



ananas.sklad

Программа автоматизации складского учета
Руководство программиста
Версия 1.1.6

Андрей П. Паскаль
Павел Е. Терещенко

3 февраля 2002 г.

Аннотация

Описывается порядок настройки печатных форм документов (отчетов) программы **ananas.sklad** версии 0.1.19. Руководство предназначено для программистов, осуществляющих внедрение и сопровождение программного обеспечения **ananas**.

(с) Copyright. Всем разрешено выполнять дословное воспроизведение и распространение дословных копий данного руководства при условии, что все копии будут содержать имена авторов и данное разрешение.

Содержание

1	Настройка отчетов	4
1.1	Введение	4
1.2	Необходимое программное обеспечение и литература	6
1.3	Настройка шаблона по шагам	6
2	Использование \LaTeX для создания шаблонов	7
2.1	Общие сведения	7
2.2	Заголовки	8
2.3	Оформление текста	8
2.3.1	Использование шрифтов	9
2.3.2	Таблицы	9
2.3.3	Выравнивание и расстановка промежутков	11
2.4	Некоторые полезные команды	11
2.5	Заключение	11
3	Язык формирования отчетов	11
3.1	Пример использования языка	11
3.2	Основные конструкции языка	14
3.2.1	Операторы	14
3.3	Функции языка для доступа к объектам ananas	15
3.3.1	Работа с накладными	15
3.3.2	Доступ к данным расходной накладной	15
3.3.3	Доступ к данным приходной накладной	16
3.3.4	Доступ к данным справочника остатков	17
3.3.5	Доступ к данным справочника клиентов	18
4	Создание и редактирование шаблонов отчетов	18

1 Настройка отчетов

1.1 Введение

Программа **ananas** позволяет пользователю получать, просматривать и распечатывать различные отчеты. **ananas** имеет гибкую подсистему формирования отчетов, которая позволяет пользователям-программистам, настраивать и даже полностью менять форму любого из существующих отчетов. Для этого достаточно заменить соответствующий файл с описанием отчета на свой собственный. Имя файла должно остаться прежним. Для верстки шаблонов отчетов используется графический текстовый редактор **LyX**¹, который позволяет сверстать шаблон отчета в режиме WYSWYG, вставить в шаблон команды и операторы тэгового языка, выполняющего генерацию отчета², и сохранить шаблон отчета в файле **LaTeX**, который может использоваться **ananas**.

Данное руководство содержит описание процедуры создания файла описания отчета или шаблона отчета. Здесь Вы также найдете сведения о том, в каких файлах на диске хранятся описания отчетов **ananas**.

Каждый отчет, формируемый программой, описан в одном отдельном файле. Все описания форм и содержания отчетов, используемых программой, хранятся в каталоге “**rep**” или точнее в каталоге “**/usr/share/ananas/rep**”. В этом каталоге находятся три типа файлов: файлы с расширением **lyx**, файлы с расширением **rep** и файлы с расширением **tex**. В действительности для генерации отчетов **ananas** использует только **rep** файлы, а файлы **lyx** и **tex** **ananas** абсолютно не нужны. Они нужны Вам, если Вы хотите настроить (переделать, доделать) какой-либо из отчетов или точнее форм выходных документов³. Количество **rep** файлов в каталоге “**rep**” соответствует количеству видов отчетов, которые могут быть созданы программой. На сегодня (для **ananas-0.1.18**) это 13 отчетов:

1. **tmpl-prihod.rep** - файл с описанием печатной формы приходной накладной. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом “Приходная накладная”, щелкает указателем мыши по кнопке “Печать” или нажимает клавишу **F5**.
2. **tmpl-razhod.rep** - файл с описанием печатной формы расходной накладной. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом “Расходная накладная”, щелкает указателем мыши по кнопке “Печать” или нажимает клавишу **F5**.
3. **tmpl-rsvod.rep** - файл с описанием печатной формы журнала расходных накладных. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом “Расходные накладные”, щелкает указателем мыши по кнопке “Свод” или нажимает клавишу **F6**.

¹Если Вы совсем испорчены вордом. На самом деле настоятельно рекомендуется не использовать **LyX**, т.к. запомнить десяток макросов **LaTeX** не сложно. После этого Вы будете получать именно то, что хочется, а не танцевать с бубном вокруг WYSIWYG оболочек.

²Механизм генерирования отчета напоминает процедуру генерирования html документов при помощи РНР.

³При этом не факт, что правильно написанный на **LaTeX** шаблон отчета откроется в **LyX**.

4. **tmpl-rreestr.rep** - файл с описанием печатной формы реестра цен. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Расходные накладные", щелкает указателем мыши по кнопке "Реестр" или нажимает клавишу **F5**.
5. **tmpl-psvod.rep** - файл с описанием печатной формы журнала приходных накладных. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Приходные накладные", щелкает указателем мыши по кнопке "Свод" или нажимает клавишу **F6**.
6. **tmpl-preestr.rep** - файл с описанием печатной формы реестра цен. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Приходные накладные", щелкает указателем мыши по кнопке "Реестр" или нажимает клавишу **F5**.
7. **tmpl-movement.rep** - файл с описанием печатной формы отчета о движении товаров. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Остатки товаров", щелкает указателем мыши по кнопке "Движение" или нажимает клавишу **F5**. В появившемся при этом диалоге пользователь ставит галочку напротив надписи "Отдельные таблицы для прихода и расхода".
8. **tmpl-move2.rep** - файл с описанием печатной формы отчета о движении товаров. В отличие от отчета, формируемого файлом **tmpl-movement.rep**, данный отчет состоит из одной таблицы в которой отображен и отпуск и поступление товара в хронологическом порядке. Для формирования печатной формы и вывода ее на принтере на бумагу пользователь, работая с диалогом "Остатки товаров", щелкает указателем мыши по кнопке "Движение" или нажимает клавишу **F5**. В появившемся при этом диалоге пользователь ставит галочку на против надписи "В одну таблицу приход и расход".
9. **tmpl-ostatok-eq.rep** - файл с описанием печатной формы отчета об остатка товаров на указанную дату. В отчете перечисляются только товары, у которых остаток на указанную дату был равен нулю. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Остатки товаров. Группы", выставляет в комбобоксе "Режим просмотра" значение "4. только=0", затем щелкает указателем мыши по кнопке "Печать" или нажимает клавишу **F5**.
10. **tmpl-ostatok-gt.rep** - файл с описанием печатной формы отчета об остатка товаров на указанную дату. В отчете перечисляются только товары, у которых остаток на указанную дату был больше нуля. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом "Остатки товаров. Группы", выставляет в комбобоксе "Режим просмотра" значение "2. только>0", затем щелкает указателем мыши по кнопке "Печать" или нажимает клавишу **F5**.
11. **tmpl-ostatok-lt.rep** - файл с описанием печатной формы отчета об остатка товаров на указанную дату. В отчете перечисляются только товары, у которых остаток на указанную дату был меньше нуля. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом

