слово | Кодовое слово |
|---|---|
| 000 | 0000 |
| 001 | 0011 |
| 010 | 0101 |
| 011 | 0110 |
| 100 | 1001 |
| 101 | 1010 |
| 110 | 1100 |
| 111 | 1111 |

Если количество символов представляет собой степень двойки ($n = 2^N$) и все знаки равновероятны $P = (1/2)^N$, то все двоичные слова имеют длину L=N=ld(n). Такие коды называют равномерными кодами.

Более оптимальным, с точки зрения объема передаваемой информации, является неравномерное кодирование, когда разным сообщениям в массиве сообщений назначают кодировку разной длины. Причем, часто происходящим событиям желательно назначать кодировку меньшей длины и наоборот, т.е. учитывать их вероятность.

Таблица 8.2 - Кодирование словами постоянной длины

| Буква | a | b | c | d | E | f | g |
|---|---|---|---|---|---|---|---|
| Кодирование | 000 | 001 | 010 | 011 | 100 | 101 | 110 |

ld(7)=2,807 и L=3.

### 8.1.1 Кодирование Шеннона Фано

Проведем кодирование, разбивая исходное множество знаков на равновероятные подмножества, то есть так, чтобы при каждом разбиении суммы вероятностей для знаков одного подмножества и для другого

подмножества были одинаковы. Для этого сначала расположим знаки в порядке уменьшения их вероятностей. Затем будем делить таблицу так, чтобы сумма вероятностей символов в одной части таблицы, была примерно равна сумме вероятностей в другой таблице, назначим одной части таблицы начальный символ 1, другой символ 0, затем продолжим делить получившиеся группы символов, так же назначая им следующие коды 1 и 0.

Итоговый результат приведен ниже, данный код получил название код Шеннона-Фано.

Таблица 8.3 – Пример неравномерного кодирования Шеннона-Фано

| Символ | Вероятность, $P_i$ | Кодировка | Длина, $L_i$ | Вероятность*Длина, $P_i*L_i$ |
|---|---|---|---|---|
| a | 0,25 | 00 | 2 | 0,5 |
| e | 0,25 | 01 | 2 | 0,5 |
| f | 0,125 | 100 | 3 | 0,375 |
| c | 0,125 | 101 | 3 | 0,375 |
| b | 0,125 | 110 | 3 | 0,375 |
| d | 0,0625 | 1110 | 4 | 0,25 |
| g | 0,0625 | 1111 | 4 | 0,25 |

В общем случае алгоритм построения оптимального кода Шеннона-Фано выглядит следующим образом:

1. сообщения, входящие в ансамбль, располагаются в столбец по мере убывания вероятностей;

2. выбирается основание кода K (в нашем случае K=2); 3. все сообщения ансамбля разбиваются на K групп с суммарными вероятностями внутри каждой группы как можно близкими друг к другу.

4. всем сообщениям первой группы в качестве первого символа присваивается 0, сообщениям второй группы – символ 1, а сообщениям K-й группы – символ (K–1); тем самым обеспечивается равная вероятность появления всех символов 0, 1,…, K на первой позиции в кодовых словах;

5. каждая из групп делится на K подгрупп с примерно равной суммарной вероятностью в каждой подгруппе. Всем сообщениям первых подгрупп в качестве второго символа присваивается 0, всем сообщениям вторых подгрупп – 1, а сообщениям K-ых подгрупп – символ (K–1).

6. процесс продолжается до тех пор, пока в каждой подгруппе не окажется по одному сообщению.

Приведем еще один пример для получения кода Шеннона-Фано. Пусть дана таблица 8.4.

Таблица 8.4 – Символы с разной вероятностью встречаемости

| Символ | Вероятность, Pi |
|---|---|
| A | 0,2 |
| E | 0,4 |
| F | 0,125 |
| C | 0,125 |
| B | 0,15 |

Упорядочим ее по убыванию вероятностей.

Таблица 8.5 – Упорядоченная по вероятностям встречаемости таблица

| Символ | Вероятность, Pi |
|---|---|
| E | 0,4 |
| A | 0,2 |
| B | 0,15 |
| F | 0,125 |
| C | 0,125 |

Приведем пример, где мы разделили сначала таблицу на символы групп E,A и группы (B, F, C), так как вероятности этих групп 0.6 и 0.4 (рисунок 8.1).

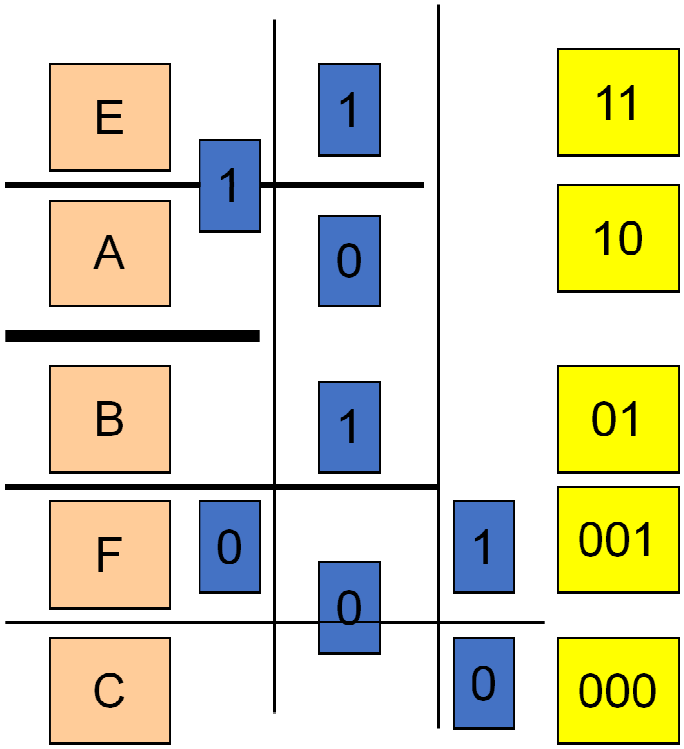



Рисунок 8.1 – Пример построения кодовой таблицы методом Шеннона-Фано

Либо разбив сначала на группы 0.4 (E), 0.6 (A, B, F, C) получим следующий результат (рисунок 8.2).

