Аннотация

Данный дипломный проект содержит постановку задачи «Разработка программы «Модуль выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции г.Узловая», алгоритм реализации данной задачи, представленный в виде блок-схемы, описание входной и выходной информации и машинную реализацию данной задачи.

Программа разработана для ПЭВМ IBM PC/AT AMD Athlon™ –1700 на языке программирования Visual FoxPro7 под управлением операционной системы Windows XP Professional SP1

Данная программа используется на любом коммерческом и государственном предприятии.

**Содержание**

Введение

1 Организационно-экономическая сущность задачи

2 Информационное обеспечение решения задачи

2.1 Режимы работы задачи

2.2 Информационная модель решения задачи

2.3 Описание входной информации

2.4 Описание выходной информации

3 Алгоритм решения задачи

3.1 Блок-схема алгоритма

3.2 Описание блок-схемы

4 Машинная реализация задачи

4.1 Характеристика технических средств

4.2 Математическое обеспечение

4.3 Обоснование выбора языка программирования

4.4 Описание программы

4.5 Инструкция пользователю по работе с программой

5 Контрольный пример

6 Экономическое обоснование программы

Список литературы

Приложения

**Введение**

В настоящее время все большее и большее распространение, как в производстве, так и в документообороте предприятий находит компьютерная техника, все шире и шире становится перечень охватываемых ею задач. Постоянно растет объем и сложность обрабатываемой информации, требуются все новые и новые виды ее представления. Как показывает практика, большинство пользователей уже не представляет себе, как бы они выполняли свою работу, с которой прекрасно справлялись еще 5 – 10 лет назад, без помощи компьютера. Вот только некоторые из преимуществ использования вычислительной техники при работе, с какими – либо данными, документами:

возможность оперативного контроля за достоверностью информации, уменьшается число возможных ошибок, при генерировании производных данных;

сразу после ввода, данные могут участвовать в различных операциях;

возможность быстрого получения необходимых отчетов;

необходимая информация может быть быстро найдена (в том числе с помощью контекстного поиска), поиск которой в обычной папке с документами затруднен;

существенная экономия времени и людских ресурсов при выполнении операций, связанных с обработкой информации;

позволяет быстро и в наиболее подходящем для конкретного человека виде просмотреть данные, что повышает их восприятие.

Все, выше изложенные преимущества, на сегодняшний день оценены не только рядовыми пользователями, но и властными структурами, особенно это касается возможности контроля и поиска информации. В нашем случае налоговая полиция заинтересована иметь единую, электронную базу по всем физическим лицам – налогоплательщикам, так как наличие такой базы существенно усложняет процесс уклонения от уплаты налогов (сокрытие, занижение доходов, повторное использование одноразовых льгот и т.д.)

Поэтому 15 июня 1998 года была принята очередная редакция инструкции государственной налоговой службы Российской Федерации №35 от 29 июня 1995года. Согласно которой, с 01 марта 1999 года все предприятия с численностью работающих свыше 100 человек обязаны предоставлять данные о доходах своих работников в налоговую инспекцию на магнитных носителях, причем в строго оговоренном формате.

На практике же налоговая инспекция требует предоставления данных на магнитных носителях организациями с общей численностью работающих свыше 20 человек.

В связи с выше изложенным, у крупных предприятий и объединений появилась острая потребность в соответствующем программном обеспечении.

В данной дипломной работе рассматривается один из режимов работы автоматизированного рабочего места бухгалтера из отдела налоговой политики предприятия, ведущего отчетность в налоговую инспекцию. Оно отвечает требованиям, предъявляемым ГНИ к отчетности.

**1 Организационно-экономическая сущность задачи**

Взимание налогов – древнейшая функция и одно из основных условий существования государства, развития общества на пути к экономическому и социальному процветанию. Этим и определяется значение рационально организованной и эффективно действующей налоговой службы.

Экономические преобразования в современной России закономерно обусловили реформирование системы налогов и создание новой налоговой службы. Реформа 1991 - 1992 г.г. учитывала, что на пути к рыночной экономике налоги становятся наиболее действенным инструментом регулирования новых экономических отношений. В частности, они призваны ограничивать стихийность рыночных процессов, воздействовать на формирование производственной и социальной инфраструктуры, укрощать инфляцию.

Определяющее значение для формирования налоговой политики, системы налогов и налоговой службы имеют законы «О государственной налоговой службе РФ», «Об основах налоговой системы в РФ», «О налоге на добавленную стоимость», «Об акцизах» и некоторые другие. Затем провести анализ организации системы налогообложения юридических лиц и необходимость совершенствования системы налогообложения в современных условиях.

С 1999 года все предприятия с численностью работающих свыше 100 человек обязаны предоставлять данные о доходах своих работников в налоговую инспекцию на магнитных носителях, причем в строго оговоренном формате. Рассматриваемый программный комплекс «Налогоплательщик ЮЛ» позволяет сформировать огромное количество отчетов различных форм не только для государственной налоговой инспекции, но и для других контролирующих организаций (ПФР, Финансовый отдел Администрации и др.)

Ввиду организационно-технической сложности комплекса, большого объёма исходного кода и огромного количества используемых нормативно-справочных и бухгалтерских документов, в данном дипломном проекте рассматривается режим выгрузки данных в текстовом формате для государственной налоговой инспекции.

Данный режим используется после формирования любого документа для его предоставления в ГНИ на магнитном носителе, как правило, один раз в квартал. При выявлении неточностей или ошибок процесс выгрузки должен быть повторен с полной соответствующей инициализацией рассматриваемого в данном дипломном проекте режима.

Входными данными для этой задачи являются базы данных в формате DBF: kp\_sd.dbf, kp\_spp.dbf, spp.dbf, kp\_sgr.dbf, sgr.dbf, xx\_pok.dbf, pok.dbf, consts.dbf, k029.dbf, kp\_ctrl.dbf, shapka.dbf, sptran.dbf, sputr.dbf, содержащие информацию о вводимых документах за различные даты текущего и предыдущих налоговых периодов.

В результате решения данной задачи производится выгрузка необходимых данных на магнитный носитель в соответствии с требованиями, предъявляемыми ГНИ, на основании информации используемых баз данных, формируемых комплексом «Налогоплательщик ЮЛ». Структура выходного файла показана в Приложении Б.

**2 Информационное обеспечение решения задачи**

**2.1 Режимы работы задачи**

НАЧАЛО

**2.2 Информационная модель решения задачи**

нети

нети

нети

нети

да

да

да

да

нети

да

Exp2f

DefAPI

InitDevice

itx=51

itx=27

itx=3

itx=2

initTheard

itx=1

Меню

КОНЕЦ

Выбор режима

Цикл работы

ZQuit

WindowPopup

kp\_sd

kp\_spp

Exp2f

Модуль выгрузки

###### HАЧАЛО

###### КОНЕЦ

kp\_ctrl

xx\_pok

consts

sptran

spp

shapka

kp\_sgr

sgr

sputr

pok

k029

File.txt

**рис 1. Схема взаимосвязи информационных модулей**

**2.3 Описание входной информации**

Для выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции используются ранее созданные базы данных kp\_sd, kp\_spp, spp, kp\_sgr, sgr, xx\_pok, pok, consts, k029, kp\_ctrl, shapka, sptran, sputr, информация в которых организована в виде DBF-файла.

База данных kp\_sd содержит описание отчетных документов, kp\_spp – справочник подформ, spp – архив вышеуказанного справочника.

Справочник граф документов содержится в базе kp\_sgr, его архив хранится в sgr.

Описание строк и архив этого справочника хранятся в xx\_pok и pok соответственно.

Справочник пользовательских констант находится в БД consts, справочник единиц измерения – в k029, контрольных соотношений – в kp\_ctrl.

Заголовочная и конечная части документа хранятся в файле shapka.

Справочники автотранспортных средств и признаков расчета уточнения реализованы файлами sptran и sputr соответственно.

Используемые файлы хранятся на жестком диске персонального компьютера.

Структура и информация используемых баз данных приведена в Приложении А. Описание входной информации дано в таблице 1.