“Остатки товаров. Группы”, выставляет в комбобоксе “Режим просмотра” значение “3. только<0”, затем щелкает указателем мыши по кнопке “Печать” или нажимает клавишу **F5**.

12. **tmpl-ostatok-ne.rep** - файл с описанием печатной формы отчета об остатка товаров на указанную дату. В отчете перечисляются только товары, у которых остаток на указанную дату был больше или меньше нуля. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом “Остатки товаров. Группы”, выставляет в комбобоксе “Режим просмотра” значение “5. без нулей”, затем щелкает указателем мыши по кнопке “Печать” или нажимает клавишу **F5**.
13. **tmpl-ostatok.rep** - файл с описанием печатной формы отчета об остатка товаров на указанную дату. В отчете перечисляются товары не зависимо от суммы остатка на указанную дату. Для формирования печатной формы и вывода ее на принтер пользователь, работая с диалогом “Остатки товаров. Группы”, выставляет в комбобоксе “Режим просмотра” значение “1. все знач.”, затем щелкает указателем мыши по кнопке “Печать” или нажимает клавишу **F5**.

1.2 Необходимое программное обеспечение и литература

Перед тем как приступить к настройке какого-либо отчета необходимо определиться, что Вы будете использовать - WYSIWYG оболочку для L^AT_EX- L^yX, мягко говоря не лишенную недостатков и накладывающую массу ограничений на свободу творчества, или потратите некоторое время на изучение основных команд L^AT_EX. В первом случае надо убедиться, что на компьютере установлен текстовый редактор L^yX. Для этого в графическом терминале xterm наберите команду “lyx” и нажмите <Enter> на клавиатуре. Если редактор запустился, то можно приступать к работе, если нет, то его необходимо установить либо с компакт диска с дистрибутивом Linux, либо загрузив по сети Internet с сайта “<http://www.lyx.org>”.

В случае правильного выбора - L^AT_EX’a в *разделе 2* Вы найдете описание некоторых команд L^AT_EX, достаточных для создания несложных отчетов. Для дальнейшего изучения рекомендуется:

1. Ознакомиться с книгой Котельников И.А., Чеботарев П.З. “Издательская система L^aT_eX 2_ε”, Новосибирск: Сибирский хронограф, 1998. или Львовский С.М. “Набор и верстка в пакете L^aT_eX” Москва: Космосинформ, 1995⁴;
2. Прочитать RU.TEX FAQ “<http://cyrtext.da.ru>”, в нем имеется множество ссылок на документацию, в том числе и на русском языке;
3. Желательно взять на “<ftp://ftp.vsu.ru/pub/tex>” пакет *PsCyr*, это даст Вам более качественные шрифты в отчетах;

1.3 Настройка шаблона по шагам

Процедура настройки любого отчета программы **ananas** состоит из следующих шагов:

⁴Обязательно второе издание и выше.

1. Если Вы используете L^AX, то запускаете его и открываете с его помощью файл с расширением **l_ux**, имя которого совпадает с именем файла требуемого отчета⁵. При редактировании шаблона напрямую запускаете свой любимый редактор и открываете файл с расширением **tex**;
2. Редактируете форму отчета и управляющие инструкции языка (описание языка и примеры его использования смотрите в следующем разделе);
3. Выполняете экспорт настроенной Вами формы в формат L^AT_EX через меню “Файл-Экспорт”/“L^AT_EX”. Если Вы редактируете **tex** файл, то просто сохраняете его;
4. Запускаете программу **ananas-maker** (поставляется с дистрибутивом **ananas**), указывая в качестве параметра имя файла, полученного на шаге 3. Файл должен находиться в текущем каталоге. Результатом работы программы является файл с описанием отчета имеющий расширение **rep**, готовый к использованию программой **ananas**.

После настройки отчета и получения файла с расширением **rep**, его необходимо скопировать в каталог “/usr/share/ananas/rep”, переписав соответствующий файл с таким же именем. Затем попробуйте запустить **ananas** и сформировать тот отчет, над которым Вы только что поиздевались. ВНИМАНИЕ! **ananas** в данной ситуации лучше всего запускать из графического терминала **xterm**, так как в случае присутствия ошибок в отчете (они обязательно должны быть :) сообщения об ошибках появятся в окне терминала⁶.

2 Использование L^AT_EX для создания шаблонов

2.1 Общие сведения

Документ в формате L^AT_EX, в простейшем случае, состоит из заголовка, в котором устанавливаются глобальные параметры, такие как размер страницы и полей, стиль документа, используемые пакеты макросов и т.д., и собственно самого текста с управляющими командами. При этом символы “{” “}” “\$” “&” “#” “%” “_” “~” “^” “\”, являются служебными, если они применяются в тексте, то их необходимо заменяются на команды **\{ \}** **\\$ \& \# \% \symbol{"5F} \symbol{"7E} \symbol{"5E} \symbol{"5C}** соответственно. Символ “%” означает начало комментария, все что следует за ним до конца строки игнорируется. То, что начинается с “\”, трактуется как команда. Пробел и перевод строки равнозначны, несколько пробелов подряд интерпретируются как один. Перенос строк и расстановка переносов осуществляется автоматически. Для принудительного переноса строк используется команда ****. В командах, использующих размеры, допустимы следующие единицы измерения:

⁵Обратите внимание на шаг №0 - сделайте резервную копию файлов, которые Вы хотите изменить, или лучше всего каталога “rep”, переписав его, например, в свой домашний каталог “/”.