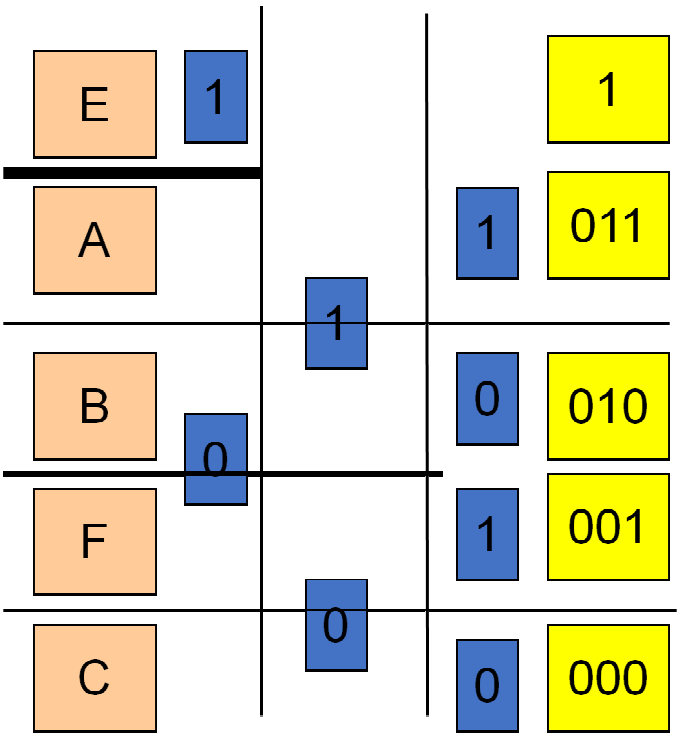


Рисунок 8.2 – Другой пример построения кодовой таблицы методом Шеннона-Фано

Заметим, что построение кода можно представить в виде бинарного дерева, где каждой ветви назначается знак 1 или 0, например, правой 1, а левой 0.

Рассмотрим пример, где возьмем не вероятности, а частоту встречаемости символа (но это, практически, одно и то же, если поделить на общее количество встреченных символов).

A (частота встречаемости 50)
B (частота встречаемости 39)
C (частота встречаемости 18)
D (частота встречаемости 49)
E (частота встречаемости 35)
F (частота встречаемости 24)
Упорядочим:
A (частота встречаемости 50)
D (частота встречаемости 49)
B (частота встречаемости 39)
E (частота встречаемости 35)
F (частота встречаемости 24)
C (частота встречаемости 18)

Получаем следующее дерево (рисунок 8.3):

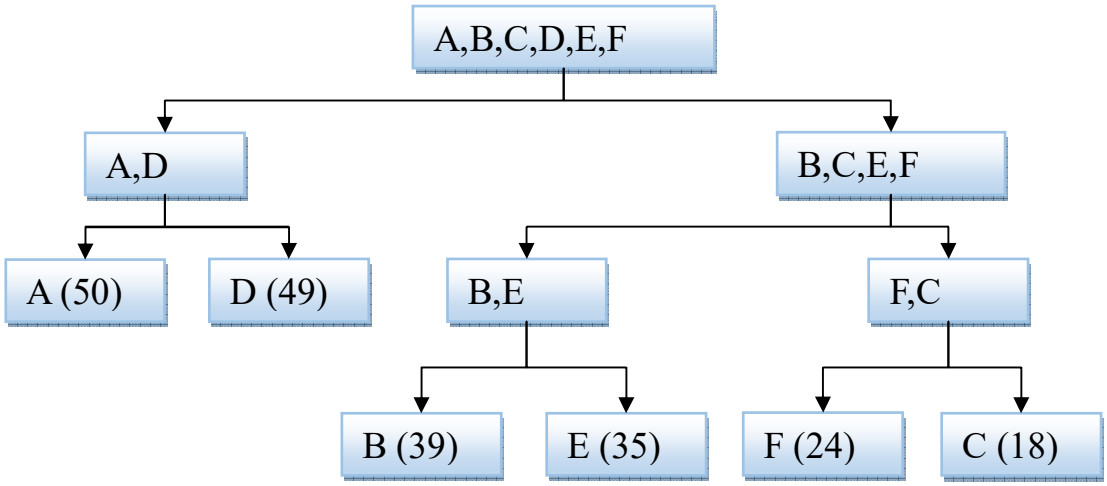



Рисунок 8.3 – пример построения дерева Шеннона-Фано по частотам встречаемости

Проходим по ветвям этого дерева, записывая код для каждой буквы.

### 8.1.2 Кодирование Хаффмана

Еще один код, относящийся к неравномерному кодированию, это код Хафмана, он используется в процедурах сжатия без потерь.

Классический алгоритм Хаффмана на входе получает таблицу частот встречаемости символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево).

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение.

2. Выбираются два свободных узла дерева с наименьшими весами.

Создается их родитель с весом, равным их суммарному весу.

4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.

5. Одной дуге, выходящей из родителя, ставится в соответствие бит 1, другой — бит 0. Битовые значения ветвей, исходящих от корня, не зависят от весов потомков.

Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

Проще это описать следующим образом: берем два наименее встречающиеся символа и объединяем вместе, приписывая им общую вероятность появления, одному символу при этом ставим в соответствие 1, другом ноль, теперь это один общий символ, повторяем все с этим новым символом и остальными.

Пример дерева для кодирования приведен ниже (рисунок 8.4):

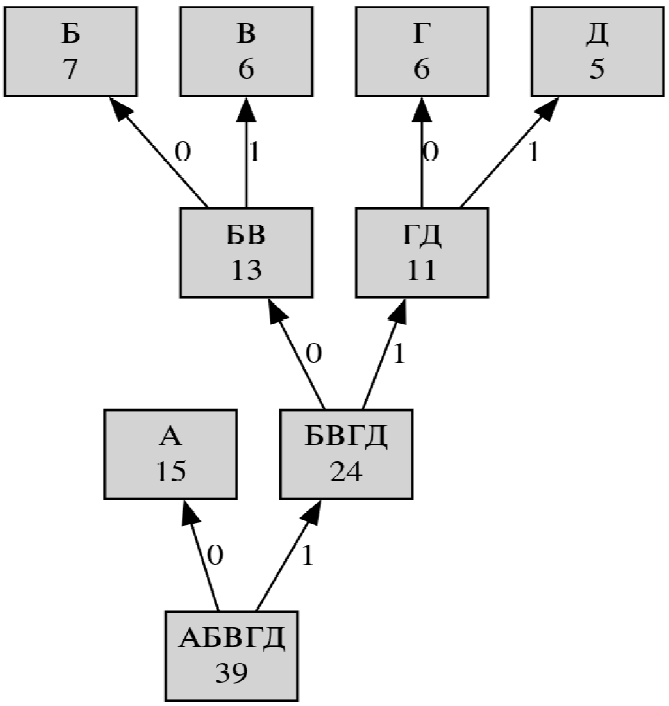



Рисунок 8.4 – Пример построения дерева Хаффмана с итоговой кодовой таблицей А: 0, Б: 100, В: 101, Г: 110, Д: 111

Здесь мы начинаем с Г и Д, далее объединяем Б и В. Потом объединяем (Г,Д) и (Б,В).