### **Таблица 1 – Описание входной информации**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Имя БД | Имя файла | Наименование поля | | Идентифи-  катор | | Тип, максимальная длина поля | | | | | | Примечание |
| симво-льная | | дата | числовая | | |
| целая | дробная | |
| KP\_SD | KP\_SD | Код формы | | KODF | | 2 | |  |  |  | |  |
|  |  | Наименование | | NAIM | | 100 | |  |  |  | |  |
|  |  | Период отчетности | | PER | | 1 | |  |  |  | |  |
|  |  | Наименование периода отчетности | | NPER | | 11 | |  |  |  | |  |
|  |  | Раздел | | RAZDEL | |  | |  | 3 |  | |  |
|  |  | Признак печати | | PRIZ | | 1 | |  |  |  | |  |
|  |  | Признак расчета налога | | PRN | |  | |  | 1 |  | |  |
|  |  | Код единицы измерения | | C257 | | 4 | |  |  |  | |  |
|  |  | Признак наличия графы ГНИ | | GNI | |  | |  | 1 |  | |  |
|  |  | Дата ввода | | WDATA | |  | | 8 |  |  | |  |
|  |  | Признак контроля документа | | KONTROL | | 1 | |  |  |  | |  |
|  |  | Приоритет | | PRIOR | |  | |  | 2 |  | |  |
|  |  | Признак документа | | PRD | | 2 | |  |  |  | |  |
|  |  | Код дохода | | TAXCODE | | 7 | |  |  |  | |  |
|  |  | Ставка | | STAVKA | |  | |  | 9 | 3 | |  |
|  |  | Код ставки | | KSTAV | | 1 | |  |  |  | |  |
|  |  | Параграф | | PARAG | |  | |  | 2 |  | |  |
|  |  | Признак принадлежности документов одному типу | | PRTIP | |  | |  | 2 |  | |  |
|  |  | Признак наличия комплекта документов | | PRKOMP | |  | |  | 2 |  | |  |
|  |  | Цифровая точность | | ZOK | |  | |  | 7 |  | |  |
|  |  | Признак единицы измерения | | PRIZM | | 1 | |  |  |  | |  |
|  |  | КНД | | KND | | 7 | |  |  |  | |  |
| Продолжение таблицы 1 | | | | | | | | | | | | |
| Имя БД | Имя файла | Наименование поля | | Идентифи-  катор | | Тип, максимальная длина поля | | | | | | Примечание |
| симво-льная | | дата | числовая | | |
| целая | дробная | |
| KP\_SPP | KP\_SPP | Код формы | | KODF | | 2 | |  |  |  | |  |
|  |  | Код подформы | | KODP | |  | |  | 2 |  | |  |
|  |  | Наименование подформы | | NAIM | | 35 | |  |  |  | |  |
|  |  | Дата ввода | | WDATA | |  | | 8 |  |  | |  |
|  |  | Признак расчета налога | | PRN | |  | |  | 1 |  | |  |
| SPP | SPP | Код формы | | KODF | | 2 | |  |  |  | |  |
|  |  | Код подформы | | KODP | |  | |  | 2 |  | |  |
|  |  | Наименование подформы | | NAIM | | 35 | |  |  |  | |  |
|  |  | Дата ввода | | WDATA | |  | | 8 |  |  | |  |
|  |  | Признак расчета налога | | PRN | |  | |  | 1 |  | |  |
| KP\_SGR | KP\_SGR | Код формы | | KODF | | 2 | |  |  |  | |  |
|  |  | Код подформы | | KODP | |  | |  | 2 |  | |  |
|  |  | Код графы | | KODGR | |  | |  | 2 |  | |  |
|  |  | Расчетная графа | | RGR | |  | |  | 2 |  | |  |
|  |  | Наименование | | NAIM | | 50 | |  |  |  | |  |
|  |  | Наименование подформы | | NAIMP | | 35 | |  |  |  | |  |
|  |  | Дата ввода | | WDATA | |  | | 8 |  |  | |  |
|  |  | Номер графы 5 | | НGO5H | |  | |  | 2 |  | |  |
|  |  | Признак графы | | PP | | 1 | |  |  |  | |  |
|  |  | Количество нерасчетных граф документа | | CHVGR | |  | |  | 2 |  | |  |
|  |  | Количество расчетных граф | | CHRGR | |  | |  | 2 |  | |  |
|  |  | Признак наличия графы ГНИ | | GNI | |  | |  | 1 |  | |  |
| SGR | SGR | Код формы | | KODF | | | 2 |  |  |  | |  |
|  |  | Код подформы | | KODP | | |  |  | 2 |  | |  |
|  |  | Код графы | | KODGR | | |  |  | 2 |  | |  |
|  |  | Расчетная графа | | RGR | | |  |  | 2 |  | |  |
|  |  | Наименование | | NAIM | | | 50 |  |  |  | |  |
| Продолжение таблицы 1 | | | | | | | | | | | | |
| Имя БД | Имя файла | Наименование поля | | Идентифи-  катор | | Тип, максимальная длина поля | | | | | | Примечание |
| симво-льная | | дата | числовая | | |
| целая | дробная | |
|  |  | Наименование подформы | | NAIMP | | |  |  |  |  | |  |
|  |  | Дата ввода | | WDATA | | |  | 8 |  |  | |  |
|  |  | Номер графы 5 | | НGO5H | | |  |  | 2 |  | |  |
|  |  | Признак графы | | PP | | | 1 |  |  |  | |  |
|  |  | Количество нерасчетных граф документа | | CHVGR |  | | |  | 2 |  | |  |
|  |  | Количество расчетных граф | | CHRGR |  | | |  | 2 |  | |  |
|  |  | Признак наличия графы ГНИ | | GNI |  | | |  | 1 |  | |  |
| XX\_POK | XX\_POK | Условный код | | YKOD |  | | |  | 3 |  | |  |
|  |  | Наименование показателя | | NAIM | 50 | | |  |  |  | |  |
|  |  | Код строки | | KODS | 5 | | |  |  |  | |  |
|  |  | Код подформы | | KODP |  | | |  | 2 |  | |  |
|  |  | Код операции | | OPER |  | | |  | 3 |  | |  |
|  |  | Тип данных строки | | TD | 1 | | |  |  |  | |  |
|  |  | Тип данных по графам | | TD1 | 25 | | |  |  |  | |  |
|  |  | Дата ввода | | WDATA |  | | | 8 |  |  | |  |
|  |  | Признак суммы операций | | PR2 | 1 | | |  |  |  | |  |
|  |  | Код единицы измерения | | C257 | 4 | | |  |  |  | |  |
|  |  | Код строки в отчете 5-Н | | 05N | 5 | | |  |  |  | |  |
|  |  | Код бюджета | | KBUD |  | | |  | 2 |  | |  |
|  |  | Контрольное соотношение 1 | | GR1 | 10 | | |  |  |  | |  |
|  |  | Контрольное соотношение 2 | | GR2 | 10 | | |  |  |  | |  |
| POK | POK | | Условный код | YKOD | |  | |  | 3 |  | |  |
|  |  | | Наименование показателя | NAIM | | 50 | |  |  |  | |  |
|  |  | | Код строки | KODS | | 5 | |  |  |  | |  |
|  |  | | Код подформы | KODP | |  | |  | 2 |  | |  |
| Продолжение таблицы 1 | | | | | | | | | | | | |
| Имя БД | Имя файла | Наименование поля | | Идентифи-  катор | | Тип, максимальная длина поля | | | | | | Примечание |
| симво-льная | | дата | числовая | | |
| целая | дробная | |
|  |  | | Код операции |  | |  | |  | 3 |  | |  |
|  |  | | Тип данных строки |  | |  | |  |  |  | |  |
|  |  | | Тип данных по графам | TD1 | | 25 | |  |  |  | |  |
|  |  | | Дата ввода | WDATA | |  | | 8 |  |  | |  |
|  |  | | Признак операций | PR2 | | 1 | |  |  |  | |  |
|  |  | | Код единицы измерения | C257 | | 4 | |  |  |  | |  |
|  |  | | Код строки в отчете 5-Н | O5 | |  | |  | 3 |  | |  |
|  |  | | Код бюджета | KBUD | |  | |  | 2 |  | |  |
|  |  | | Контрольное соотношение 1 | GR1 | | 10 | |  |  |  | |  |
|  |  | | Контрольное соотношение 2 | GR2...GR25 | | 10 | |  |  |  | |  |
|  |  | | Код формы | KODF | | 2 | |  |  |  | |  |
| CONSTS | CONSTS | | Код константы | CONST | | 6 | |  |  |  | |  |
|  |  | | Значение константы | DATА | | 150 | |  |  |  | |  |
|  |  | | Комментарий | REMARKA | | 50 | |  |  |  | |  |
|  |  | | Код единицы измерения | С257 | | 4 | |  |  |  | |  |
|  |  | | Наименование | C258 | | 15 | |  |  |  | |  |
|  |  | | Коэффициент | K | | 10 | |  |  |  | |  |
| KP\_CTRL | KP\_CTRL | | Код формы документа | KODF | | 2 | |  |  |  | |  |
|  |  | | Дата создания документа | WDATA | |  | | 8 |  |  | |  |
|  |  | | Начальная дата действия | DAT1 | |  | | 8 |  |  | |  |
|  |  | | Конечная дата действия | DAT2 | |  | | 8 |  |  | |  |
|  |  | | Текстовый файл контрольных соотношений | FCONT | | 10 | |  |  |  | |  |
| SHAPKA | SHAPKA | | Код формы | KODF | | 2 | |  |  |  | |  |
|  |  | | Заголовочная часть документа | ZAGOL | | 10 | |  |  |  | |  |
| Продолжение таблицы 1 | | | | | | | | | | | | |
| Имя БД | Имя файла | Наименование поля | | Идентифи-  катор | | Тип, максимальная длина поля | | | | | | Примечание |
| симво-льная | | дата | числовая | | |
| целая | дробная | |
|  |  | | Заключительная часть документа | ZAKL | | 10 | |  |  |  | |  |
| SPTRAN | SPTRAN | | Код вида транспорта | NTRAN | | 6 | |  |  |  | |  |
|  |  | | Марка транспорта | MARKA | | 15 | |  |  |  | |  |
|  |  | | Наименование вида транспорта | NAIM | | 40 | |  |  |  | |  |
|  |  | | Мощность двигателя в л.с. | MOLS | |  | |  | 3 |  | |  |
|  |  | | Мощность двигателя в квт | MOKVTN | | 5 | |  |  |  | |  |
|  |  | | Отметка | MODE | |  | |  | 1 |  | |  |
|  |  | | Дата выдачи уведомления | DUVED | |  | | 8 |  |  | |  |
| SPUTR | SPUTR | | Раздел | RAZDEL | |  | |  | 3 |  | |  |
|  |  | | Код дохода | TAXCODE | | 5 | |  |  |  | |  |
|  |  | | Дата действия | NDAT | |  | | 8 |  |  | |  |
|  |  | | Признак расчета уточнения | PRUT | | 1 | |  |  | |  |  |
|  |  | | Параграф | PARAG | |  | |  | 2 | |  |  |

**2.4 Описание выходной информации**

Информация на магнитных носителях создается в виде текстовых файлов в текстовом формате DOS (модифицированная альтернативная кодировка ASCII) с именем DААААААА.txt, где D – префикс, соответствующий данным деклараций юридического лица.

ААААААА – набор букв латинского алфавита и цифр длиной от 1 до 8 символов для файлов, представляемых организациями (предприятиями, учреждениями). Для файлов, представляемых инспекциями МНС РФ, первые четыре символа представляют собой код инспекции по классификатору СОГНИ, а последующие три символа – порядковый номер файла в текущем году. Расширение txt – обязательно.

Файл содержит следующую информацию:

* информацию, идентифицирующую источник сведений;
* информацию, идентифицирующую налогоплательщика;
* информацию, идентифицирующую форму отчетности;
* информацию с данными отчетности.

Структурными элементами файла с информацией являются:

* служебная часть;
* сведения о юридическом лице;
* разделители.

Разделителями являются:

@@@ - разделитель документов;

**===** -конец файла.

Служебная часть файла служит для идентификации информации, содержащейся в файле, и состоит из следующих реквизитов: идентификатор файла; тип информации; наименование отправителя; телефон отправителя; должность отправителя; фамилия, имя, отчество лица, сформировавшего файл; количество документов в файле; версия программы, с помощью которой подготовлен файл.

Файл состоит из строк. В качестве признаков окончания строки используются пары символов “возврат каретки” и “ перевод строки ” (ASCII-коды соответственно 13 и 10, которые формируются при нажатии клавиши [Enter]).

Строка состоит из кода реквизита, символа "**:**" (двоеточие) и значения реквизита.

Текстовое (символьное) значение может состоять из прописных букв, цифр и любых других символов, за исключением символов "возврат каретки" и "перевод строки" (коды ASCII соответственно 13 и 10, которые формируются при нажатии клавиши “Enter”), "**:**" (двоеточие) и "**,**" (запятая).

Числовое значение может состоять только из цифр (в том числе и с лидирующими нулями) и может включать десятичную точку и знак "-" (минус) для отрицательных чисел.

Значение типа ДАТА имеет специальный формат для представления дат в виде ДД.ММ.ГГГГ, где ДД – номер дня, ММ – номер месяца, ГГГГ – цифры года.

Каждый реквизит записывается в отдельной строке. Признак окончания строки – пара символов с кодами ASCII соответственно 13 и 10. Код реквизита отделяется от значения знаком ":" (двоеточие). Пробелы между ними не допускаются.

Реквизиты, не имеющие значений, могут быть опущены при формировании файла (кроме обязательных реквизитов, которые заполняются нулевыми значениями).