⁶ Для того, чтобы многократно проверять настраиваемый Вами отчет нет необходимости каждый раз выходить из программы и запускать ее заново, так как при каждом формировании отчета всякий раз программа выполняет чтение файла с диска из каталога “/usr/share/ananas/rep”. И Вам достаточно позаботиться только о том, чтобы последняя версия настраиваемого Вами отчета лежала в этом каталоге в момент начала формирования отчета.

pt	пункт 0.35
pc	пика=12pt
mm	мм
cm	=10mm
in	дюйм=25.4mm
dd	пункт Дидо 0.037pt
cc	цицеро=12dd

2.2 Заголовок

Для отчетов *ananas*'а можно использовать следующий заголовок:

```
\documentclass[a4paper,12pt,landscape]{article}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[russian]{babel}
\usepackage{psycyr,longtable}
\pagestyle{empty}
```

Здесь первая строка означает, что создается документ формата A4, текст идет параллельно широкому краю бумаги, шрифт по умолчанию - 12pt. Вторая, третья и четвертая - текст на русском языке в кодировке KOI8-R. Пятая подключает пакеты *PsCyr* и *LongTable*. Первый из них даст приличные русские шрифты, второй необходим для использования макроса `\longtable`, используемого для оформления таблиц. Последняя строка отключает нумерацию страниц и колонтитулы.

Также в заголовок полезно добавить размер полей⁷, это можно сделать с помощью следующих команд:

```
\textwidth=17cm - задает ширину страницы равной 17 см,
\textheight=22cm - задает высоту страницы равной 22 см,
\oddsidemargin=1cm -увеличивает размер левого поля на 1 см8,
\oddsidemargin=1cm - увеличивает размер верхнего поля на 1 см.
```

Также в заголовке обычно пишут команды, заданные пользователем, которые задаются командой `\newcommand{имя}{выражение}`, где *имя* - название Вашего макроса, *выражение* - последовательность команд L^AT_EX. Например `\newcommand{\bs}{\symbol{"5C}}` - задает команду `\bs` выдающую символ “\”.

2.3 Оформление текста

При использовании L^AT_EX, текст начинается с команды `\begin{document}` и заканчивается командой `\end{document}`. Внутри этого блока Вы можете с помощью управляющих макросов расположить текст, таблицы и рисунки.

⁷По умолчанию в L^AT_EX размер всех полей один дюйм.

⁸Допустимы отрицательные значения, в этом случае поля уменьшаются.

2.3.1 Использование шрифтов

Для задания начертания шрифтов используются команды:

Команда	Пример шрифта
<code>\bf Шрифт</code>	Шрифт
<code>\it Шрифт</code>	<i>Шрифт</i>
<code>\sl Шрифт</code>	<i>Шрифт</i>
<code>\sf Шрифт</code>	Шрифт
<code>\sc Шрифт</code>	ШРИФТ
<code>\tt Шрифт</code>	Шрифт
<code>\rm Шрифт</code>	Шрифт

При этом область действия команды находится между { и }. Для изменения размера шрифта относительно стандартного `\normalsize`, заданного в заголовке, существуют команды:

Команда	Размер шрифта
<code>\tiny Размер</code>	Размер
<code>\scriptsize Размер</code>	Размер
<code>\footnotesize Размер</code>	Размер
<code>\tiny small</code>	Размер
<code>\normalsize Размер</code>	Размер
<code>\large Размер</code>	Размер
<code>\Large Размер</code>	Размер
<code>\LARGE Размер</code>	Размер
<code>\huge Размер</code>	Размер

Команды смены начертания и размера шрифта можно комбинировать, например команда `\it\Large крупный курсив` даст результат - “*крупный курсив*”. Подчеркивание осуществляется командой `\underline{слово}` - слово.

2.3.2 Таблицы

В L^AT_EX есть два способа задания таблиц:

с помощью команд `\begin{tabbing} \end{tabbing}`;

с помощью команд `\begin{tabular} \end{tabular}`.

Первый из них достаточно ограничен и позволяет создавать только простейшие таблицы, второй - не позволяет делать многостраничные таблицы. Поэтому мы рекомендуем подключить пакет *lontable*, созданный для многостраничных таблиц и поддерживающем все возможности окружения *tabular*. Это достигается приписыванием в заголовке документа команды `\usepackage{longtable}`. Вся дальнейшая информация относится как к окружению *lontable*, так и к *tabular*.

Таблица начинается с команды `\begin{longtable}{вид таблицы}` и заканчивается командой `\end{longtable}`, где {вид таблицы} строка выглядящая как “вертикальная

левая линейка расположение текста в ячейке вертикальная правая линейка ...”, линейки задаются символами “|” - одинарная линейка, “||” - двойная линейка, и т.д., расположение текста - “r” - выравнивание по правому краю, “l” - по левому краю, “c” - по центру. Внутри таблицы, для отрисовки горизонтальных линеек, используется команда `\hline`⁹, соответственно двойная линия - `\hline\hline`. Столбцы в таблице разделяются командой `&`, строки `\\`. Например последовательность команд

```
\begin{longtable}{||l|c|r||}
\hline
слева&по центру&справа\\
\hline
1&2&3\\
\hline
\end{longtable}
```

даст следующую таблицу

слева	по центру	справа
1	2	3

Если необходимо разбить столбец на несколько частей, используйте команду `\multicolumn`. Например

```
\begin{longtable}{||l|c|c||}
\hline
\No&\multicolumn{2}{c|}{Приход}\\
&Цена&Сумма\\
\hline
1&20.0&200.0\\
\hline
\end{longtable}
```

Этот код даст Вам таблицу

№	Приход	
	Цена	Сумма
1	20.0	200.0

При этом может понадобиться отчеркивать не всю строку, а тодько отдельные столбцы, для этого служит команда `\cline`. Ее аргументом является пара чисел, задающая отчеркиваемые столбцы. Например `\cline{2-5}` отчеркнет столбцы со второго по пятый.