При неравномерном кодировании вводят среднюю длину кодировки, которая определяется по формуле

$$L = \sum_{i=1}^{n} P_i L_i \qquad (8.1)$$

В общем же случае связь между средней длиной кодового слова L и энтропией H источника сообщений дает следующая теорема кодирования Шеннона:

имеет место неравенство L>=H, причем L = H тогда, когда набор знаков можно разбить на точно равновероятные подмножества;

всякий источник сообщений можно закодировать так, что разность L – H будет как угодно мала.

Разность L–H называют избыточностью кода (мера бесполезно совершаемых альтернативных выборов).

Таблица 8.6 – Пример кодировки по одному символу

| Символ | Вероятность | Кодировка | Длина | В×Д |
|---|---|---|---|---|
| *A* | 0,7 | 0 | 1 | 0,7 |
| *B* | 0,2 | 10 | 2 | 0,4 |
| *C* | 0,1 | 11 | 2 | 0,2 |

Средняя длина слова: $L = 0,7+0,4+0,2=1,3$.

Среднее количество информации, содержащееся в знаке (энтропия):
$H = 0,7\times\mathrm{ld}(1/0,7)+0,2\times\mathrm{ld}(1/0,2)+0,1\times\mathrm{ld}(1/0,1) = 0,7\times0,515+0,2\times2,322+$
$+0,1\times3,322 = =1,1571$.

Избыточность $L - H = 1,3 - 1,1571 = 0,1429$.

Следует не просто кодировать каждый знак в отдельности, а рассматривать вместо этого двоичные кодирования для *nk* групп по *k* знаков.

Таблица 8.7 – Пример кодировки по два символа

| Пары | Вероятность | Кодировка | Длина | В×Д |
|---|---|---|---|---|
| *AA* | 0,49 | 0 | 1 | 0,49 |
| *AB* | 0,14 | 100 | 3 | 0,42 |
| *BA* | 0,14 | 101 | 3 | 0,42 |
| *AC* | 0,07 | 1100 | 4 | 0,28 |
| *CA* | 0,07 | 1101 | 4 | 0,28 |
| *BB* | 0,04 | 1110 | 4 | 0,16 |
| *BC* | 0,02 | 11110 | 5 | 0,10 |
| *CB* | 0,02 | 111110 | 6 | 0,12 |
| *CC* | 0,01 | 111111 | 6 | 0,06 |
| Средняя длина кодовой группы из 2-х символов равна | | | | 2,33 |

Средняя длина кода одного знака равна 2,33/2=1,165 – уже ближе к энтропии. Избыточность равна $L - H = 1,165 – 1,1571 \approx 0,008$.

Как видно при объединении в группы, избыточно стала уменьшаться и среднее число бит на символ стало ближе к энтропии.

## 8.2 Кодирование в условиях помех

При передаче данных по каналам связи, происходит их искажение за счет наличия различных шумов, воздействующих на передающийся сигнал. Часто возникает так называемая аддитивная помеха, которая изменяет уровень сигнала, что приводит к его неправильной интерпретации в процессе демодуляции.

Используют разные способы физического кодирования, потенциальное кодирование, импульсное кодирование, выделяют методы манипуляции и модуляции, которые используют ту или иную физическую характеристику сигнала, изменяющуюся за время такта (фазу, частоту и амплитуду). Понятие манипуляция относится к цифровому сигналу, а модуляция к аналоговому.

Приведем пример потенциального кода и на его примере рассмотрим основные два вида ошибок, которые возникают при передаче. Первый тип — это стирание или вставка, второй тип — это замена.

Потенциальный код предполагает передачу в течение такта положительного уровня для передачи 1 и нулевого уровня или отрицательного уровня для передачи 0. Рассмотрим код 101001110 (рисунок 8.5).

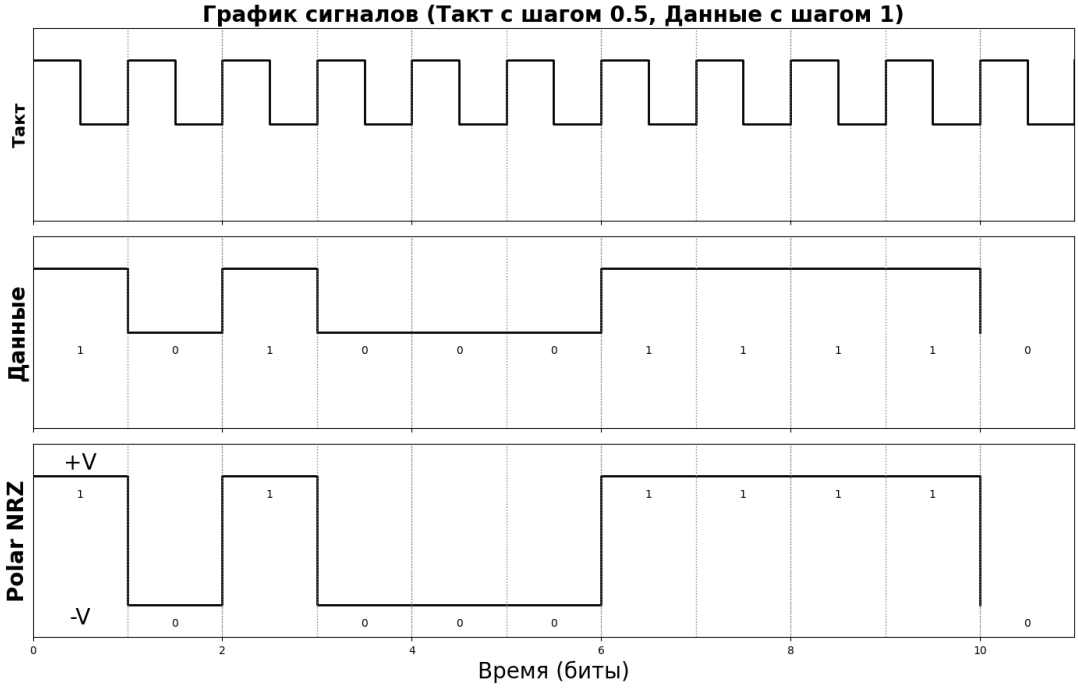



Рисунок 8.5 – Пример потенциального кодирования

Предположим, что мы передаем данные на удаленное расстояние, тогда мы будем вынуждены прокладывать еще дополнительный тактовый канал (Clock), тратя на это провода или дополнительные ресурсы, тем более не факт, что на больших расстояниях у нас не произойдет рассинхронизация. Допустим, мы попытаемся передавать информацию по одному каналу связи, используя тактовые генераторы на приемнике и передатчике, при этом пусть