В состав данных включаются общие реквизиты, приведенные в настоящем документе и реквизиты (показатели) конкретных форм, состав которых разрабатывается и утверждается для сдачи отчетности по каждому отчетному периоду.

Последовательность строк файла следующая:

Реквизиты, идентифицирующие источник информации;

Начало документа 1;

Реквизиты, идентифицирующие налогоплательщика 1;

Реквизиты, идентифицирующие форму отчетности 1 за конкретный период;

Реквизиты формы отчетности 1;

Реквизиты, идентифицирующие форму отчетности 2 за конкретный период;

Реквизиты формы отчетности 2;

Конец документа 1;

Начало документа 2…

Реквизиты, идентифицирующие налогоплательщика 2;

…

Конец документа 2;

Конец файла.

Все данные о налоговых декларациях налогоплательщика помещаются после идентифицирующих его реквизитов, а все данные конкретной формы отчетности располагаются после реквизитов, идентифицирующих форму. Порядок следования форм в пределах данных по одному налогоплательщику, а также порядок следования реквизитов в пределах одной формы отчетности не существенен. Для конкретных форм должна соблюдаться уникальность идентификаторов в пределах одной формы.

Описание выходного документа (текстового файла) дано в таблице 2.

Информация выходного файла дана в приложении Б.

**Таблица 2 – Описание выходной информации**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название документа | Назначение документа | Ключевые признаки | Периодичность составления | Количество | | | Куда передается |
| экз. | строк на листе | символов в строке |
| file.txt | Предоставление в ГНИ информации на магнитном носителе | Имя файла, определяемое по указанным выше правилам | 1 раз в квартал | 1 | перем. | перем. | в ГНИ |

**3 Алгоритм решения задачи**

**3.1 Блок-схема алгоритма**

да

Очистка формы (экрана)

**1**

**2**

**3**

Конец файла?

Exp2f.prg

**A**

**B**

**C**

**D**

**E**

**F**

**G**

3/B3

3/A3

нет

filenamec=kp\_sd.kodf+skw+sdt periodtxtc=yearc+',м'+alltrim(str(sperio,2))

nrepN=int((sysMonthN-1)/3)+1

‘Идет подго­товка данных’

нет

да

kp\_sd.per=‘1’

kvartc=str(int((sperio-1)/3)+1,1) filenamec=kp\_sd.kodf+kvartc+sdt periodtxtc=yearc+',к'+kvartc

нет

да

kp\_sd.per=‘2’

нет

val(sdt)>50

да

YerHlfC=iif(sPERIO>6,'2','1') KvartC=iif(sPERIO>6,'4','2')

FileNameC=KP\_SD.KODF+YerHlfC+sDT

PeriodTxtC=YearC+',К'+KvartC

нет

да

kp\_sd.per=‘3’

YearC=

‘19’+sDT

FileNameC=KP\_SD.KODF+sDT PeriodTxtC=YearC+',Г'

YearC=

‘20’+sDT

нет

да

kp\_sd.per=‘4’

нет

kp\_sd пуст?

FileNameC=''

PeriodTxtC=''

prepN=\_sysMonthN

FltDocC=

‘KONTROL=”1”’

да

getmper (param1)

getmper (param2)

Установка фильтра FltDocC на KP\_SD

2/A1

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**1**

**2**

**3**

1/G3

Открытие вре­менной БД с име­нем \_user+FileNameC в области XXZZYY

нет

да

KvartC=str(int((sPERIO-1)/3)+1,1) PeriodTxtC=YearC+',К'+KvartC

mperiod.period=’2’

Обработка Select-запроса

нет

да

YerHlfC=iif(sPERIO>6,'2','1')

KvartC=iif(sPERIO>PeriodTxtC=YearC+',К'+KvartC

mperiod.period=’3’

3/A1

да

Конец файла?

нет

да

PeriodTxtC=YearC+',Г'

mperiod.period=’4’

3/B1

нет

да

\_UseKND сущест.?

нет

да

Конец файла?

PeriodTxtC=YearC+',М'+alltrim(str(sPERIO,2))

нет

KNDTxtC=

KP\_SD.KODF

flycodC= flykodC+alltrim(str(pok.ykod,3))+';'

flycodC=‘’

KNDTxtC= iif(empty (alltrim(KP\_SD.KND)) or (alltrim (KP\_SD. KND)='0'),'1150000',KP\_SD.KND)

Открытие POK в области pok, фильтр1

нет

да

docwdata=kp\_sd.wdata

Добавление данных в БД средствами REPLACE

Переход на первую запись базы данных

Открытие POK, фильтр2

3/A1

Добавление записей по Append Blank и REPLACE

2/G3

scrC=subdat1 (pok.twd, kp\_sgr.rgr)

Переход на след. запись базы данных

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**1**

**2**

**3**

1/A2

1/A3

2/С3

2/С3

aPath=alltrim (upper(bPath))

Очистка формы (экрана)

нет

да

scrC пуст?

нет

right(aPath,1)<>'\'

typeC=substr(scrC,1,1)

sc1N=at('(',scrC)

sc2N=at(')',scrC)

zpN=at(',',scrC)

Модификация ХХ\_РОК средствами REPLACE

да

нет

aPath=aPath+'\'

zpN=0

Фильтр, переход на первую запись базы данных

да

aDrv=left (aPath,1)

lenN=val(substr(scrC,sc1N+1,sc2N-sc1N-1))

decN=0

нет

да

Конец файла?

нет

Диск в дисководе?

lenN=val(substr(scrC,sc1N+1,zpN-sc1N-1)) decN=val(substr(scrC,zpN+1,sc2N-zpN-1))

Фильтр на kp\_sgr, переход на первую запись базы данных

4/А1

да

да

‘Вставьте диск в дисковод’

Присваивание значений Ykodn, kodpn,kodsc,kodgn, rgrn, gnin, FieldnameC

нет

Конец файла?

**A**

**B**

**C**

**D**

**E**

**F**

**G**

**1**

**2**

**3**

3/G3

‘Выгрузка завер­шена’

AddDoc

да

aDrv существ?

нет

ВЫХОД

‘Неправильный путь к файлу’

AddDoc

Уничтожение

окна wtxt

Переназначение устройства вывода (на экран)

‘Идет подготов­ка файла’

да

Конец файла?

!isOkAll

Oпределение окна wtxt

нет

да

‘Идет подготов­ка данных’

‘Нет пере­даваемых показа­телей’

Активация

окна wtxt

isOkAll=

iif(count()>0,

.t., .f.)

‘Выпол­няется копирование’

SET-установки и переназначе­ние устройства вывода

WinToDos

Вывод инфор­мации в файл

ВЫХОД

**3.2 Описание блок – схемы**

Лист 1.

Блок А1 – начало модуля Exp2f, реализующего выгрузку документов;

Блок B1 – вывод сообщения ‘Идёт подготовка данных’;

Блок С1 – проверка условия;

Блок D1 – присвоение значения переменной YearC;

Блок Е1 – присвоение значения переменной YearC;

Блок F1 – присвоение значения переменной FltDocC;

Блок G1 – фильтрация по маске FltDocC базы данных KP\_SD;

Блок А2 – проверка условия достижения конца файла;

Блок В2 – проверка условия kp\_sd.per=‘1’;

Блок С2 – проверка условия kp\_sd.per=‘2’;

Блок D2 – проверка условия kp\_sd.per=‘3’;

Блок Е2 – проверка условия kp\_sd.per=‘4’;

Блок F2 – присвоение значения переменным;

Блок G2 – вызов процедуры getmper с передачей параметров;

Блок А3 – очистка формы (экрана)

Блок В3 – присвоение значения переменным;

Блок С3 – присвоение значения переменным;

Блок D3 – присвоение значения переменным;

Блок Е3 – присвоение значения переменным;

Блок F3 – проверка условия;

Блок G3 – вызов процедуры getmper с передачей параметров.

Лист 2.

Блок А1 – проверка условия period=‘2’;

Блок В1 – проверка условия period=‘3’;

Блок С1 – проверка условия period=‘4’;

Блок D1 – присвоение переменной PeriodTxtC значения;

Блок Е1 – присвоение переменной flycod значения;

Блок F1 – проверка условия;

Блок G1 – открытие БД с оконечным префиксом &&pok и фильтрация данных открываемой базы данных;

Блок А2 – присвоение переменным значений;

Блок В2 – присвоение переменным значений;

Блок С2 – присвоение переменным значений;

Блок D2 – проверка условия достижения конца файла;

Блок Е2 – присвоение переменной flycod значения;

Блок F2 – открытие БД с оконечным префиксом &&pok и фильтрация данных открываемой базы данных;

Блок G2 – переход на первую запись базы данных;

Блок А3 – открытие временной БД с переменным именем;

Блок В3 – обработка select-запроса к базе данных;

Блок С3 – проверка условия достижения конца файла;

Блок D3 – проверка условия существования UseKnd;

Блок Е3 – присвоение переменной значения;

Блок F3 – присвоение значений переменным;

Блок G3 – добавление записей в базу данных средствами APPEND BLANK и ее модификация посредством REPLACE (множественное изменение полей);

Лист 3

Блок А1 – переход на следующую запись базы данных и

завершение цикла обработки по активной базе;

Блок В1 – очистка формы (экрана);

Блок С1 – модификация БД с оконечным префиксом &&pok;

Блок D1 – фильтрация с возвратом на первую запись БД;

Блок Е1 – проверка условия достижения конца файла;

Блок F1 – фильтрация с возвратом на первую запись БД;

Блок G1 – проверка условия достижения конца файла;

Блок А2 – присвоение значений переменным;

Блок В2 – проверка условия;

Блок С2 – присвоение значений переменным;

Блок D2 – проверка условия zpN=0;

Блок Е2 – присвоение значения переменной;

Блок F2 – присвоение значений переменным;

Блок G2 – присвоение значений переменным

Блок А3 – добавление записей в базу данных средствами APPEND BLANK и ее модификация посредством REPLACE (множественное изменение полей);

Блок В3 – присвоение значения переменной;

Блок С3 – проверка условия;

Блок D3 – присвоение значения переменной;

Блок Е3 – присвоение значения переменной;

Блок F3 – проверка наличия диска в дисководе;

Блок G3 – выдача предупреждающего сообщения ‘Вставьте диск в дисковод’.