Для задания таблиц с полями фиксированной ширины используются заголовки таблицы вида `\begin{longtable}{||p{размер}}...`. Пример

```
\begin{longtable}{||p{2cm}|p{3cm}|p{4cm}||}
\hline
два сантиметра&три сантиметра&четыре сантиметра\\
\hline
```

⁹У нее есть необязательный параметр задающий вид пересечения линеек, но здесь мы его описывать не будем, читайте документацию к L^AT_EX.

1&2&3
\hline
\end{longtable}
 Результат

два сантиметра	три сантиметра	четыре сантиметра
1	2	3

2.3.3 Выравнивание и расстановка промежутков

Текст между командами `\begin{flushright}` и `\end{flushright}` будет выровнен по правому краю, между `\begin{flushleft}` и `\end{flushleft}` по левому, между `\begin{center}` и `\end{center}` отцентрирован.

Для расстановки промежутков можно использовать команды `\vspace` и `\hspace`. Например `\vspace{1cm}` устанавливает вертикальный промежуток 1 см, `\hspace{3cm}` устанавливает горизонтальный промежуток 3 см.

2.4 Некоторые полезные команды

`\No` - №,
`\copyright` - с,
`\footnote{текст сноски}` - сноска,
`\underline{\hspace{4cm}}` - _____,
`\input{файл}` - включает “файл.tex” в документ.

2.5 Заключение

Выше рассмотрено меньше одного процента команд L^AT_EX, однако их достаточно для создания и редактирования большинства отчетов. Для создания более сложных документов, а также для облегчения работы, читайте документацию.

3 Язык формирования отчетов

3.1 Пример использования языка

Ниже приведен шаблон отчета “Расходная накладная” в том виде, в котором он виден в редакторе L^yX

Расходная накладная

```
<:while 1:>
<:if env.rlog.isinset!=0:>
<:zsum:=env.rlog.vsego:>
<:nsum:=env.rlog.nakrutka:>
```

```

<:osum:=zsum+nsum:>
<:onalog:=osum-(osum/1.05):>
<:i:=1; dn:=env.rlog.docnum:>
<:while substr(dn,i,1)='0' && i<9;i:=i+1; endwhile:>
      N      Дата      Покупатель
<:substr(dn,i,9-i+1):> <:env.rlog.data:> <:env.rlog.client:>

```

Наименование товара	Кол-во	Зак.цена	Зак.сумма	Отп.цена	Отп.сумма	Наценка
<pre> <:while 1:> <:env.rlog.tab.name:> <:k2:=env.rlog.tab.kol:> <:k3:=env.rlog.tab.price_in:> <:k4:=env.rlog.tab.summa:> <:k5:=env.rlog.tab.price:> <:k6:=env.rlog.tab.kol*env.rlog.tab.price:> <:k7:=env.rlog.tab.nakrutka:> </pre>	<:k2:>	<:k3:>	<:k4:>	<:k5:>	<:k6:>	<:k7:>
<pre> <:if env.rlog.tab.next!=0; break; en- dif:><:endwhile:> <:if env.rlog.percent=1:> Налог с продаж 5 % </pre>	<:onalog:>					
<:endif:> ИТОГО	<:zsum:>			<:osum:>	<:nsum:>	

```

<:endif:> <:if env.rlog.next !=0; break; endif:>
<:endwhile:>

```

Рассмотрим части, из которых состоит шаблон.

Сначала на печать выводиться заголовок **Расходная накладная**. Этот текст не заключен в скобки <: и :>, поэтому он выводится на печать без изменений. Затем идет команда начала цикла **while 1**. Цикл является безусловным. Выход из него осуществляется по команде **break**. Этот цикл позволяет выбрать требуемую для печати расходную накладную. В начале текущей является накладная с наименьшей датой. При помощи проверки **if env.rlog.isinset!=0** можно определить, была ли выбрана данная накладная пользователем для печати. Пользователь мог выбрать как одну, так и несколько накладных.

Если текущая накладная не была выбрана, то переход к следующей накладной выполняется командой **env.rlog.next**. Если эта команда возвращает значение не равное 0, то значит текущей является последняя накладная с самой большой датой и в данном случае происходит выход из цикла самого верхнего уровня. Для этого используется конструкция **if env.rlog.next !=0; break; endif**.

Далее идут команды присвоения.

В команде **zsum:=env.rlog.vsego** использовано обращение к полю текущей расходной накладной - **env.rlog.vsego**. В поле **vsego** хранится итоговая сумма в денежном выражении по расходной накладной, рассчитанная по приходным (закупочным) ценам товаров, занесенных в накладную. Значение этого поля присваивается переменной **zsum**. Следующая команда **nsum:=env.rlog.nakrutka**

присваивает переменной `nsum` значение суммарной наценки по накладной. Команда `osum:=zsum+nsum` присваивает переменной `osum` сумму переменных `nsum` и `zsum`. Таким образом в переменной `osum` получаем значение суммы по расходной накладной, рассчитанной на основе отпускных цен. В переменную `dn` командой `dn:=env.rlog.docnum` заносим номер накладной.

Цикл `while substr(dn,i,1)='0' && i<9;i:=i+1; endwhile` необходим для удаления префиксных нулей из номера документа перед печатью.