мы передаем много единиц или много нулей по каналу связи. Тогда на приемнике мы получаем один и тот же уровень сигнала без каких-либо различий, при этом при рассинхронизации тактового генератора мы получим или лишнюю единицу, или не дополучим и пропустим ее, это и будет ошибкой вставки или стирания. Здесь мы видим, что наш код плохо самосинхронизован, потому что на приемнике не видно прихода нового символа, не происходит смена физического состояния, которое мы могли бы отличить. Потому все использующиеся физические коды самосинронизированы, например, биполярный импульсный код, код AMI, код MLT-3, манчестерский код, в них всегда есть смена состояния в такте, которая может быть различена приемником. Здесь представлен код AMI (рисунок 8.6), который синхронизован по единицам, каждая новая 1 кодируется новой полярностью. Более того, так еще можно и отслеживать ошибки, если, например, за положительной полярностью следует опять положительная, а не отрицательная. Для улучшения синхронизации по нулям используются специальные коды, вводящие избыточность и запрещающие использование кодов, где, например, присутствует много нулей подряд, например, b8zs, hdb3. Использующие коды где запрещено использование множества подряд нулей или единиц, например, 4b5b.

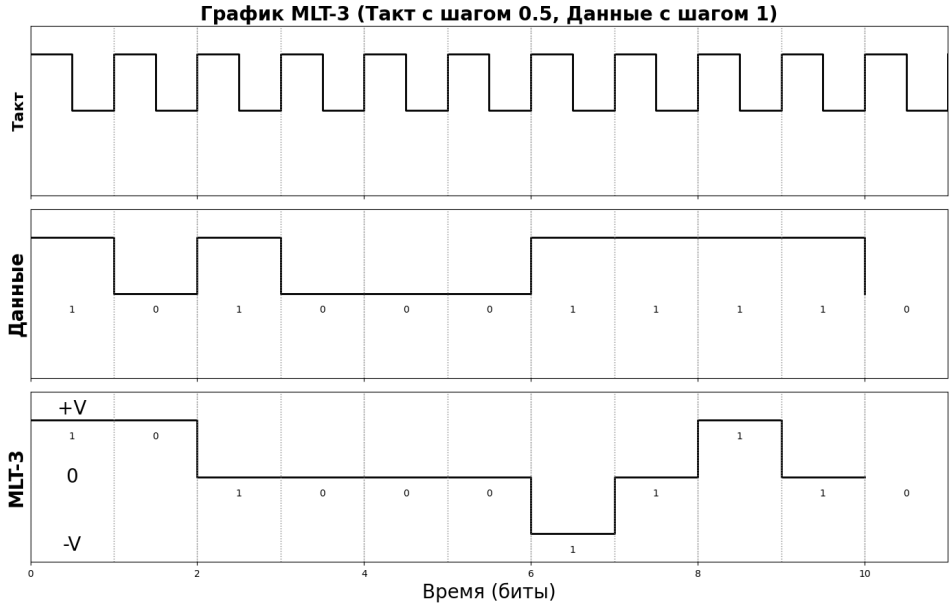



Рисунок 8.6 - Пример AMI кодирования

За счет чего происходят искажения вида замена? За счет различных физических эффектов: затухание сигнала в канале связи, поглощение или рассеяние, возможно, что эти эффекты сильнее на каких-то длинах волн, что приводит к сглаживанию сигнала, и, например, можно запутать единицу с нулем. Так же возможны помехи, которые приводят к тому, что на приемнике появляется усиленный или ослабленный сигнал, или вовсе изменивший свое состояние.

Для того, чтобы бороться с помехами типа замена используется избыточное кодирование, дополняющее информационные биты или символы

дополнительными проверочными, здесь выделяют так называемое блочное кодирование и сверточное кодирование.

Прежде чем рассматривать данные методы, рассмотрим, что такое расстояние Хэмминга. Расстояние Хэмминга между двумя словами есть число разрядов, в которых эти слова различаются.

Примеры:

Расстояние Хэмминга d(000, 011) есть 2.

Расстояние Хэмминга d(10101, 11110) равно 3

В процессе декодирования происходит исправления ошибок, если они произошли. Рассмотрим пример:

Пусть есть множество кодовых слов {00000,01101,10110,11011}

Если полученное слово 10000, то декодируем в «ближайшее» слово 00000.

Если полученное слово 11000 – то только обнаружение ошибки, так как есть два варианта с минимальными до данного слова расстояниями хэмминга: 11000 – в 00000 или 11000 – в 11011.

Коды, в которых возможно автоматическое исправление ошибок, называются самокорректирующимися. Для построения самокорректирующегося кода, рассчитанного на исправление одиночных ошибок, одного контрольного разряда недостаточно.

Количество контрольных разрядов k должно быть выбрано так, чтобы удовлетворялось неравенство:

$$2^k \geq k + m + 1 \qquad (8.2)$$

где m количество разрядов кодируемого слова.

Минимальные значения k при заданных значениях m.

Таблица 1.8 – Соответствие минимального числа кодовых символов информационным

| Диапазон m | $k_{min}$ |
|---|---|
| 1 | 2 |
| 2-4 | 3 |
| 5-11 | 4 |
| 12-26 | 5 |
| 27-57 | 6 |

### 8.2.1 Блочные коды, код Хэмминга

Блочные коды помехоустойчивого кодирования основаны на добавлении к группе (блоку) информационных бит дополнительных кодовых проверочных бит, при этом размер блоков всегда постоянный и проверочные биты проверяют только данный блок, например, код Хэмминга.

Код Хэмминга является систематическим кодом, что означает, что

исходные данные (сообщение) добавляются к закодированному сообщению. Биты проверки вычисляются путем выбора битов, которые являются степенями двойки (например, 1, 2, 4, 8, 16 и т.д.) и не включены в закодированное сообщение. Затем каждый бит проверки вычисляется как контрольный бит для соответствующих битов данных и битов проверки. Контрольные биты являются битами четности, которые показывают, сколько единиц есть в группе битов, для которых они являются контрольными битами. Данный код позволяет обнаружить что в коде есть 1 или две ошибки, а также может исправить одну ошибку.