Лист 4

Блок А1 – проверка существования aDrv (устройства вывода);

Блок В1 – выдача сообщения ‘Неправильный путь к файлу’;

Блок С1 – выдача сообщения ‘Идет подготовка файла’;

Блок D1 – определение окна wtxt;

Блок Е1 – активация окна wtxt;

Блок F1 – SET-установки и переназначение устройства ввода вывода (экран –> файл);

Блок G1 – вывод информации в файл (печатью по ?);

Блок А2 – вызов процедуры AddDoc, реализующей добавление реквизитов документа в выгружаемый файл;

Блок В2 – уничтожение окна wtxt;

Блок С2 – переназначение устройства ввода вывода (файл –> экран);

Блок D2 – проверка состояния переменной-флага isOkAll (возвращаемой из процедуры AddDoc);

Блок Е2 – вывод сообщения ‘Нет передаваемых показателей’;

Блок F2 – вывод сообщения ‘Выполняется копирование’;

Блок G2 – вызов процедуры WinToDoc, осуществляющей конвертацию текстовых показателей;

Блок А3 – вывод сообщения ‘Выгрузка завершена’;

Блок В3 – выход из модуля Exp2f;

Блок С3 – начало процедуры AddDoc;

Блок D3 – проверка достижения конца файла;

Блок Е3 – вывод сообщения ‘Идет подготовка данных’;

Блок F3 – установка флажка isOkAll (по условию наличия записей);

Блок G3 – выход из процедуры AddDoc.

**4 Машинная реализация задачи**

**4.1 Характеристика технических средств**

Данный проект реализован с помощью ПЭВМ IBM PC/AT-совместимого компьютера. Данная ПЭВМ создана на основе процессора AMD AthlonXP-1700 и состоит из следующих компонентов:

* процессор;
* оперативное запоминающее устройство (ОЗУ);
* постоянное запоминающее устройство (ПЗУ);
* накопитель на жёстком магнитном диске (НЖМД, «винчестер»);
* накопитель на гибких магнитных дисках (НГМД);
* устройства ввода информации (клавиатура, «мышь»);
* устройства вывода информации (дисплей).

Процессор является основным устройством ЭВМ и предназначен для непосредственной обработки информации, которая поступает от различных внешних и внутренних устройств ПЭВМ.

Процессор AthlonXP разработки компании AMD имеет следующие технические характеристики:

* разрядность по ширине данных 64 бит
* тактовая частота 1460 MGz
* максимальный объем ОЗУ 128 Mb
* кэш-память 2х64 Kb
* быстродействие 1700 MGz (~1460)

ОЗУ предназначено для временного хранения информации и имеет объём 256 Mb.

ПЗУ предназначено для хранения неизменяемой информации, то есть программ типа BIOS (basic input-output system – базовая система ввода-вывода), которые позволяют связать компоненты компьютера в единое целое.

Процессор, модули ОЗУ и ПЗУ находятся на материнской плате компьютера. Данная материнская плата создана на основе чипсета VIA.

Накопитель на жестком диске (винчестер, HDD) предназначен для постоянного хранения информации, используемой при работе с компьютером: программ операционной системы, часто используемые пакеты прикладных программ, редакторов документов, трансляторов с языков программирования и т.д. Наличие винчестера значительно повышает удобство работы с компьютером.

Для пользователя накопители на жестком диске отличаются друг от друга, прежде всего своей емкостью. На данной модели жесткий диск имеет емкость 40,1 Гбайт. Вторая существенная для пользователя характеристика диска – время доступа к информации. Для областей применения, требующих эффективного обмена с дисками, этот показатель является исключительно важным. Данная ПЭВМ оснащена жестким диском производства Maxtor со скоростью вращения шпинделя 7200 об/мин., что обеспечивает скорость доступа порядка 10 наносекунд для внутреннего интерфейса винчестера (от головок чтения/записи до вывода на шину данных) и 18 наносекунд для внешнего (при передаче до процессора и в обратном направлении).

Накопитель на гибких магнитных дисках (НГМД) является одним из старейших устройств, входящих в базовую конфигурацию и предназначен для хранения информации небольших объёмов, её переноса между компьютерами, архивации и других целей. На данной машине имеется один НГМД 3,5” ёмкостью 1,44 Mb.

Клавиатура предназначена для ввода пользователем различной информации и управления ПЭВМ. Данный ПК оборудован 104-клавишной клавиатурой.

Манипулятор “мышь” позволяет указывать или выбирать объекты, отображаемые на экране. Компьютер, на котором разрабатывался данный дипломный проект, оснащен двухкнопочной мышью с оптико-механическим принципом работы производства фирмы Logitech.

Монитор (дисплей) компьютера IBM PC предназначен для вывода на экран текстовой и графической информации. Мониторы бывают цветными и монохромными. Они могут работать в одном из двух режимов: текстовом или графическом.

Мониторы Super VGA давно и заслуженно получили очень широкое распространение, они фактически стали стандартом для тех применений, в которых требуются графические средства приемлемого качества. Данная ПЭВМ оснащена монитором Samsung 765MB, который имеет следующие характеристики:

- тип Samsung SM765MB

- количество цветов 16,5 млн.

- разрешающая способность:

в текстовом режиме 600х800

в графическом режиме 1024х724 (рекомендуемая)

1600х1200 (максимальная)

- размер экрана по диагонали 17 ''

- частота обновления экрана 85 Гц (в режиме 1024х724)

ПЭВМ AMD Athlon XP–1700 является современным вычислительным средством, обладает высоким быстродействием, значительным объемом памяти и удобным интерфейсом.

**4.2 Математическое обеспечение**

Программа, описанная в данном дипломном проекте, была разработана на ПЭВМ Athlon XP–1700 под управлением операционной системы WindowsXP фирмы Microsoft в визуальной среде быстрой разработки приложений VisualFoxPro.

Программное обеспечение представляет собой неотъемлемую часть любой ЭВМ. С конца семидесятых годов для персональных компьютеров был разработан огромный фонд программного обеспечения, который открыл путь к их массовому применению практически во всех сферах – от игр и до решения сложных научно-технических задач.

Математическое обеспечение – это программное обеспечение машин, предназначенное для повышения эффективности их использования, для снижения ёмкости работ, связанных с эксплуатацией и облегчением труда при подготовке данных и обработки их на ЭВМ. Математическое обеспечение делится на прикладное и системное.

Системное – это интегрированные системы отладки программ и операционные системы. Операционные системы (ОС) – программы, которые делают возможным диалог человек – компьютер. На данном компьютере установлена ОС WindowsXP, которая сочетает в себе надёжность и поддержку большого количества разнообразных устройств.

Многозадачный (многопрограммный) режим позволяет разнообразным приложениям работать взаимосвязано, наличие единого буфера обмена даёт возможность переносить данные между приложениями как MS-DOS, так и Windows.

Интегрированные системы отладки программ – программы, помогающие программисту создавать, отлаживать и компилировать новые программы. На компьютере установлена интегрированная система визуальной разработки приложений Microsoft Visual FoxPro.

Visual FoxPro позволяет создавать полноценные Windows-приложения, используя максимум её ресурсов.

Широкие возможности VisualFoxPro по отладке и тестированию программ делает её незаменимой в работе над созданием приложений.

Данный дипломный проект набран и сверстан с помощью программы Word 9.0, входящей в пакет Microsoft Office 2000.

Microsoft Office – это пакет программ, разработанных фирмой Microsoft, который содержит текстовый процессор Microsoft Word, табличный процессор Microsoft Excel и другие программы. Microsoft Office – это, наверное, самый продаваемый компьютерный продукт после Windows. Microsoft Office обладает удобным интерфейсом, понятным, наверное, каждому. При этом с его помощью можно выполнять как легкие задачи, так и очень сложные. В Microsoft Office существуют такие необходимые функции, как взаимодействие приложений (вставка объекта одного приложения в другое), настройка приложений, поиск любых документов и файлов, созданных как в Microsoft Office, так и в других программах.

Microsoft Word делает написание, редактирование и печать документов, содержащих текст, настоящим удовольствием. С помощью Microsoft Word можно подготовить на профессиональном уровне письма, отчеты, таблицы, записки, рекламные листки, а также многое другое. Word можно использовать практически для всего, что содержит текст.

В дополнение к обработке собственно текста, Microsoft Word включает простые способы индивидуальной настройки символов, форматирования абзацев и встраивания в документ элементов графики. Для получения документа на профессиональном уровне можно проверить правописание, добавить графические образы, линии, границы, выделить текст тенью. Word позволяет объединить в одном документе текст, рисунки, графики, электронные таблицы и диаграммы, а затем просмотреть на экране все страницы в том виде, как они будут выглядеть на бумаге.

**4.3 Обоснование выбора языка программирования**

Опыт применения ЭВМ для построения прикладных систем обработки данных показывает, что самым эффективным инструментом в этом случае являются не языки программирования высокого уровня, а специализированные языки для создания систем управления данными. Такие средства обычно включаются в состав систем управления базами данных (СУБД), но они могут существовать и отдельно. СУБД дают возможность осуществлять непосредственное управление данными, а программистам быстро разрабатывать более совершенные программные средства их обработки.

Хотя известны попытки создания систем управления базами данных, поддерживающих сетевую модель для персональных компьютеров, в настоящее время реляционные системы лучше соответствуют их техническим возможностям и вполне удовлетворяют большинство пользователей. Скоростные характеристики этих СУБД поддерживаются специальными средствами ускоренного доступа к информации индексированием баз данных.

Собственно СУБД - это оболочка пользователя. Ввиду того, что такая среда ориентирована на немедленное удовлетворение его запросов, это всегда система-интерпретатор. Наличие в СУБД языка программирования позволяет создавать сложные системы обработки данных, ориентированные под конкретные задачи и даже под конкретного пользователя. Есть также СУБД, которые имеют только язык и не имеют оболочки пользователя, и являются системами компилирующего типа, или просто компиляторами.

Важнейшей характеристикой любой СУБД является используемый в ней тип транслятора (интерпретатор или компилятор). Программы, написанные для системы-интерпретатора, исполняются лишь в присутствии самой системы. В настоящее время скорость работы таких программ уступает скорости работы программ, сгенерированных компилятором. Бесспорным преимуществом интерпретаторов для программиста является удобство в разработке и отладке программных продуктов, а также в освоении языка.

Группа реляционных СУБД представлена на рынке программных продуктов очень широко. Однако СУБД Visual FoxPro (фирмы Microsoft) обладает достаточно высокими скоростными характеристиками и в этом отношении выделяется среди других интерпретирующих систем. Набор команд и функций, предлагаемых разработчикам программных продуктов в среде FoxPro, по мощи и гибкости отвечает современным требованиям к представлению и обработке данных. Здесь может быть реализован удобный и гибкий пользовательский интерфейс. В FoxPro поддерживаются разнообразные всплывающие и многоуровневые меню, работа с окнами и мышью, реализованы функции низкоуровневого доступа к файлам, управление цветами, настройка принтера, данные могут быть представлены в виде, похожем на электронные таблицы, и т.п. Система также средствами быстрой генерации экрана, отчетов и меню, поддерживает язык SQL, хорошо работает в сети. В пакете имеется компилятор, позволяющий придать завершенной программе вид готового коммерческого продукта в форме независимого ЕХЕ - файла.

Для функционирования СУБД Visual FoxPro необходимо наличие на машине Windows 95 и выше, на жестком диске свободного пространства около 100 MB и минимального объема памяти персонального компьютера 8 MB.