Все рассмотренные выше команды, заключенные в скобки `<: и :>`, не оставляют следа при печати. Если вы хотите вывести на печать значение переменной то её просто следует заключить в скобки `<: и :>`. Так вместо

```
<:substr(dn,i,9-i+1):> <:env.rlog.data:> <:env.rlog.client:>
```

будут выведены на печать номер, дата накладной и наименования покупателя.

Перейдем к рассмотрению таблицы.

Первая строка таблицы будет выведена на печать без изменений. Вторая строка таблицы будет распечатана столько раз, сколько раз будет выполнен цикл, оператор начала `while 1` которого находится в первой ячейке второй строки, а оператор завершения `endwhile` - в первой ячейке третьей строки.

Этот цикл обеспечивает размножение второй строки для вывода на печать информации по каждому наименованию товара, занесенного в накладную.

Обратите внимание, что в след за `while 1` идет `<:env.rlog.tab.name:>`. Это обращение к атрибуту пате товара расходной накладной. В пате храниться наименование товара и в данном случае оно будет выведено на печать.

Далее идет присвоение переменным значений других атрибутов товара, что бы затем эти переменные использовать для вывода значений в соответствующих ячейках таблицы. Хочу сразу же сказать, что использование промежуточных переменных было сделано только для того, что бы столбцы таблицы оставались достаточно узкими что бы шаблон было удобно править в редакторе LyX.

Для перехода к следующему товару, занесенному в накладную используется команда `env.rlog.tab.next`. В случае, если достигнут последний товар, команда возвращает значение не равное 0.

Третья строка таблицы будет выведена на печать только если условие `if env.rlog.percent=1` окажется выполненным. В противном случае третья строка не появиться при печати.

Четвертая строка будет выведена на печать в любом случае. Слово **ИТОГО** будет выведено без изменений. А вот вместо `<:zsum:> <:osum:> <:nsum:>` будет распечатано три значения.

Вот и все. После таблицы идут операторы завершающие условие проверки необходимости печати данной накладной и цикла, обеспечивающего обход всех накладных.

3.2 Основные конструкции языка

3.2.1 Операторы

Язык обеспечивает стандартный набор операторов. Результатом работы оператора является символьное или числовое значение.

Оператор присвоения “:=” устанавливает значение локальной переменной. Только локальные переменные могут быть установлены таким образом. Присвоение значения переменной, стоящей слева, происходит только после вычисления значения выражения, стоящего справа. Имя переменной, стоящей в левой части оператора, также может находиться и в правой части.

Примеры. ...

Инкремент “++” увеличивает значение локальной переменной на 1. Оператор может располагаться слева или справа от имени локальной переменной. Если инкремент стоит справа, то результатом выражения будет текущее значение переменной. Если - слева, то результатом будет новое значение переменной.

Декремент “--” уменьшает значение локальной переменной на 1. Он подчиняется тем же правилам, что и инкремент. Инкремент и декремент можно применять только к локальным переменным.

Строковые операции выполняются при помощи “\$” и **substr()**. \$ выполняет конкатенацию строк. Результатом является строка, полученная соединением двух других строк. Если в конкатенации участвует числовое значение, то оно автоматически конвертируется в строку перед выполнением конкатенации. **substr** вызывается с тремя параметрами. Исходная строка, смещение и количество. Результатом является подстрока, полученная из исходной строки символов, начиная с символа находящегося по указанному смещению и до символа, определяемого параметром количество. Если в качестве смещения и количества указаны символьные переменные, то их значения автоматически преобразуются в численные перед передачей в качестве параметров.

Арифметическими операциями являются сложение “+”, вычитание “-”, умножение “*”, деление “/”, модуль “%”. Результатом деления по модулю является целая часть остатка от деления.

Операторы сравнения меньше “<”, больше “>”, меньше или равно “<=”, больше или равно “>=”, равно “=”, не равно “!=”. Результатом этих операций является либо единица, если значение истина, либо ноль, если значение ложь. Если операции сравнения применяются к строковым значениям, то строки сравниваются в алфавитном порядке. Иначе значения сравниваются как числовые.

Логические операторы и “&&”, или “||”, не “!” также возвращают 1 или 0, если результат истина или ложь соответственно.

Перед переменными с числовыми значениями можно ставить знак минус “-” или знак плюс “+”. Знак минус инвертирует значение переменной. Знак плюс, позволяет получить абсолютное значение.

При написании выражений можно использовать круглые скобки для управления порядком выполнения операций.

Выражения, состоящие из одной переменной или не используемые другими конструкциями типа if, while, записывают результат в выходной поток.

Примеры....

Конструкция “**if**” позволяет выполнять набор операций в зависимости от условий. Конструкция “**if**” состоит из нескольких элементов.

Открывающего “**if**” выражения.

Блока операторов выполняемых при удовлетворения условия.

Необязательного одного или нескольких “**elseif**” выражений с сопутствующими блоками операторов.

Необязательного одного утверждения “**else**” с сопутствующим блоком операторов.

Одного закрывающего “**endif**”.

...

3.3 Функции языка для доступа к объектам **ananas**

3.3.1 Работа с накладными

В любой момент времени команда шаблона отчета может обратиться к данным только одной расходной накладной и только одной приходной накладной. Эти накладные называются текущими. В начале¹⁰ генерирования отчета текущей расходной¹¹ накладной является накладная с наименьшей датой. Для смены текущей накладной используются функции **next**¹² и **first**. **next** возвращает значение не равное 0, в случае если достигнута самая последняя накладная. Таким образом в языке шаблонов используется навигационный доступ к накладным и другим объектам данных **ananas**.

3.3.2 Доступ к данным расходной накладной

Доступ к данным расходной накладной осуществляется через конструкцию **env.rlog**. Здесь используется объектно-подобная нотация, поэтому можно говорить, что расходная накладная представлена объектом **env.rlog**. Для смены текущей расходной накладной следует использовать методы **next** и **first** данного объекта.

env.rlog.next

env.rlog.first

В случае удачного перехода к очередной расходной накладной эти методы возвращают 0. В противном случае - значение не равное 0.