Таблица 8.9 – Таблица для построения кодов Хэмминга

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^0$ | $2^1$ | | $2^2$ | | | | $2^3$ | | | | | | | | $2^4$ | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 1 |
| | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 0 |
| | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 1 |
| | | | | | | | X | X | X | X | X | X | X | X | | | | | | | 0 |
| | | | | | | | | | | | | | | | X | X | X | X | X | X | 0 |

Как видим из таблицы проверочные биты стоят в определенных местах соответствующих степеням двойки и обозначены желтым цветом, сначала они заполняются нулями, поля которые обозначены розовым цветом, содержат информационные биты отправляемого слова. Ниже обозначены в строках разными цветами и X проверочные подпоследовательности, например, первый проверочный бит будет проверяться первой последовательностью на четность, достаточно сложить по модулю два или операцией xor все биты в отмеченных позициях. Запишем все вычисления для каждой последовательности, здесь плюс — это операция xor (напомним таблицу истинности, 0+0=0, 1+0=1,0+1=1, 1+1=0):

0+0+1+0+0+0+0+1+1+1+1=1

0+0+0+0+1+0+0+1+1+1=0

0+1+0+0+0+0+0+1+0+1=1

0+0+1+0+0+0+0+1=0

0+1+1+1+0+1=0.

Можно так же получить остаток от деления на два после суммирования (потому и называется сложение по модулю два), проверить на четность-нечетность.

Теперь вставим указанные биты в соответствующие позиции по порядку:

Таблица 8.10 – Итоговая кодовая таблица Хэмминга

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^0$ | $2^1$ | | $2^2$ | | | | $2^3$ | | | | | | | | $2^4$ | | | | | | |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |

Теперь можно передавать это сообщение по сети и в случае одиночной ошибки исправить ее. Проверим это утверждение, сделав намеренное искажение одиннадцатого бита.

Таблица 8.11 – Проверка искажения бита

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^0$ | $2^1$ | | $2^2$ | | | | $2^3$ | | | | | | | | $2^4$ | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 1 |
| | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 1 |
| | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 0 |
| | | | | | | | X | X | X | X | X | X | X | X | | | | | | | 1 |
| | | | | | | | | | | | | | | | X | X | X | X | X | X | 0 |

Проведем те же операции что и раньше:

1+0+1+0+0+1+0+1+1+1+1 = 1 1

0+0+0+0+1+**1**+0+1+1+1 = 1 2

**1**+1+0+0+0+0+0+1+0+1 = 0 4

0+0+1+**1**+0+0+0+1 = 1 8

0+1+1+1+0+1 = 0 16

Если теперь сложить полученные коды с соответствующими весами степени двойки, мы получим число 11, которое соответствует номеру ошибочного бита, можно теперь его исправить.

$01011_2=11_{10}$

Можно интерпретировать еще код Хэмминга следующим образом: смотрим, например, последовательность, если в ней ошибка есть значит ошибка среди бит данной последовательности, если ошибки в этой последовательности нет, значит тут правильные биты и можно методом исключения выбрать неправильный бит. В нашем случае, например, ошибка в 1, 2 и 4 последовательностях, значит ошибка в 11 или 15 бите, но в третьей последовательности ошибки нет, значит ошибка в 11.

Почему в методе Хэмминга удается получить номер искаженного бита? В том числе благодаря свойству xor и тому, что каждый бит под своим номером проверяется последовательностью, которая имеет определенный вес для кодирования данного числа, и когда искажается именно эти последовательности, мы получаем номер искаженного бита. Например, первый бит проверяется только первой последовательностью, потому ее искажение выделяет нам первый бит – 1, при искажении второго бита исказится и первая и вторая последовательности.

### 8.2.2 Блочные коды, циклические коды

Циклический код является линейным блочным кодом без памяти, кодер обрабатывает данные блоками одинакового размера, получая на основе

какого-то определенного количества информационных бит, какое-то определенное число проверочных. Математически код основан на теории полей Галуа (операций по основанию остатка от деления на полином). Свойство циклического кода, в том, что каждая циклическая перестановка любого кодового слова порождает кодовое слово.

### 8.2.3 Сверточные коды, код Финка

Сверточные коды получаются путем вставки между последовательностями информационных бит проверочных, при этом проверка потоковая, нет деления на блоки. Состояние сверточного кодера имеет память, то есть последующие биты зависят, не только от текущего кодируемого бита, но и от состояния кодера, зависящего от предыдущих бит. Витерби показал, что наиболее оптимальными являются сверточные коды.

Например, рекуррентный код Финка получается следующим образом:

$A_1$,$B_1$, $A_2$, $B_2$, $A_3$, $B_3$... $A_k$, $B_k$ $A_{k+1}$, $B_{k+1}$....

$B_k=A_{i-s}\oplus A_{i+s+1}$ расчет проверочного символа B.

С учетом данного способа, ошибка исправляется если в проверочных битах k1=i-s-1, и k2=i+s ошибка, и k2-k1=2s+1. Например, s=0, B1 и B2 неверные значит A2 инвертировать. Сам проверочный символ k=i неверен если ошибка при $A_{i-s}$, $A_{i+s+1}$

Часто сверточные коды описываются многочленами.

Допустим, зададим информационную последовательность:

$A(x)=a_0+a_1X+a_2X^2+\ldots$ a {0,1..}

Формируемые передаваемые биты могут описываться следующим образом:

$B_j(X)=b_0+b_1X+b_2X^2+\ldots$

$B_j(X)=G_j(X)A(X)$

где G – это порождающие полиномы.

При этом систематические коды не меняют информационную последовательность, а несистематические меняют.

Пример сверточного кода Финка в терминах порождающих полиномов с s=1: $G_1(X)=1$; $G_2(X)=1+X^2$ (рисунок 8.7).

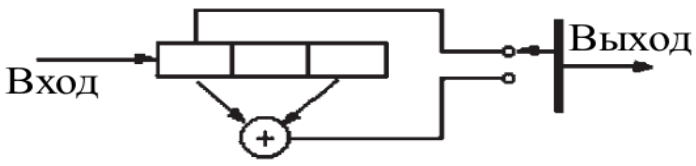



Рисунок 8.7 - Схема порождающего полинома

Для кода Финка с s=0 порождающие полиномы имеют вид:

$G_1(X)=1$; $G_2(X)=1+X$;

Приведем еще пример несистематического сверточного кода:

$G_1(X)=1+X+X^2$; $G_2(X)=1+X^2$ (рисунок 8.8).

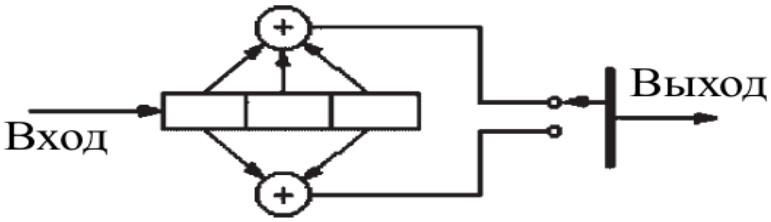



Рисунок 8.8 – Пример несистематического сверточного кода

## 8.3 Кодирование числовых данных

Система счисления — это способ записи числа с помощью специального набора знаков.

