**Учет квартплаты (АРМ бухгалтера-расчетчика)**

Введение

Краткая характеристика предприятия

Усть-Кутская перевалочно-распределительная нефтебаза является филиалом ОАО "Иркутскнефтепродукт".

Расположена в г. Усть-Куте, Иркутской обл., на берегу р. Лена, в устье реки Якурим.

Нефтебаза классифицируется по следующим признакам:

* по общей вместимости и максимальному объему одного резервуара относится к первой категории
* по функциональному назначению – перевалочно-распределительная
* по транспортным связям – смешанная, автомобильно-водно-железнодорожная
* по номенклатуре – нефтебаза общего хранения.

Предприятие имеет восемь АЗС, а также пять филиалов-нефтескладов.

Характеристика производственной деятельности

По своему функциональному назначению нефтебаза является перевалочным пунктом для отправки нефтепродуктов в северные районы Иркутской области, Республику Саха-Якутия, Хабаровский край водным путем (по р. Лена) за период навигации с 1/05 по 25/10. Наличие мощного резервуарного парка позволяет производить накопление и хранение нефтепродукта в любое время года.

Через имеющиеся восемь АЗС и пять филиалов-нефтескладов нефтебаза покрывает потребности в нефтепродуктах в некоторых районах Иркутской обл.

Маслами, смазками также обеспечиваются предприятия Братского и Усть-Илимского районов.

Предприятие имеет развитую инфраструктуру по обеспечению предприятий и организаций и жилых микрорайонов следующими видами услуг:

* установка телефонов и обеспечение телефонной связью
* передача электроэнергии через собственную подстанцию
* снабжение паром, холодной и горячей водой от собственных источников (ТЭЦ, водозабор)
* перевозка нефтепродуктов автотранспортом.

Производственные мощности и их оснащения

*Чтобы управлять организацией, необходимо определенным образом связать цели, людей и виды, выполняемой ими работ между собой. Для этого создается организационная структура управления организацией, и составляются должностные инструкции на всех членов организации, в которых оговариваются: права, ответственность и обязанности каждого.*

Схема 1

Структура управления предприятия

Схема 2

Подразделение предприятия на цеха

Непосредственная реализация нефтепродуктов осуществляется через такие основные цеха, как 1 цех, 4 цех и АЗС. Объем реализации нефтепродуктов по цехам распределяется в зависимости от сезонных условий, т.е. в летний период наибольший объем реализации нефтепродукта приходится на 1 и 4 цеха, т.к. в этих цехах производится отгрузка нефтепродуктов водным путем (танкерами).В зимний же период эти цеха производят "затарку" резервуаров на хранение нефтепродукта. Прием нефтепродукта осуществляется с вагоноцистерн через эстокады. Отгрузка производится водным путем танкерами через причалы

Цех № 1 предназначен как для хранения нефтепродукта , перевалки водным путем (танкерами), обеспечением города и филиалов темными и светлыми нефтепродуктами.

Цех № 4предназначен для хранения светлых нефтепродуктов и отправки водным путем.

Лаборатория осуществляет контроль за качеством принимаемых и отпускаемых нефтепродуктов, а также периодический контроль за состоянием нефтепродуктов в резервуарах.

Тарный производит перевалку горюче-смазочных материалов через бочкатару и поштучно (литол, циатим, тосол …).

Цех № 3, АЗС осуществляют реализацию нефтепродуктов населению и предприятиям наличным и безналичным расчетом.

Вспомогательное производство осуществляет обслуживания основных объектов нефтебазы, а также и близлежащий жилой поселок (обеспечение теплом, связью, электроэнергией при необходимости)

 *Основные производственные фонды*

Условное распределение объема реализации нефтепродукта основными цехами

Основные средства предприятия представлены различными видами, которые отражают производственную деятельность. Наибольший удельный вес имеют здания и сооружения.

В состав зданий входят производственные цеха, которые были построены в основном 25-30 лет назад и большей частью требуют капитального ремонта. Низка норма износа, проводимые переоценки значительно увеличили стоимость зданий, что не всегда соответствует истинной рыночной стоимости.

Сооружение, в основном, представлены резервуарами, которые на 90% не имеют остаточной стоимости и фактически по сроку службы подлежат списанию. Но так как сделать это не возможно, следует произвести оценку остаточной стоимости этой части основных фондов, это позволит получить дополнительные амортизационные отчисление на восстановление износа

Из общего количества резервуаров на нефтебазе 10% требуют немедленной замены, остальным необходимо проведение полного обследования, определения остаточного ресурса и ремонт.