Уместно кратко перечислить возможности языка, важные для программиста:

1. поддержка и активное использование мыши.
2. язык Visual FoxPro позволяет создавать хорошо структурированные программы. Практически из всех опций команд могут быть вызваны пользовательские процедуры или функции, что придает ему исключительную гибкость. Допускаются как внешние, так и внутренние процедуры;
3. средства удобного доступа и обработки мемо-полей: теперь они могут использоваться наравне с символьными полями;
4. новые типы индексных файлов, а также возможность применения индексов не только для поиска отдельной записи, но и быстрой локализации группы записей с общим признаком (технология Rushmore);
5. установка реляционных связей между данными вида одна запись - ко многим;
6. возможность построения многоуровневых световых меню, в качестве элементов которых могут фигурировать как произвольные строки, так и элементы массивов и компоненты файлов. Допускается множественный отбор данных; меню получили статус объектов в памяти компьютера;
7. СУБД предоставляет программисту средства конфигурации и системного меню самого Visual FoxPro, что дает возможность легко настраивать его для работы в оболочке Visual FoxPro (например, при отладке) и даже интегрировать в прикладную программу;
8. широкое использование концепции окон, допускающее многооконный интерфейс, изменение пользователем размера, вида и положения окон, управление цветом;
9. Visual FoxPro располагает средствами языка SQL, дающими программисту исключительные возможности по формированию сложных запросов к базе, обрабатываемых с использованием одного из самых интеллектуальных оптимизаторов запросов;
10. специализированные команды обработки массивов;
11. удобный и мощный отладчик;
12. язык FoxPro включает средства создания интерфейса, подобного интерфейсу Windows, т.е. так называемого интерфейса, управляемого событиями.

Придание меню статуса объектов и введение Windows-подобных средств управления коренным образом меняет концепцию программирования систем обработки данных, хотя остается возможность работы и в традиционном стиле. Кроме того, FoxPro имеет в своем составе такие важные вспомогательные средства, как генераторы экранов, отчетов, меню; менеджер проектов и компилятор; драйверы принтера. Стоит сказать, что FoxPro дает разработчику возможность самому написать драйвер принтера.

Visual FoxPro обладает эффективным языком программирования прикладных информационных систем и пользуется большим успехом, особенно среди отечественных разработчиков программного обеспечения.

На вопрос - почему в качестве основного инструмента для написания программы был выбран язык Visual Fox Pro фирмы Microsoft, можно привести несколько плюсов в пользу именно его по сравнению с другими.

Во-первых, он создан в качестве продолжения линии Fox Pro, известной уже давно и снискавшей славу доступного и обладающего широким спектром возможностей языка.

Во-вторых, изготовлен Visual Fox Pro именно той фирмой, которая произвела на свет действительно ставшую общемировым стандартом (хотя и де-факто) операционную систему Windows – что обещает легкую интеграцию приложений под данной операционной системой.

В-третьих, новый язык не зря носит приставку Visual – работа с ним идет гораздо проще, нежели с более старыми версиями того же продукта. Появились богатейшие возможности по быстрому оформлению любой программы. Внешний вид – отнюдь не второстепенный фактор (а с точки зрения конечного пользователя - вообще стоит на первом месте).

В-четвертых, в Visual Fox Pro уже внедрены компоненты по созданию глобальных структур типа клиент-сервер. Хотя и не лишенные присущим фирме-изготовителю ошибок, они позволяют достаточно прочно объединить технологию Inter Base и пользовательские рабочие станции.

Учитывая возможности описанного языка программирования СУБД VisualFoxPro и информационный характер задачи дипломного проекта, необходимость обработки нескольких связанных таблиц, в качестве языка программирования для «Разработки модуля выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции» был выбран язык программирования FoxPro.

**4.4 Описание программы**

|  |  |
| --- | --- |
| Наименование программы: | «Разработка модуля выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для ГНИ» |
| Идентификатор: | Exp2f.prg |
| Язык программирования: | Microsoft VisualFoxPro 7.0 |
| Назначение программы: | Предоставление информации в ГНИ на магнитном носителе (результатов работы комплекса «Налогоплательщик ЮЛ») |
| Оборудование: | Компьютер IBM PC/AT AMD Athlon™ XP-1700 ОЗУ 256Мбайт, жесткий диск 40 Гб (Maxtor), монитор 17'' Samsung 765MB, клавиатура, мышь, принтер HP LaserJet1100 |
| Входные данные: | Базы данных: kp\_sd.dbf, kp\_spp.dbf, spp.dbf, kp\_sgr.dbf, sgr.dbf, xx\_pok.dbf, pok.dbf, consts.dbf, k029.dbf, kp\_ctrl.dbf, shapka.dbf, sptran.dbf, sputr.dbf |
| Выходные данные: | Текстовый файл специального формата |
| Краткий алгоритм: | На основании баз данных, содержащих сведения о введенных документах, модуль формирует текстовый файл специального формата, пригодный для передачи в ГНИ, а также выгружает его на магнитный носитель |

**4.5 Инструкция пользователю по работе с программой**

Доступ к режиму выгрузки документов на магнитный носитель осуществляется выбором пункта «Выгрузка» в главной форме приложения.

Данный режим предназначен для копирования введенных документов на магнитный носитель.

Для выгрузки данных на магнитный носитель предназначены следующие пункты меню:

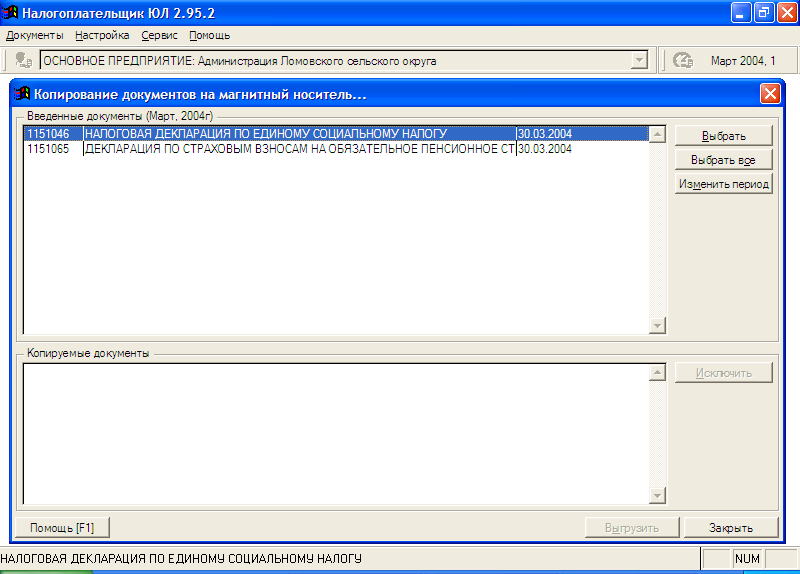
“Копирование на магнитный носитель” -выгрузка данных налоговой и бухгалтерской отчетности по внутреннему формату RNAL\_N?

“Копирование на магнитный носитель по формату 1.00” -выгрузка данных налоговой и бухгалтерской отчетности по унифицированному формату версии 1.00,

“Копирование на магнитный носитель по формату 2.00/2.01/3.00” - выгрузка данных налоговой и бухгалтерской отчетности по унифицированным форматам версий 2.00, 2.01, 3.00.

“Декларации об объемах производства и оборота алк.прод.” -выгрузка документов “Сводная декларация об объемах производства и оборота этилового спирта, алкогольной и спиртосодержащей продукции” и “Отчет об использовании специальных марок организацией - производителем алкогольной продукции и о начислении и уплате акцизов на алкогольную продукцию”,

“Выгрузка в XML” - выгрузка данных налоговой и бухгалтерской отчетности в формате XML.



Назначение функциональных кнопок:

**< Выбрать >** - выбрать документ для копирования,

**< Выбрать все >** - выбрать все документы за установленный отчетный период,

**< Исключить >** - исключить документ из числа копируемых,

**< Продолжить >** - подготовить файл с отчетными документами для копирования на магнитный носитель,

**< Помощь >** - помощь по выбору документов,

**< Закрыть >** - выход из режима.

В списке введенных документов отображаются: код документа, номер уточненного расчета, наименование документа.

В списке копируемых документов отображаются: код документа, номер уточненного расчета, отчетный период документа, наименование документа.

Отчетный период записан в виде:

ГГГГ,ПНН

где ГГГГ – отчетный год,

П–период представления документа (К-квартал, Г -год)

НН – номер периода.

Например 2004,К2 (второй квартал 2004 года).

Для выбора документа необходимо установить указатель списка введенных документов на нужный документ и нажать кнопку **<Выбрать>**, при этом документ появится в списке копируемых документов.

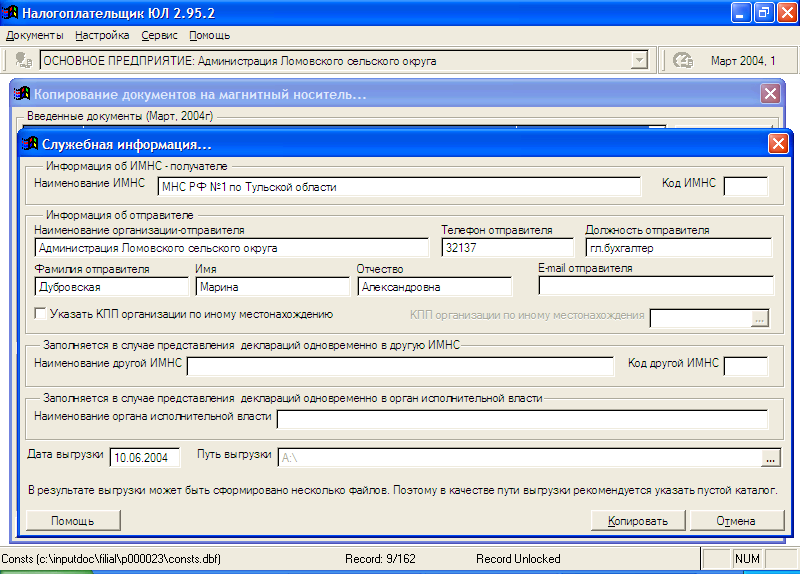
Для выбора всех документов нажмите кнопку **<Выбрать все>.**

Для исключения документа из числа копируемых необходимо установить указатель списка копируемых документов на нужный документ и нажать кнопку **< Исключить >**.

Есть возможность передавать в одном файле документы за разные отчетные периоды, для этого необходимо выбрать нужный отчетный период в этом же диалоговом окне.

При нажатии на кнопку **< Продолжить >** появляется окно “Служебная часть”.

В данном режиме необходимо указать информацию, которая будет включена в выгружаемый файл.