Известно, что число с дробной частью в позиционной системе счисления можно представить следующим образом:

$$p = \sum_{i=-m}^{n} a_i p^i \qquad (8.3)$$

где $a_i$ – цифра числа в позиции, p – основание системы счисления.

Таким образом, позиция цифры определяет вес цифры в числе, определяемый основанием системы счисления в степени номера позиции. Отрицательные номера определяют дробную часть числа, положительные и нулевая степень целую часть числа.

### 8.3.1 Перевод из десятичной системы в двоичную и обратно

Один из способов перевода числа из какой-либо системы в десятичную заключается в разложение на соответствующие степени и выполнения операций в десятичной системе для получения числа в десятичной системе, так же можно перевести из десятичной системы в любую другую, реализовав разложение, начиная с большей степени, так чтобы в итоге получить исходной число.

Например, переведем из десятеричной системы счисления число $25_{10}$ в двоичную систему счисления. Первая степень, которая входит в данное число это 16, остается 9, потом входит 8, и затем 1. Степени, которые входят, помечаем как 1, те которые не входят в число, помечаем как 0.

| $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|
| 16 | 8 | 4 | 2 | 1 |
| 1 | 1 | 0 | 0 | 1 |

Можем проверить, сложив соответствующие степени, где стоит 1 и получим 25.

Обратно из двоичной перевести так же просто, допустим число $10111_2$.

$1*2^4+0*2^3+1*2^2+1*2^1+1*2^0=23_{10.}$

$111000.11_2=1\cdot2^5+1\cdot2^4+1\cdot2^3+1\cdot2^{-1}+1\cdot2^{-2}=32+16+8+½+¼=56{,}75_{10}$

Так же можно переводить из одной системы счисления в десятичную операцией деления, делим на основание системы счисления, получаем остаток от деления и записываем его в качестве младшей цифры числа, затем результат от целого деления продолжаем делить на основание системы счисления, пока не получим 0.

$76_{10}=1001100_2$.

```
 76 | 2
 76   38 | 2
  0   38   19 | 2
      18    9 | 2
   1   8    4 | 2
       4    2 | 2
   0   2    1
       0
```

Очевидно, здесь для последней 1 можно не брать остаток от деления на 2, итак ясно что нужно записать 1 как значение в старшем разряде.

Дробная часть получается из целых частей (0 или 1) при ее последовательном умножении на 2 до тех пор, пока дробная часть не обратится в 0 или получится требуемое количество знаков после разделительной точки.

```
 0,375
×    2
 0,750
×    2
 1,500
×    2
 1,000
```

$0{,}375_{10}=0.011_2$

### 8.3.2 Перевод в восьмеричную, шестнадцатеричную системы

Здесь можно использовать основание системы счисления:

$421.5_8=4\cdot8^2+2\cdot8^1+1\cdot8^0+5\cdot8^{-1}=256+16+1+5/8=273{,}625_{10}$

Пример для шестнадцатеричной системы счисления:

$A7.C_{16}=10\cdot16^1+7\cdot16^0+12\cdot16^{-1}=160+7+12/16=167{,}75_{10}$

Либо если перевести в двоичную систему и далее разбивать представления на тройки или четверки бит начиная с младшего разряда для целой части и со старшего разряда для дробной части..

$100110111.001_2=467.1_8$

$100110111.001_2=137.2_{16}$

$10100101110.11_2=52E.C_{10}$

Проще всего в данном случае чтобы запомнить соответствие между

кодами и буквами в шестнадцатеричной системе счисления, перевести двоичное число в десятичное, например, 1010 = 10, тогда после девяти идет А, тогда заменяем на А. Для некратного количества цифр можно для целых дополнить нулями слева и для дробных справа.

### 8.3.3 Дополнительный код числа

Дополнительный код — это способ представления отрицательных чисел в двоичной системе счисления. Он используется для упрощения арифметических операций с отрицательными числами. В дополнительном коде отрицательные числа представляются как инвертированный двоичный код положительного числа, увеличенного на единицу. Например, число -5 в дополнительном коде будет представлено как 11111011. При выполнении операций с числами в дополнительном коде, результат также будет представлен в дополнительном коде. Дополнительный код используется во многих компьютерных системах, включая процессоры и операционные системы. Можно заменить операцию вычитания на операцию сложения с числом, представленным в дополнительном коде.

Также число в дополнительном коде получается, как вычитание из самого минимального числа со значащей цифрой в большей на единицу разрядной сетке, чем исходное число. Например, в десятичной системе счисления для числа 67, будет 100-67 = 33. Для числа 259, 1000 - 259 = 741. Можно легко заметить, что, число с минимальной цифрой 1 в большей на единицу разрядной сетке, можно представить как максимальное число в данной разрядной сетке плюс 1. Например 1000 = 999+1. При этом, запишем: 1000-x = (999-xxx)+1. Операция 999-xxx может быть представлена как инвертирование, получаемое соответствием 9-0, 8-1, 7-2, 6-3, 5-4, 4-5, 3-6, 2-7, 1-8, 0-9.

Допустим, мы проводим вычетание:

xxx-yyy, представим что для yyy мы нашли дополнительный код, который равен zzz = 1000-yyy, очевидно при этом yyy = 1000-zzz, тогда:

xxx – (1000-zzz) = xxx+zzz-1000, таким образом мы получаем сложение с дополнительным кодом, вместо вычитания.

При этом действительно очевидным становится, что для получения дополнительного кода в двоичной системе счисления достаточно провести и инвертирование числа и сложить с единицей, например, инвертирование делается просто элементом «Не», и далее делается инкрементация с помощью регистра на Т-триггерах.

Найти разность $8_{10} - 13_{10}$ в восьмибитном представлении.

| | $8_{10}$ | $-13_{10}$ |
|---|---|---|
| Прямой код | **00001000** | **10001101** |
| Обратный код | **-** | **11110010** |

| | | |
|---|---|---|
| Дополнительный код | - | **11110011** |

0 0 0 0 1 0 0 0
1 1 1 1 0 0 1 1
1 1 1 1 1 0 1 1

Теперь возьмем дополнительный код от полученного числа:

Инвертируем

0 0000 1 00

Прибавим 1

0 0000 1 00

0 0000 0 01

0 0000 1 01 = 5. Все правильно.

Кроме того, для получения отрицательного числа в дополнительном коде с большим числом разрядов, данное число расширяется единицами. Например, при расширении от байта, к слову, char к short.