Расходная накладная имеет атрибуты соответствующие по наименованию и содержанию полям таблицы базы данных, хранящейся в файле **rn.dbf**

env.rlog.doc - Символьное поле для хранения типа документа (не используется в настоящее время). Сейчас в каждую запись просто прописывается строка из двух букв РН

env.rlog.num - уникальный идентификатор расходной накладной (записи в таблицы) - первичный ключ. Символьное поле длиной 9 символов. Все значения дополняются символом <0> (ноль) до длинны 9. Пример: <000033204>. Значения генерируются и заносятся программой. Пользователь не имеет доступа к этому полю.

env.rlog.docnum - номер накладной, задаваемый пользователем. Может существовать несколько накладных с одинаковыми номерами. Символьное поле длиной 9 символов.

¹⁰ До выполнения первой команды шаблона отчета

¹¹ Тоже самое справедливо и для приходных накладных

¹² Использование **next** и **first** с конкретным типом накладной см. ниже

Все значения дополняются символом <0> (ноль) до длины 9. Пример: <000033204>

env.rlog.data - Дата расходной накладной

env.rlog.client - Наименование покупателя. Это не ссылка, а именно наименование

env.rlog.percent - Используется при начислении налога с продаж. Если имеет значение 1, то по этой накладной необходимо начислять налог с продаж. Иначе поле пустое

env.rlog.opl - Поле для контроля оплат (не используется в настоящее время)

env.rlog.bn - Форма оплаты (не используется в настоящее время)

env.rlog.kol - Количество позиций в накладной (не используется в настоящее время)

env.rlog.vsego - Сумма по накладной

env.rlog.prim - Используется при начислении налога с продаж. Если имеет значение "НП 5%", то по этой накладной необходимо начислять налог с продаж. Иначе поле пустое. Также используется при печати отчетов

env.rlog.nakrutka - Наценка по накладной

Все эти атрибуты доступны только для чтения. Таким образом исключена возможность исправления данных в процессе генерирования отчета.

В расходной накладной есть табличная часть, в которую заносятся данные о товарах. Для доступа к атрибутам табличной части используется объект **env.rlog.tab**, который имеет атрибуты соответствующие полям таблицы *razhod.dbf*

env.rlog.tab.num - идентификатор расходной накладной

env.rlog.tab.tip - наименование группы товаров, к которой принадлежит данный товар

env.rlog.tab.name - Наименование товара. Это не ссылка, а именно наименование

env.rlog.tab.kol - Количество отпускаемого товара

env.rlog.tab.price - Отпускная цена товара

env.rlog.tab.summa - Сумма по товару

env.rlog.tab.nakrutka - Наценка по товару

env.rlog.tab.price_in - Закупочная цена товара

env.rlog.tab.tovar_id - Идентификатор товара (не используется в настоящее время)

В любой момент времени можно обратиться к атрибутам только одной строки табличной части накладной. Эта строка называется текущей. Для смены текущей строки используются методы **next** и **first**

env.rlog.tab.next

env.rlog.tab.first

3.3.3 Доступ к данным приходной накладной

Доступ к данным приходной накладной осуществляется через конструкцию **env.plog**. Здесь используется объектно-подобная нотация, поэтому можно говорить, что приходная накладная представлена объектом **env.plog**. Для смены текущей приходной накладной следует использовать методы **next** и **first** данного объекта.

env.plog.next

env.plog.first

В случае удачного перехода к очередной приходной накладной эти методы возвращают 0. В противном случае - значение не равно 0.

Приходная накладная имеет атрибуты соответствующие по наименованию и содержанию полям таблицы базы данных, хранящейся в файле `pn.dbf`

env.plog.doc - Символьное поле для хранения типа документа (не используется в настоящее время). Сейчас в каждую запись просто прописывается строка из двух букв ПН

env.plog.num - уникальный идентификатор приходной накладной (записи в таблицы) - первичный ключ. Символьное поле длиной 9 символов. Все значения дополняются символом <0> (ноль) до длины 9. Пример: <000033204>. Значения генерируются и заносятся программой. Пользователь не имеет доступа к этому полю.

env.plog.docnum - номер накладной, задаваемый пользователем. Может существовать несколько накладных с одинаковыми номерами. Символьное поле длиной 9 символов. Все значения дополняются символом <0> (ноль) до длины 9. Пример: <000033204>

env.plog.data - Дата приходной накладной

env.plog.client - Наименование покупателя. Это не ссылка, а именно наименование

env.plog.opl - Поле для контроля оплат (не используется в настоящее время)

env.plog.bn - Форма оплаты (не используется в настоящее время)

env.plog.kol - Количество позиций в накладной (не используется в настоящее время)

env.plog.vsego - Сумма по накладной

env.plog.prim - В настоящее время не используется

Все эти атрибуты доступны только для чтения. Таким образом исключена возможность исправления данных в процессе генерирования отчета.

В приходной накладной есть табличная часть, в которую заносятся данные о товарах. Для доступа к атрибутам табличной части используется объект **env.plog.tab**, который имеет атрибуты соответствующие полям таблицы `prihod.dbf`

env.plog.tab.num - идентификатор приходной накладной

env.plog.tab.tip - наименование группы товаров, к которой принадлежит данный товар

env.plog.tab.name - Наименование товара. Это не ссылка, а именно наименование

env.plog.tab.kol - Количество отпускаемого товара

env.plog.tab.price - Отпускная цена товара

env.plog.tab.summa - Сумма по товару

env.plog.tab.tovar_id - Идентификатор товара (не используется в настоящее время)

В любой момент времени можно обратиться к атрибутам только одной строки табличной части накладной. Эта строка называется текущей. Для смены текущей строки используются методы **next** и **first**

env.plog.tab.next

env.plog.tab.first

3.3.4 Доступ к данным справочника остатков

env.ostatok

3.3.5 Доступ к данным справочника клиентов

`env.clnt`

4 Создание и редактирование шаблонов отчетов

Теперь, когда Вы ознакомились с основными конструкциями языка формирования отчетов, рассмотрим создание и редактирование шаблонов с использованием только текстового редактора. Зачем это нужно? Во-первых, если Вы посмотрите на шаблоны, созданные в \LaTeX , то увидите в них массу мусора, который вставляется \LaTeX с одному ему ведомыми целями и потенциально может вызвать проблемы при выдаче отчетов. Во-вторых - \LaTeX не поддерживает все команды \LaTeX , а они могут Вам понадобиться. В третьих, при наличии практики, создание и отладка шаблонов, без использования дополнительных средств, происходит гораздо быстрее.