Назначение функциональных кнопок:

**< Копировать >** - копировать файл на магнитный носитель,

**< Помощь >** - помощь по вводу служебной части,

**< Отмена >** - отмена копирования.

**5 Контрольный пример**

В качестве входной информации при выполнении программы Exp2f.prg используются базы данных, описание которых дано в пункте 3.3 в таблице 2.

Входная информация дана в приложении А.

На основании разработанного алгоритма, представленного в виде блок-схемы в пункте 4.1. и описания в пункте 4.2. разработан модуль выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции, листинг которого показан в приложении В.

В результате работы программы формируется выходная информация, описание которой дано в пункте 3.4. в таблице 2, структура и информация выходного текстового файла дана в приложении Б.

Полученные выходные данные доказывают работоспособность программы и являются контрольным примером.

**6 Экономическое обоснование программы**

Применение персональных компьютеров на различных предприятиях и учреждениях позволяет совершенно по-новому организовать процесс решения множества экономических задач, в том числе и задачу выгрузки данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции. Это приводит к уменьшению административно-управленческих расходов, повышению производительности труда, сокращению сроков выдачи документов, что в свою очередь влияет на эффективность работы предприятия по расчетам с органами налоговой инспекции. На решение этой задачи без применения средств вычислительной техники расчетный отдел и отдел бухгалтерского учета затрачивают массу дефицитного времени, а компьютер призван обеспечить разгрузку этих подразделений от рутинной работы.

Входной информацией для этой задачи являются базы данных, хранящие информацию о введенных документах, получаемых в процессе работы любого предприятия или организации.

Выходная информация представляет собой текстовый файл специального формата, который подлежит передаче в государственную налоговую инспекцию.

Экономическая эффективность – это оценка результатов, выполнение различных действий, направленных на изменение способа, скорости и качества обслуживания клиентов.

Экономический эффект – это прежде всего экономия денежных средств предприятия, которая может быть достигнута за счет:

высвобождения штатных сотрудников для выполнения других заданий;

снижения затрат на транспортировку информации между подразделением и аппаратом;

снижение риска пень и штрафных санкций со стороны налоговой инспекции, имеющих место быть в результате несвоевременных выплат и предоставления отчётности.

Наиболее распространенным методом определения экономической эффективности является метод сравнения. Его сущность заключается в сопоставлении трудоемкости и себестоимости выполненных работ с аналогичными показателями предыдущих периодов, то есть экономическую эффективность определяют по совокупности высвобож-денного времени, работников, совершенствованию способов обработки информации путём применения более современных методов обработки.

Выделяются два вида показателей эффективности: прямая (непосредственно снижает трудовые и стоимостные затраты) и косвенная (косвенно улучшает учет за счет усовершенствований в работе сотрудников бухгалтерии).

Основным показателем прямой эффективности является снижение стоимостных и трудовых затрат на расчётные работы. В основе выявления прямой эффективности лежит принцип сопоставления затрат при различных способах решения.

Показателями прямой экономической эффективности являются:

абсолютное снижение трудовых и стоимостных затрат, показывающих, на сколько времени и на какую сумму снизились затраты;

относительное снижение трудовых и стоимостных затрат, показывающее, какая часть затрат при предполагаемом варианте экономится;

индекс роста производительности труда, показывает, во сколько раз произошло изменение.

Косвенная эффективность имеет скрытую форму и проявляется в улучшении качества управления объектом, может быть определена экспертным путем.

Косвенная эффективность позволяет оценить влияние задачи на различные стороны деятельности через качественные факторы.

Из-за отсутствия единой методики расчета прямой эффективности программных продуктов, внедряемых на персональном компьютере, в данном дипломном проекте рассматривается косвенная эффективность.

Косвенная эффективность автоматизации задачи «Выгрузка данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции» заключается в следующем:

внедрение безбумажной технологии обработки информации сильно сокращает расходы на покупку бумаги и других расходных материалов;

использование разработанной программы позволяет упорядочить большие объёмы информации для удобного использования;

возможность оперативного контроля за достоверностью информации, уменьшается число возможных ошибок, при генерировании производных данных;

сразу после ввода, данные могут участвовать в различных операциях;

возможность быстрого получения необходимых отчетов;

необходимая информация может быть быстро найдена (в том числе с помощью контекстного поиска), поиск которой в обычной папке с документами затруднен;

существенная экономия времени и людских ресурсов при выполнении операций, связанных с обработкой информации;

позволяет быстро и в наиболее подходящем для конкретного человека виде просмотреть данные, что повышает их восприятие.

Все изложенные выше преимущества автоматизации задачи выгрузки данных в текстовом формате на сегодняшний день оценены не только рядовыми пользователями, но и властными структурами, особенно это касается возможности контроля и поиска информации. ГНИ заинтересована иметь единую, электронную базу по всем лицам – налогоплательщикам, так как наличие такой базы существенно усложняет процесс уклонения от уплаты налогов (сокрытие, занижение доходов, повторное использование одноразовых льгот и т.д.)

Поэтому 15 июня 1998 года была принята очередная редакция инструкции государственной налоговой службы Российской Федерации №35 от 29 июня 1995года. Согласно которой, с 01 марта 1999 года все предприятия с численностью работающих свыше 100 человек обязаны предоставлять данные о доходах своих работников в налоговую инспекцию на магнитных носителях, причем в строго оговоренном формате.

Перечисленные факторы эффективности автоматизации задачи «Выгрузка данных в текстовом формате комплекса «Налогоплательщик ЮЛ» для государственной налоговой инспекции» приводят к уменьшению расходов на обработку экономической информации, повышению производительности труда, что в свою очередь влияет на эффективность работы и является основанием для анализа деятельности.

Косвенный эффект позволяет оценить влияние задачи на различные стороны деятельности данного объекта. Проявляется он через качественные факторы.

Режимы работы выбираются с таким расчетом, чтобы в своей совокупности они обеспечивали максимальную эффективность задачи.

При этом ускоряется не только процесс поиска и обработки данных, но что немаловажно, повышается контроль за качеством выполненной работы.

Внедрение новых информационных технологий существенно влияет на характер и способы обслуживания. Таким образом, при наименьших затратах труда и времени, осуществляется оперативный и безошибочный учет и поиск информации. Из этого следует целесообразность внедрения данной задачи на ПЭВМ.

**Список литературы**

1. Автоматизированное рабочее место в системе управления предприятием. Сборник научных трудов, - Ленинград, 1989.
2. В.В.Шураков. Автоматизированное рабочее место для статической обработки данных, 1990.
3. И.Л.Кантарь. Автоматизированные рабочие места управленческого аппарата, 1990.
4. Данилевский Ю.Г., Петухов ИА., Шибанов B.C. Информационная технология в промышленности. Л.: Машиностроение, 1988, 284с.
5. П.И. Камышанов. Практическое пособие по бухгалтерскому учету:- Издание третье, дополненное и переработанное - Москва: ООО «МЕДпресс»; - Элиста: АПП «Джангар», 1998. – 528 с.
6. А.А. Попов Программирование в среде СУБД FoxPro 2.0. Построение систем обработки данных. - М.: Издательство Март, 1996--352с.: илл.
7. Глушаков С.В., Ломотько Д.В.; Базы данных. - Харьков: Издательство «Фолио», 2001
8. Нортон П.; Персональный компьютер. Аппаратно-программная реализация. Книга 1. - М: Издательство «АСТ», 1999