### 8.3.4 Представление вещественных чисел (мантисса, порядок)

Вещественные числа хранятся и обрабатываются в компьютере в формате *с плавающей запятой*, использующем экспоненциальную форму записи чисел.

$A = M \times q^n$

где M – мантисса числа (правильная отличная от нуля дробь), q – основание системы счисления, n – порядок числа.

Диапазон ограничен максимальными значениями M и n.

Например, $123{,}45 = 0{,}12345 \cdot 10^3$

Порядок указывает, на какое количество позиций и в каком направлении должна сместиться десятичная запятая в мантиссе.

Число в формате с плавающей запятой может занимать в памяти 4 байта *(обычная точность)* или 8 байтов *(двойная точность)*.

При записи числа выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.

Мантисса **M** и порядок **n** определяют диапазон изменения чисел и их точность.

Число в форме с плавающей точкой занимает в памяти компьютера **четыре** (число обычной точности) байта или **восемь** (число двойной точности) байта.

Для записи чисел в разрядной сетке выделяются разряды для знака порядка и мантиссы, для порядка и для мантиссы. Ниже приведен пример представления по стандарту ieee754 и представления мантисса порядок с отдельно знаком порядка и знаком числа (мантиссы). В стандарте порядок получается путем процесса нормализации и добавления к настоящему порядку смещения 127. Если, например, порядок 9 бит, то 255 (рисунок 8.9).

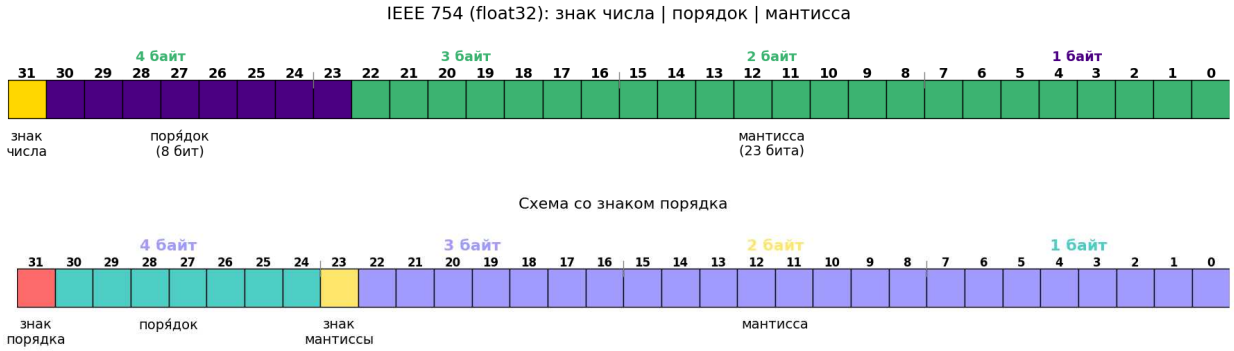



Рисунок 8.9 – Пример представления числа мантисса порядок (float32)

Пример.

Представить число 250,187510 в формате с плавающей точкой в 4-байтовой разрядной сетке.

Переведем число в двоичную систему счисления с 23 значащими цифрами:

$250{,}1875_{10} = 11111010{,}001100000000000_2$;

Нормализуем мантиссу: 11111010,001100000000000 = 0,111110100011 00000000000·$10^{1000}$;

0,11111010001100000000000 · $10^{1000}$; (мантисса положительная), (порядок положительный).

Запишем число в 32-разрядной сетке (рисунок 8.10):

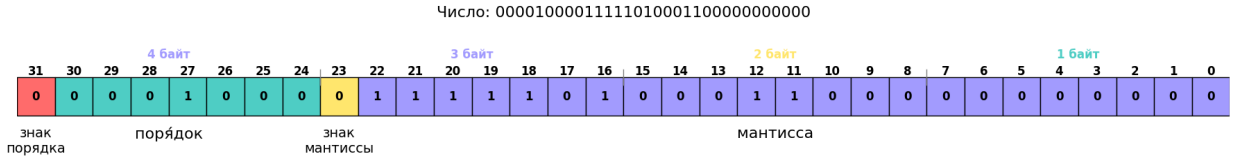



Рисунок 8.10 – Пример заполнения числа в представлении мантисса порядок

## 8.4 Практические задания на занятие

Пусть заданы числа:

d – день рождения, если d<10, то 01, 02 и т.д.

m – месяц, если m<10, то 01, 02 и т.д.

Получим число x=d+m+40 в десятичной системе.

1) x перевести в двоичную, 8-ю, 16-ю системы счисления.

2) $1100110011.10101_2$ – перевести в десятичную

Пусть

k = m/3+(16-m)/4 – округлить до целых

k(единичек)001.100k(единичек), например k=2, 11001.10011 – перевести в десятичную, восьмеричную, шестнадцатеричную.

Пусть даны числа:

$x.d_{10}$, $d.x_{10}$, $-0.000xd_{10}$ , $xd445_{10}$

Перевести в двоичную систему счисления, перевести в представлении

мантисса-порядок (учесть знак порядка и мантиссы, отвести 2, 4 байта под число).

Для данных выражений:

-x+50, -50+x, -50-x – вычислить используя дополнительный код.

Даны символы a,b,c,d,e,f,g, вероятность a – 0.d, вероятность b – 0.m, c – 0.2, d – 0.1, e – 0.1, f – 0.05, g - 1-0.m-0.d-0.45, закодировать методом Шеннона-Фано и Хафмана, равномерным кодом, посчитать среднее число бит на символ, посчитать энтропию источника сообщения, сравнить равномерный код и неравномерный.

$X_2$ – закодировать кодом Хэмминга и проверить ошибку искажения одного бита. Привести пример сверточного кода Финка.

Посчитать пропускную способность канала связи если, полоса пропускания равна 100 КГц, уровень сигнала 5 Вт, уровень шума 1 Вт.

Посчитать скорость передачи оцифрованного звукового сигнала с частотой 8000 Гц и 8, 16, 256 уровнями квантования.

Контрольные вопросы:

Какие методы сжатия без потерь и с потерями вы знаете.

Данные, знания, свойства знаний, энтропия, информация в узком смысле

1. Лазерный принтер.
2. CD-Rom.
3. Струйный принтер.
4. Наборно ассоциативный кэш.
5. Оптическая мышка.
6. Флэш память.
7. Матричный принтер.
8. Ассоциативный кэш.
9. Механическая мышка
10. Кэш прямого отображения.
11. Монитор на ЭЛТ.
12. Набор регистров и основные характеристики процессора 8086.
13. ЖК-Монитор.
14. Прерывания
15. Плазменный монитор.
16. Супер-скалярный процессор.
17. Сканнер.
18. Конвейерное исполнение команд
19. Жесткий диск.
20. Машина фон-неймана. Гарвардская и Принстонская архитектуры.