Итак создадим расходную накладную, соответствующую требованиям Российского законодательства. В первую очередь создадим файл *origin.tex*, в котором определим команды для реквизитов фирмы.

% Данный файл служит для определения реквизитов Вашей организации.

% (c) Ilya Alexeev, Pavel Tereshchenko

`\newcommand{\name}{ООО "Рога и копыта"г. Нью Васюки}`

`\newcommand{\address}{ Бендер стрит, д. 65/67}`

`\newcommand{\bank}{ ОАО Бзнк оф Нью Васюки }`

`\newcommand{\bik}{ 011020654}`

`\newcommand{\fax}{ 275-18-56}`

`\newcommand{\inn}{ 7814203516}`

`\newcommand{\kschet}{ 30104810500600000959}`

`\newcommand{\okonh}{ 67100}`

`\newcommand{\okpo}{ 31936420}`

`\newcommand{\rschet}{ 40502590200000001931}`

`\newcommand{\tel}{ 273-22-22}}`

`\newcommand{\boss}{ О. Бендер}}`

`\newcommand{\buh}{ А. Балаганов}}`

Затем перейдем непосредственно к редактированию файла шаблона расходной накладной *tmpl-razhod.rep*.

% Шаблон расходной накладной

% (c) Andey Paskal, Ilya Alexeev, Pavel Tereshchenko

`<:start:>`

`<:setoutput file("lastreport.tex"):>`

Задаем имя выходного файла.

`\documentclass[russian,a4paper,landscape]{article}`

`\usepackage[koi8-r]{inputenc}`

`\usepackage[T1]{fontenc}`

`\usepackage{babel}`

`\usepackage{longtable}`

`\textwidth=260mm`

\textheight=180mm

\topmargin=-10mm

\hoffset=-1cm

\voffset=-1cm

\oddsidemargin=-10mm

Устанавливаем шрифт, кодировку, размеры бумаги и полей, подключаем пакет *longtable*.

\pagestyle{empty}

Отключаем колонтитулы и нумерацию страниц.

\input{origin}

Подключаем созданный ранее файл *origin.tex* с новыми командами для реквизитов.

\begin{document}

Начало собственно шаблона.

<:while 1:>

<:if env.rlog.isinset!=0:>

Цикл выбора накладной.

Поставщик: \name

ИНН \inn

Адрес: \address

Р/сч \rschet, \bank

К/сч \kschet, БИК \bik

Код по ОКОНХ: \okonh, Код по ОКПО: \okpo

Тел.: \tel Факс: \fax

Грузоотправитель и его адрес: \name, \address

Реквизиты продавца.

Грузополучатель и его адрес:

<:env.rlog.clnt.name:>, <:env.rlog.clnt.address:>

Достаем из базы адрес грузополучателя.

<:i:=1; dn:=env.rlog.docnum:>

<:while substr(dn,i,1)='0' && i<9;i:=i+1; endwhile:>

Получаем номер накладной и удаляем из него лишние нули.

**{\bf\large Накладная N <:substr(dn,i,9-i+1):> от <:env.rlog.data:>} **

Крупным жирным шрифтом печатаем номер и дату накладной.

Покупатель: <:env.rlog.clnt.name:>

ИНН: <:env.rlog.clnt.inn:>

Адрес: <:env.rlog.clnt.address:>

Р/сч: <:env.rlog.clnt.schet:>, <:env.rlog.clnt.bank:>

К/сч: <:env.rlog.clnt.korschet:> БИК: <:env.rlog.clnt.bik:>

Тел: <:env.rlog.clnt.phone:> Факс:<:env.rlog.clnt.fax:>

Реквизиты покупателя.

Основание:

Дополнение:

<:nds:=20:> <:snds:=0:> <:sum:=0:> <:i:=1:> <:tsnds:=0:> <:ng:=0:> <:sng:=0:>

Инициализируем переменные

{ \begin{longtable}{|p{12mm}|p{100mm}|p{15mm}|p{25mm}| p{35mm}|p{35mm}|}

Создаем таблицу из шести столбцов шириной 12мм, 100мм, 15мм, 25мм, 35мм и 35мм.

\hline

Проводим горизонтальную линию.

N& Наименование товара& Ед. Изм.& Количество& Цена с НДС &Всего с НДС

Задаем шапку таблицы.

\hline \hline

Отделяем ее двойной горизонтальной линией.

<:while 1:>

Начало цикла заполнения таблицы.

<:if i<10;substr(i,1,1):>

<:else; i;endif:>

<:i=i+1:>&

Печатаем номер по порядку.

<:env.rlog.tab.name:>&

Наименование товара.

шт.&

Единицы измерения.

<:tm:=env.rlog.tab.kol;j:=1:>

<:while substr(tm,j,1)!='.';j:=j+1;endwhile:>

<:substr(tm,0,j):>&

Количество.

<:env.rlog.tab.price * 1.2:>&

Цена с НДС за единицу товара.

<:t:=env.rlog.tab.kol * env.rlog.tab.price:>

<:tsnds:=env.round(env.rlog.tab.price * 1.2):>

<:sn:=tsnds*env.rlog.tab.kol-t:>

<:snds:=snds+sn:>

<:vs:=t+sn:>

<:vs:>

Цена с НДС за весь товар.

\hline

Отделяем сформированную строчку в таблице от последующих горизонтальной чертой.