**Приложение А**

**(рекомендуемое)**

**Входная информация**

Structure for database: C:\OF\KP\_SPP.DBF

Number of data records: 49

Date of last update : 04/14/04

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 KODF Character 2

2 KODP Numeric 2

3 NAIM Character 35

4 WDATA Date 8

5 PRN Numeric 1

\*\* Total \*\* 49

Structure for database: C:\OF\SPP.DBF

Number of data records: 69

Date of last update : 04/14/04

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 KODF Character 2

2 KODP Numeric 2

3 NAIM Character 35

4 WDATA Date 8

5 PRN Numeric 1

\*\* Total \*\* 49

Structure for database: C:\OF\KP\_SGR.DBF

Number of data records: 347

Date of last update : 04/14/04

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 KODF Character 2

2 KODP Numeric 2

3 KODGR Numeric 2

4 RGR Numeric 2

5 NAIM Character 50

6 NAIMP Character 35

7 WDATA Date 8

8 GO5N Numeric 2

9 PP Character 1

10 CHVGR Numeric 2

11 CHRGR Numeric 2

12 GNI Numeric 1

13 GNI\_IS Numeric 1

\*\* Total \*\* 111

Structure for database: C:\OF\SGR.DBF

Number of data records: 448

Date of last update : 04/14/04

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 KODF Character 2

2 KODP Numeric 2

3 KODGR Numeric 2

4 RGR Numeric 2

5 NAIM Character 50

6 NAIMP Character 35

7 WDATA Date 8

8 GO5N Numeric 2

9 PP Character 1

10 CHVGR Numeric 2

11 CHRGR Numeric 2

12 GNI Numeric 1

13 GNI\_IS Numeric 1

\*\* Total \*\* 111

Structure for database: C:\OF\K029.DBF

Number of data records: 51

Date of last update : 04/16/04

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 C257 Character 4

2 C258 Character 15

3 K Numeric 10

\*\* Total \*\* 30

Structure for database: C:\OF\SPUTR.DBF

Number of data records: 2

Date of last update : 05/22/02

Code Page : 866

Field Field Name Type Width Dec Index Collate

1 RAZDEL Numeric 3

2 TAXCODE Character 9

3 NDAT Date 8

4 PRUT Character 1

5 PARAG Numeric 2

\*\* Total \*\* 24

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ KP\_SPP.DBF**

Record# KODF KODP NAIM WDATA PRN

1 sy 1 Сведения об адвокате 01/01/01 0

2 sy 2 Данные о платежах 01/01/01 0

3 sp 1 Справка 01/01/01 0

4 sp 2 Расчет авансовых платежей 01/01/01 0

5 sp 3 Справочно (не подл.налогообложению) 01/01/01 0

6 sk 1 Налоговая декларация по единому соц 01/01/01 0

7 sk 2 Расчет суммы налога 01/01/01 0

8 sk 3 Справочно 01/01/01 0

9 sl 1 II.Расчет для заполн. стр.300 и 400 01/01/01 0

10 sm 1 III.Расчет для заполн.стр.300 и 400 01/01/01 0

11 qc 1 Расчет 01/01/02 0

12 qg 1 II.Расчет для заполн. с.0300 и 0400 01/01/02 0

13 qg 2 Расчет усл.на право прим.регр.став. 01/01/02 0

14 qb 1 Сведения о налогоплательщике 01/01/02 0

15 qb 2 Налоговая декларация 01/01/02 0

16 qf 2 Расчет по страх.взносам 01/01/03 0

17 qf 3 Уплата страх.взносов на ОПС 01/01/03 0

18 qf 1 Сведения о налогоплательщике 01/01/03 0

19 qd 1 Расчет для заполнения строки 0300 01/01/03 0

20 qd 2 Расчет условия на право прим.ставок 01/01/03 0

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ SPP.DBF**

Record# KODF KODP NAIM WDATA PRN

1 qf 2 Расчет по страх.взносам 01/01/02 0

2 qf 3 Справочно 01/01/02 0

3 qf 1 Сведения о налогоплательщике 01/01/02 0

4 qd 2 Расчет для заполнения с.0300 и 0400 01/01/02 0

5 qd 3 Расчет условия на право прим.ставок 01/01/02 0

6 qd 1 Справочно 01/01/02 0

7 qe 1 Выплаты в пользу работников (инв.) 01/01/02 0

8 qe 2 Расчет условий на право прим.льгот 01/01/02 0

9 qe 3 Примечание в разделу II 01/01/02 0

10 qe 4 Выплаты в пользу иностран.граждан 01/01/02 0

11 ba 1 финансовые результаты 01/01/94 0

12 ba 2 испoльзование прибыли 01/01/94 0

13 ba 3 платежи в бюджет 01/01/94 0

14 ba 4 затраты и расходы 01/01/94 0

15 ba 1 финансовые результаты 07/01/95 0

16 ba 2 испoльзование прибыли 07/01/95 0

17 ba 4 платежи в бюджет 07/01/95 0

18 ba 5 затраты и расходы 07/01/95 0

19 ba 3 движение денежных средств 07/01/95 0

20 ba 1 финансовые результаты 10/01/95 0

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ KP\_SGR.DBF**

Record# KODF KODP KODGR RGR NAIM NAIMP WDATA GO5N PP CHVGR CHRGR GNI GNI\_IS

1 sy 2 1 0 Авансовые платежи по ЕСН Данные о платежах 01/01/01 2 3 0

2 sy 2 2 0 Код строки Данные о платежах 01/01/01 2 3 0

3 sy 2 3 1 ПФР (2001),федеральный бюджет Данные о платежах 01/01/01 2 3 0

4 sy 2 4 2 ФФОМС Данные о платежах 01/01/01 2 3 0

5 sy 2 5 3 ТФОМС Данные о платежах 01/01/01 2 3 2

6 sy 1 1 0 Наименование Сведения об адвокате01/01/01 2 1 0

7 sy 1 2 0 Код строки Сведения об адвокате01/01/01 2 1 0

8 sy 1 3 1 Значение Сведения об адвокате01/01/01 2 1 2

9 sr 1 1 0 Наименование 01/01/01 2 14 0

10 sr 1 2 0 Код строки 01/01/01 2 14 0

11 sr 1 3 1 Налоговая база нар.итогом ПФР 01/01/01 2 14 0

12 sr 1 5 3 Налоговая база нар.итогом ФСС РФ 01/01/01 2 14 0

13 sr 1 6 4 Начислено авансовых платежей ПФР: ставка 01/01/01 2 14 0

14 sr 1 8 6 Начислено авансовых платежей ФСС: ставка 01/01/01 2 14 0

15 sr 1 10 8 Начислено авансовых платежей ФФОМС: ставка 01/01/01 2 14 0

16 sr 1 12 10 Начислено авансовых платежей ТФОМС: ставка 01/01/01 2 14 0

17 sr 1 14 12 Численность ПФР 01/01/01 2 14 0

18 sr 1 16 14 Численность ФСС 01/01/01 2 14 2

19 sr 1 7 5 Начислено авансовых платежей ПФР: cумма 01/01/01 2 14 0

20 sr 1 9 7 Начислено авансовых платежей ФСС: ставка 01/01/01 2 14 0

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ SGR.DBF**

Record# KODF KODP KODGR RGR NAIM NAIMP WDATA GO5N PP CHVGR CHRGR GNI GNI\_IS

1 qf 2 1 0 Наименование показателя Расчет по страх.взносам 01/01/02 2 8 0

2 qf 2 2 0 Код строки Расчет по страх.взносам 01/01/02 2 8 0

3 qf 2 3 1 Для мужчин 1952г., женщин 1956г.по дан.страховат. Расчет по страх.взносам 01/01/02 2 8 0

4 qf 2 4 2 Для мужчин 1952г., женщин 1956г.по дан.НО Расчет по страх.взносам 01/01/02 2 8 1

5 qf 2 5 3 Для муж.1953-1966, жен.1957-1966 по дан.страховат. Расчет по страх.взносам 01/01/02 2 8 0

6 qf 2 6 4 Для муж.1953-1966, жен.1957-1966 по дан.НО Расчет по страх.взносам 01/01/02 2 8 1

7 qf 2 7 5 Для лиц 1967г.р. и моложе по данным страхователя Расчет по страх.взносам 01/01/02 2 8 0

8 qf 2 8 6 Для лиц 1967г.р. и моложе по данным НО Расчет по страх.взносам 01/01/02 2 8 1

9 qf 2 9 7 Всего по данным страхователя Расчет по страх.взносам 01/01/02 2 8 2

10 qf 2 10 8 Всего по данным налогового органа Расчет по страх.взносам 01/01/02 2 8 1

11 qf 3 1 0 Наименование показателя Справочно 01/01/02 2 3 0

12 qf 3 2 0 Код строки Справочно 01/01/02 2 3 0

13 qf 3 3 1 Всего Справочно 01/01/02 2 3 0

14 qf 3 4 2 В т.ч. на страховую часть трудовой пенсии Справочно 01/01/02 2 3 2

15 qf 3 5 3 В т.ч. на накопительную часть трудовой пенсии Справочно 01/01/02 2 3 0

16 qf 1 1 0 Наименование Сведения о налогоплательщике 01/01/02 2 1 0

17 qf 1 2 0 Код строки Сведения о налогоплательщике 01/01/02 2 1 0

18 qf 1 3 1 Значение/Код Сведения о налогоплательщике 01/01/02 2 1 0

19 qd 2 1 0 Наименование Расчет для заполнения с.0300 и 0400 01/01/02 2 14 0

20 qd 2 2 0 Код строки Расчет для заполнения с.0300 и 0400 01/01/02 2 14 0

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ K029.DBF**

Record# C257 C258 K

1 0373 млн.руб. 1000000

2 0374 млрд.руб. 1000000000

3 0006 м 1

4 0008 км 1000

5 0055 кв.м 1

6 0056 тыс.кв.м 1000

7 0059 га 10000

8 0061 кв.км 1000000

9 0062 тыс.га 10000000

10 0744 % 1

11 0770 кратн. 1

12 0113 куб.м 1

13 0114 тыс.куб.м 1000

14 0115 млн.куб.м 1000000

15 0162 карат 1

16 0165 тыс.кар 1000

17 0166 кг 1

18 0167 ц 100

19 0168 т 1000

20 0170 тыс.т 1000000

21 0238 лс 1

22 0796 шт 1

23 0798 тыс.шт 1000

24 0797 100 шт 100

25 0799 млн.шт 1000000

26 0120 л 1

27 0124 тыс.л 1000

28 0371 руб. 1

29 0372 тыс.руб. 1000

30 0642 Ед. 1

31 0792 чел. 1

32 0793 тыс.чел. 1000

33 0794 млн.чел. 1000000

34 0795 млрд.чел. 1000000000

35 0214 КВт 1

36 0119 тыс.дкл 1000

37 0116 дкл 1

**ИНФОРМАЦИЯ БАЗЫ ДАННЫХ SPUTR.DBF**

Record# RAZDEL TAXCODE NDAT PRUT PARAG

1 21 102011101 09/01/95 1 1

2 80 101020201 01/01/99 1 1

**Приложение Б**

**(обязательное)**

**Выходная информация**

ИдФайл:2129000988\*\*212901001200101102936

ТипИнф:ДЕКЛАРАЦИЯЮЛ

НаимОтпрЮл:Здесь должно быть наименование предприятия.

ТелОтпр:000-00-01

ДолжОтпр:

ФИООтпр:ФИО бухгалтера

КолДок:1

ВерсПрог:INPUTDOC

@@@

ИдДок:2129000988\*\*212901001200100000001

КНД:1151006

НаимФОтч:Расчет налога от фактической прибыли

ПеридОтч:2000,К4

ДатаДок:30.11.2000

ОКЕИ:383

ВарДекл:999

КолПокФОтч:18

П010000103:1230.000

П010000303:5670.000

П010000403:0.000

П010000503:0.000

П010000603:0.000

П010000703:0.000

П010000803:0.000

П010000903:0.000

П010001003:0.000

П010001103:0.000

П010001203:0.000

П010001303:01 12..2000

П010001603: . .

П010001703: 04.06.1999

П010002а03:0.000

П010002б03:0.000

П010002в03:0.000

П010002г03:0.000

КНД:0710001

НаимФОтч:Бухгалтерский баланс

ПеридОтч:2000,К4

ДатаДок:09.12.2000

ОКЕИ:384

ВарДекл:999

КолПокФОтч:17

П010011003:456.000

П010011004:0.000

П010011103:45645.000

П010011104:0.000

П010011203:65.000

П010011204:0.000

П010011303:456.000

П010011304:0.000

П010012003:45.000

П010012004:0.000

П010012103:64.000

П010012104:0.000

П010012203:456.000

П010012204:0.000

П010013003:4564.000

П010013004:0.000

П010013503:6456.000

@@@

===

**Приложение В**

**(обязательное)**

**Текст программы**

&&---получение списка существующих документов--(открытие БД осуществляет вызывающий модуль!)

wait window nowait 'Идет подготовка данных'

YearC=iif(val(sDT)>50,'19'+sDT,'20'+sDT)

FltDocC='KONTROL="1" ' && фильтр на KP\_SD

select KP\_SD

set filter to &FltDocC

go top

do while not eof()

&& определяем имя и период

do case

case KP\_SD.PER='1'

FileNameC=KP\_SD.KODF+sKW+sDT

PeriodTxtC=YearC+',М'+alltrim(str(sPERIO,2))

case KP\_SD.PER='2'

KvartC=str(int((sPERIO-1)/3)+1,1)

FileNameC=KP\_SD.KODF+KvartC+sDT

PeriodTxtC=YearC+',К'+KvartC

case KP\_SD.PER='3'

YerHlfC=iif(sPERIO>6,'2','1')

KvartC=iif(sPERIO>6,'4','2')

FileNameC=KP\_SD.KODF+YerHlfC+sDT

PeriodTxtC=YearC+',К'+KvartC

case KP\_SD.PER='4'

FileNameC=KP\_SD.KODF+sDT

PeriodTxtC=YearC+',Г'

otherwise

FileNameC=''

PeriodTxtC=''

ENDCASE

&&---------------------------------------------------------------------------------------------------------------

DO CASE

CASE kp\_sd.per='2'

nperN=int((\_sysMonthN-1)/3)+1

CASE kp\_sd.per='3'

nperN=iif(\_sysMonthN>6,2,1)