## Список источников


1. Указ Президента РФ от 30 марта 2022 г. N 166 "О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации" (с изменениями и дополнениями) [Электронный ресурс]: сайт база Гарант. URL: https://base.garant.ru/403784114/ (дата обращения: 16.11.2025).
2. Реестр российского программного обеспечения [Электронный ресурс]: сайт Реестр российского программного обеспечения. URL: https://reestr.digital.gov.ru/ (дата обращения: 16.11.2025).
3. Добро пожаловать в справку LibreOffice Writer [Электронный ресурс]: LibreOffice 25.8 Help. URL: http://help.libreoffice.org/Writer/Welcome_to_the_Writer_Help/ru. (дата обращения: 16.11.2025).
4. Скачайте последнюю версию Python для Windows [Электронный ресурс]: сайт Python. URL: https://www.python.org/downloads/ (дата обращения: 16.11.2025).
5. Учебное пособие [Электронный ресурс] / А. Я. Суханов. — Томск: ТУСУР, 2024. — 182 с. — Режим доступа: https://edu.tusur.ru/publications/10801
6. Python 3 для начинающих [Электронный ресурс]: URL: https://Pythonworld.ru (дата обращения: 12.05.2025)
7. Официальная документация Python [Электронный ресурс]: URL: https://docs.Python.org/3/tutorial/index.html. (дата обращения: 12.05.2025)
8. Справка LibreOffice [Электронный ресурс]: сайт справка LibreOffice 25.8. URL: https://help.libreoffice.org/latest/ru/text/shared/05/new_help.html. (дата обращения: 16.11.2025).

**Приложение А**
**Образец титульного листа для лабораторной работы**

Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Факультет систем управления (ФСУ)

Кафедра автоматизированных систем управления (АСУ)

**Название лабораторной работы**
Отчет по лабораторной работе №__
по дисциплине “Информатика”.

Выполнил:
Студент гр. _____
_____ И.О. Фамилия
«___»_____ 2025 г.

Проверил:

_____
(должность, ученая степень, звание)

_____ _____ И.О. Фамилия
оценка

«___»_____2025 г.

Томск 2025

## Приложение Б
## Пример оформления библиографических ссылок

**КНИГА**

**Книга одного автора**

Юрко В.А. Введение в теорию обратных спектральных задач. М.: Физматлит, 2009. 384 с.

**Книга двух авторов**

Кузелев М.В., Рухадзе А.А. Методы теории волн в средах с дисперсией. М.: Физматлит, 2009. 272 с.

**Книга трех авторов**

Баранов В.М., Карасевич А.М., Сарычев Г.А. Диагностика материалов и конструкций. М.: Высш. шк., 2007. 379 с.

**Книга четырех и более авторов**

Синергетические методы управления сложными системами / А.А. Колесников [и др.]. М.:КомКнига, 2009. 247 с.

**СТАТЬЯ**

**Статья в сборнике (сборники научных трудов, материалы докладов конференций)**

Билевич Д.В., Сальников А.С., Горяинов А.Е. Моделирование ВАХ GaAs-pHEMT-транзисторов для цифровых применений // Электронные средства и системы управления: материалы докл. XVI Междунар. науч.-практ. конф. В 2 ч. Томск: В-Спектр, 2020. Ч. 1. С. 71–73.

**Статья в журнале**

Харахардин А.Н. Дискретная топология критического состояния инертных газов // Изв. вузов. Физика. 2019. №2(62). С. 9–18.

Буханов Д.Г., Сулохин Д.В. Выявление вредоносного программного обеспечения на основе классификации графов потоков исходных кодов // Доклады ТУСУР. 2018. №3(21). С. 30–34.

Полинская М.В., Бондарь А.М., Воробьева Ж.А. Роль налогового контроля в области налога на доходы физических лиц // Вестн. Академии знаний. 2020. № 1(36). С. 331–336.

Сорбционная активность наночастиц серебра / С.С. Джимак [и др.] // Изв. вузов. Физика. 2019. №2(62) С. 114–122.

**ДИССЕРТАЦИИ, АВТОРЕФЕРАТЫ ДИССЕРТАЦИЙ**

Перминова М.Ю. Алгоритмы и программный модуль получения явных выражений коэффициентов производящих функций: дис. … канд. техн. наук. Томск, 2017. 113 с.

Перминова М.Ю. Алгоритмы и программный модуль получения явных выражений коэффициентов производящих функций: автореф. на соиск. ученой степ. канд. техн. наук. Томск, 2017. 20 с.

**НОРМАТИВНЫЕ ПРАВОВЫЕ ДОКУМЕНТЫ**

**Авторское свидетельство**

Индикатор напряжения: а. с. 3704920 СССР. № SU 1193593 A1 / Габитов Н.Ш.; заявл. 28.02.84 ; опубл. 23.11.85.

**Патент**

Корригирующее устройство для позвоночного столба: пат. 2128021 Рос. Федерация. № 97101617/14 / Иванов И.И ; заявл. 31.01.97 ; опубл. 27.03.99.

**ГОСТ**

ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления. М.: Стандартинформ, 2008. 38 с.

**ЭЛЕКТРОННЫЕ РЕСУРСЫ**

**Электронный ресурс локального доступа**

Техника спинальной анестезии [Электронный ресурс] / под ред. Е.М. Шифмана. М.: ИнтелТек, 2005. 1 электрон. опт. диск (CD-ROM).

Мишина В., Федоренко В. Тенденции развития российского валютного рынка // Исследование информационно-аналитического управления ЗАО «Московская Межбанковская валютная биржа (ММВБ). 2008. № 1 [Электронный ресурс]: сайт ЗАО ММВБ. URL: http://www.micex.ru (дата обращения: 21.03.2017).

Текущие показатели состояния Российского валютного рынка [Электронный ресурс]: сайт Московской Межбанковской Валютной Биржи. URL: http://www.micex.ru (дата обращения: 16.01.2017).

Развитие сети центров предоставления государственных и муниципальных услуг по принципу «одного окна» [Электронный ресурс]: официальный сайт Министерства экономического развития РФ. URL: http//ar.gov.ru (дата обращения: 12.12.2012).

## Приложение В
## Описание команд

### 1 Команда 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.

### 2 Команда MORE

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.

**3 Команда COPY**

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+...".

**4 Команды DEL и ERASE**

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.

**5 Команды RMDIR и RD**

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.