<:sum:=sum+vs:>

Добавляем к общей сумме по накладной сумму по текущей позиции.

<:if env.rlog.nalog!=0;

ng:=env.round(tsnds/20)*env.rlog.tab.kol;

sng:=sng+ng;endif:>

Если налог с продаж отличен от нуля, добавляем к общей сумме налога с продаж налог с продаж по текущей позиции.

<:if env.rlog.tab.next!=0;break;endif:>

<:endwhile:>

Если в накладной кончились записи, выход из цикла.

<:if env.rlog.nalog!=0:>

Если налог с продаж отличен от нуля. **\cline6-6**

Подчеркиваем последнюю колонку. **\multicolumn{5}{r}{\bf Всего:}&**

\bf <:sum:>}

Пишем в пятой колонке "Всего:", а в шестой - сумму.

```
<:endif:>
```

```
\cline{6-6}
```

```
<: total:=sum:>
```

```
<:if env.rlog.nalog!=0 :> \multicolumn{5}{r}{\bf Сумма налога с продаж:}& {\bf
```

```
<:sng:>}}\\
```

```
\cline{6-6}
```

```
<:endif:>
```

Аналогично выводим сумму налога с продаж.

```
\multicolumn{5}{r}{\bf Всего к оплате:} & {\bf <: total+sng :>}}\\
```

```
\cline{6-6}
```

И общую сумму.

```
\end{longtable}
```

Конец таблицы.

```
\vspace{1cm}
```

Промежуток 1 см.

```
Сумма прописью: <:env.rlog.nspell(total):>\\
```

Сумма прописью :)

```
\vspace{1cm}
```

```
\Отпустил:{\underline{\hspace{4cm}}}}\\
```

```
\Получил:{\underline{\hspace{4cm}}}}\\
```

После вертикального промежутка в 1 см, печатаем жирным шрифтом "Отпустил:" и "Получил:", оставляя подчеркнутые промежутки для подписей.

```
<:endif:><:if env.rlog.next !=0; break; endif:><:endwhile:>\\
```

Конец цикла выбора накладной.

```
\end{document}
```

```
<:endsetoutput:>
```

```
<:endtmpl:>
```

Конец шаблона.

Вид накладной, выписанной с использованием этого шаблона показан на рисунке [4](#)

Предметный указатель

- [:>, 12](#)
- [<:, 12](#)
- арифметические операции, [14](#)
- декремент, [14](#)
- идентификатор накладной, [17](#)
- инкремент, [14](#)
- логические операторы, [14](#)
- оператор присвоения, [14](#)
- операторы сравнения, [14](#)
- расширение lxx, [4](#)
- расширение per, [4](#)
- строковые операции, [14](#)
- текущая накладная, [15](#)
- текущая строка, [16](#)
- LyX, [4](#)

- [break, 12](#)

- [cc, 8](#)
- [cm, 8](#)

- [dd, 8](#)

- [env.plog.bn, 17](#)
- [env.plog.client, 17](#)
- [env.plog.data, 17](#)
- [env.plog.doc, 17](#)
- [env.plog.docnum, 17](#)
- [env.plog.first, 16](#)
- [env.plog.kol, 17](#)
- [env.plog.next, 16](#)
- [env.plog.num, 17](#)
- [env.plog.opl, 17](#)
- [env.plog.prim, 17](#)
- [env.plog.tab.first, 17](#)
- [env.plog.tab.kol, 17](#)
- [env.plog.tab.name, 17](#)
- [env.plog.tab.next, 17](#)
- [env.plog.tab.num, 17](#)
- [env.plog.tab.price, 17](#)
- [env.plog.tab.summa, 17](#)
- [env.plog.tab.tip, 17](#)
- [env.plog.tab.tovar_id, 17](#)
- [env.plog.vsego, 17](#)

- [env.rlog.bn, 16](#)
- [env.rlog.client, 13, 16](#)
- [env.rlog.data, 13, 16](#)
- [env.rlog.doc, 15](#)
- [env.rlog.docnum, 13, 15](#)
- [env.rlog.first, 15](#)
- [env.rlog.isinset, 12](#)
- [env.rlog.kol, 16](#)
- [env.rlog.nakrutka, 13, 16](#)
- [env.rlog.next, 12, 15](#)
- [env.rlog.num, 15](#)
- [env.rlog.percent, 13, 16](#)
- [env.rlog.prim, 16](#)
- [env.rlog.tab.kol, 16](#)
- [env.rlog.tab.nakrutka, 16](#)
- [env.rlog.tab.name, 13, 16](#)
- [env.rlog.tab.next, 13](#)
- [env.rlog.tab.num, 16](#)
- [env.rlog.tab.price, 16](#)
- [env.rlog.tab.price_in, 16](#)
- [env.rlog.tab.summa, 16](#)
- [env.rlog.tab.tip, 16](#)
- [env.rlog.tab.tovar_id, 16](#)
- [env.rlog.vsego, 12, 16](#)

- [in, 8](#)

- [LaTeX, 6](#)

- [mm, 8](#)

- [pc, 8](#)
- [pn.dbf, 17](#)
- [prihod.dbf, 17](#)
- [PsCyr, 6](#)
- [pt, 8](#)

- [razhod.dbf, 16](#)
- [rn.dbf, 15](#)

- [tmpl-move2.rep, 5](#)
- [tmpl-movement.rep, 5](#)
- [tmpl-ostatok-eq.rep, 5](#)
- [tmpl-ostatok-gt.rep, 5](#)

tmpl-ostatok-lt.rep, [5](#)

tmpl-ostatok-ne.rep, [6](#)

tmpl-ostatok.rep, [6](#)

tmpl-preestr.rep, [5](#)

tmpl-prihod.rep, [4](#)

tmpl-psvod.rep, [5](#)

tmpl-razhod.rep, [4](#)

tmpl-rreestr.rep, [5](#)

tmpl-rsvod.rep, [4](#)

while, [12](#)

xterm, [6](#)