CASE kp\_sd.per='4'

nperN=1

OTHERWISE

nperN=\_sysMonthN

ENDCASE

&&-------------------------------------------------------------------------------------------------------------------

if empty(kp\_sd)

do getmper with kp\_sd.kodf,\_sysYearN,kp\_sd.per,nperN

else

do getmper with left(kp\_sd.prvlog,2),\_sysYearN,kp\_sd.per,nperN

endif

DO CASE

CASE mperiod.period='2'

KvartC=str(int((sPERIO-1)/3)+1,1)

PeriodTxtC=YearC+',К'+KvartC

CASE mperiod.period='3'

YerHlfC=iif(sPERIO>6,'2','1')

KvartC=iif(sPERIO>6,'4','2')

PeriodTxtC=YearC+',К'+KvartC

CASE mperiod.period='4'

PeriodTxtC=YearC+',Г'

OTHERWISE

PeriodTxtC=YearC+',М'+alltrim(str(sPERIO,2))

ENDCASE

&& получаем список переменных строк----------------------------------------------------------------------------

flykodC=''

SELECT 0

IF docwdataD=kp\_sd.wdata

USE (\_bases+kp\_sd.kodf+'\_POK') ALIAS pok

SET ORDER TO TAG pok

SET FILTER TO inlist(left(alltrim(kods),1),':','~')

ELSE

USE (\_bases+'POK')

SET ORDER TO TAG pok

SET FILTER TO kodf=kp\_sd.kodf ;

AND wdata=docwdataD ;

AND inlist(left(alltrim(kods),1),':','~')

ENDIF

GO TOP

DO WHILE NOT eof()

flykodC=flykodC+alltrim(str(pok.ykod,3))+';'

SKIP

ENDDO

USE IN pok

select 0

use (\_\_\_user+FileNameC) alias XXZZYY

select 0

&& получаем список уточнений------------------------------------------------------------------------------------------

select max(val(UT)) AS utn ;

from XXZZYY ;

into table (old\_vtemp+'AllUT')

select AllUT

go top

do while not eof()

&& заполняем список документов------------------------------------------------------------------------------------

if \_UseKND

KNDTxtC=iif(empty(alltrim(KP\_SD.KND)) or (alltrim(KP\_SD.KND)=='0'),'1150000',KP\_SD.KND)

else

KNDTxtC=KP\_SD.KODF

endif

select HaveDoc

append blank

replace KODF with KP\_SD.KODF, ;

UT with str(AllUt.UTn,2), ;

NAIM with KP\_SD.NAIM, ;

FileName with FileNameC, ;

SDWDATA with KP\_SD.WDATA, ;

PERIODTXT with PERIODTXTC, ;

KNDTXT with KNDTxtC, ;

wdata WITH docwdataD, ;

flykod WITH flykodC, ;

prvlog WITH kp\_sd.prvlog, ;

prd WITH kp\_sd.prd, ;

\_knd WITH kp\_sd.knd, ;

prmnog WITH kp\_sd.prmnog

select AllUt

skip

enddo

select AllUt

use

select XXZZYY

use

endif

select KP\_SD

skip

enddo

wait clear

if DocList.WDATA=DocList.SDWDATA

select ExpDcPok

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000101', ;

FieldName with 'KOMM'

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000201', ;

FieldName with 'RASSH'

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000301', ;

FieldName with 'C257N'

SELECT 0

USE (\_bases+doclist.kodf+'\_POK') alias pok

set order to tag POK

go top

do while not eof()

if not empty(KODS) and (not (alltrim(KODS)=='0'))

select kp\_sgr

set filter to KODF=DocList.KODF and KODP=POK.KODP and RGR<>0

go top

do while not eof()

scrC=subdat1(pok.twd,kp\_sgr.rgr)

IF NOT empty(scrC)

typeC=substr(scrC,1,1)

sc1N=at('(',scrC)

sc2N=at(')',scrC)

zpN=at(',',scrC)

IF zpN=0

lenN=val(substr(scrC,sc1N+1,sc2N-sc1N-1))

decN=0

ELSE

lenN=val(substr(scrC,sc1N+1,zpN-sc1N-1))

decN=val(substr(scrC,zpN+1,sc2N-zpN-1))

ENDIF

ENDIF

YKodN=POK.YKOD

KodPN=POK.KODP

KodSC=strtran(POK.KODS,':','')

KodGN=kp\_SGR.KODGR

RGRN=kp\_SGR.RGR

GNIN=iif(kp\_SGR.GNI=1,1,0)

TxtPokC='П'

TxtPokC=TxtPokC+iif(KodPN>=10,str(KodPN,2),'0'+str(KodPN,1))

TxtPokC=TxtPokC+replicate('0',5-len(alltrim(KodSC)))+alltrim(KodSC)

TxtPokC=TxtPokC+iif(KodGN>=10,str(KodGN,2),'0'+str(KodGN,1))

FieldNameC=DocList.KODF+alltrim(str(YKodN,3))+'\_'+alltrim(str(RGRN,2))

select ExpDcPok

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with TxtPokC, ;

FieldName with FieldNameC, ;

type WITH typeC, ;

len WITH lenN, ;

dec WITH decN

select kp\_SGR

skip

enddo

endif

select POK

skip

enddo

USE IN pok

else

select ExpDcPok

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000101', ;

FieldName with 'KOMM'

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000201', ;

FieldName with 'RASSH'

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with 'П000000301', ;

FieldName with 'C257N'

SELECT 0

USE (\_bases+'POK')

set order to tag POK

set filter to KODF=DocList.KODF and WDATA=DocList.WDATA

go top

do while not eof()

if not empty(KODS) and (not (alltrim(KODS)=='0'))

select SGR

set filter to KODF=DocList.KODF and KODP=POK.KODP and WDATA=DocList.WDATA and RGR<>0

go top

do while not eof()

scrC=subdat1(pok.twd,sgr.rgr)

IF NOT empty(scrC)

typeC=substr(scrC,1,1)

sc1N=at('(',scrC)

sc2N=at(')',scrC)

zpN=at(',',scrC)

IF zpN=0

lenN=val(substr(scrC,sc1N+1,sc2N-sc1N-1))

decN=0

ELSE

lenN=val(substr(scrC,sc1N+1,zpN-sc1N-1))

decN=val(substr(scrC,zpN+1,sc2N-zpN-1)) ENDIF

ELSE

IF pok.td='U'

typeC=substr(pok.td1,sgr.rgr,1)

ELSE

typeC=pok.td

ENDIF

DO CASE

CASE typeC='C'

lenN=15

decN=0

CASE typeC='D'

lenN=8

decN=0

OTHERWISE

typeC='N'

lenN=15

decN=3

ENDCASE

ENDIF

if pok.td=='U'

if len(alltrim(pok.td1))>=SGR.RGR

TypeC=substr(pok.td1,SGR.RGR,1)

else

TypeC='N'

endif

else

TypeC=pok.td

endif

YKodN=POK.YKOD

KodPN=POK.KODP

KodSC=strtran(POK.KODS,':','')

KodGN=SGR.KODGR

RGRN=SGR.RGR

GNIN=iif(SGR.GNI=1,1,0)

TxtPokC='П'

TxtPokC=TxtPokC+iif(KodPN>=10,str(KodPN,2),'0'+str(KodPN,1))

TxtPokC=TxtPokC+replicate('0',5-len(alltrim(KodSC)))+alltrim(KodSC)

TxtPokC=TxtPokC+iif(KodGN>=10,str(KodGN,2),'0'+str(KodGN,1))

FieldNameC=DocList.KODF+alltrim(str(YKodN,3))+'\_'+alltrim(str(RGRN,2))

select ExpDcPok

append blank

replace KODF with DocList.KODF, ;

WDATA with DocList.WDATA, ;

TxtPok with TxtPokC, ;

FieldName with FieldNameC, ;

type WITH typeC, ;

len WITH lenN, ;

dec WITH decN

select SGR

skip

enddo

endif

select POK

skip

enddo

USE IN pok

endif

&&-----------------------------------------------------------------

aPath=alltrim(upper(bPath))

if right(aPath,1)<>'\'

aPath=aPath+'\'

endif

aDrv=left(aPath,1)

if !cdrv(aDrv)

if aDrv='A' .or. aDrv='B'

do while !cdrv(aDrv)

if MESSAGEBOX('Вставьте диск в дисковод '+aDrv+':',64+1)<>1

RETURN .F.

endif

enddo

else

MESSAGEBOX('Диск '+aDrv+': не существует.', 16)

RETURN .F.

endif

endif

isOkPath=.t.

PRIVATE olderrC

olderrC=ON('ERROR')

on error isOkPath=.f.

SetDefault(aPath) &&&&set default to &aPath

ON ERROR &olderrC

SetDefault(root\_dir) &&&&set default to &root\_dir

if !isOkPath

MESSAGEBOX('Неправильный путь к файлу.', 16)

RETURN .F.

endif

&& формирование текстового файла

wait window nowait 'Идет подготовка файла'

define window wtxt from 24,79 to 24,79 none

activate window wtxt noshow

set alternate to (old\_vtemp+aFileName)

set alternate on

set console off

?? 'ИдФайл:'+alltrim(sIdOtp)

? 'ТипИнф:'+alltrim(sType)

? 'НаимОтпрЮл:'+alltrim(sDan)

? 'ТелОтпр:'+alltrim(sTel)

? 'ДолжнОтпр:'+alltrim(sDol)

? 'ФИООтпр:'+alltrim(sFIO)

? 'КолДок:1'

? 'ВерсПрог:'+alltrim(sVer)

? '@@@'

? 'ИдДок:'+alltrim(sIdDoc)

isOkAll=.f.

do AddDoc

? '@@@'

? '==='

release window wtxt

set alternate to

set console on

wait clear

if !isOkAll

MESSAGEBOX('Нет передаваемых показателей во всех передаваемых документах. Копируемый файл не создан', 16)

else

wait window nowait 'Выполняется копирование файла'

DO WinToDos IN \_bin+'oninit.prg' WITH old\_vtemp+aFileName, aPath+aFileName

wait clear

messagebox('Выгрузка завершена.', 64)

endif

clear read

SELECT 0

USE (\_bases+'NOMFILE')

LOCATE FOR year=\_sysYearN

REPLACE nomfile WITH nomfile+1

USE

return .t.

&&-----------------------------------------------------------------

procedure AddDoc

select DocList

go topdo while not eof()

wait wind nowait 'Идет подготовка данных'

select ExpDcPok

set filter to KODF=DocList.KODF and WDATA=DocList.WDATA

count to PkCntN

if PkCntN>0

isOkAll=.t.

do AddDoc01

else

MESSAGEBOX('Документ: '+DocList.NAIM + CHR(13) +;

'Уточнение: '+iif(val(DocList.UT)=0,'Основной расчет',DocList.UT+' уточнение')+;

'Нет передаваемых показателей. Документ будет пропущен.',64)

endif select DocList

skip

enddo

wait clear

return