Высокая степень износа (более 60%) характеризует такие объекты основных средств как машины и оборудование, и транспортные средства. Их удельный вес в общем объеме составляет 2%.

Практически изношенными являются передаточные устройства, имеющие остаточную стоимость 22,7 млн.руб. при восстановительной стоимости 96,2 млн. руб.,

В связи с уменьшением капитальных вложений, ввод основных фондов за 1999-2000 сократился и не оказывает влияния. За 1999 г. введено основных средств на сумму 2,9 млн. руб. Списание основных средств в 2000 г. не производилось. Это связано с тем, что полностью амортизированные основные средства эксплуатируются, а списание основных средств физически изношенных и не эксплуатирующихся, но не имеющих остаточную стоимость не разрешается.

В целом основные средства нефтебазы имеют значительную восстановительную стоимость 606,9 млн. руб., большой износ – 356,8 млн. руб.

Для определения истинной рыночной стоимости каждого объекта основных фондов необходимо провести натуральную оценку с привлечением профессиональных оценщиков. Это положительно скажется на определение налогооблагаемой базы для определения налога на имущество так и для увеличения амортизационных отчислений на ремонт и полное восстановление основных средств

Себестоимость, прибыль и рентабельность продукции

Прибыль и рентабельность представляют собой важные показатели эффективности производства.

Достижение *наибольших результатов при наименьших затратах* является непреложным законом хозяйственного деятельности.

Повышение эффективности производства означает не только рост производительности труда, фондоотдачи и снижение себестоимости продукции, но и увеличение суммы прибыли и уровня рентабельности.

На предприятии различают прибыль от реализации продукции и балансовую прибыль.

*Прибыль* представляет собой часть чистого дохода, полученного в процессе хозяйственной деятельности, который образуется после возмещения из доходных поступлений (выручки) затрат на себестоимость продукции.

Сумма прибыли изменяется под влиянием *интенсивных* и *экстенсивных* факторов.

К числу интенсивных факторов изменение суммы прибыли относится улучшение использования производственных фондов, рост объема производства без увеличения численности работающих, повышение производительности труда и снижение себестоимости.

*Рентабельность* - важный обобщающий показатель работы предприятия. В ней находит отражение не только величена прибыли и производственных фондов, но и производительность труда, фондоотдачи, скорости оборачиваемости оборотных средств и т.д.

Различают три вида рентабельности:

* общая рентабельность - отношение суммы балансовой прибыли к среднегодовой стоимости основных производственных фондов и нормируемых оборотных средств, не про кредитованных банком;
* ⮚расчетная рентабельность - отношение суммы расчетной прибыли к среднегодовой стоимости основных производственных фондов и нормируемых оборотных средств за вычетом тех основных фондов, по которым не взимается плата в бюджет, и нормируемых оборотных средств, про кредитованных банком. Для исчисления суммы расчетной прибыли балансовая прибыль уменьшается на сумму платежей в бюджет за производственные основные фонды и нормируемые оборотные средства и на сумму уплаченных процентов за пользование банковским кредитом;
* ⮚рентабельность по отношению к расходам - отношение суммы прибыли к расходам как в целом по анализируемому предприятию, так и по отдельным видам деятельности или видам работ.

 Планирование производственно-хозяйственной деятельности предприятия

План производства и реализации продукции

Исходным элементом организации производственного процесса является качественно разработанный, своевременно доведенный до цехов и подразделений предприятия планы и плановые задания.

Планирование капитального строительства и капитального ремонта основных фондов

Основной задачей планирования капитального строительства являются увеличение и обновление производственных мощностей и основных фондов. Эта задача должна решаться на современной научно-технической основе в объемах, необходимых для обеспечения прироста продукции и услуг, повышения их качества, роста эффективности производства.

При планировании капитального строительства следует обеспечить сбалансированность лимитов капитальных вложений и строительно-монтажных работ с финансовыми и материальными ресурсами.

К основным показателям плана капитальных вложений относится:

* прирост производственных мощностей за счет технического перевооружения и реконструкции
* ввод в действие производственных мощностей, отдельных объектов производственного назначения за счет расширения действующих и строительства новых зданий и сооружений
* ввод в действие объектов охраны природы
* ввод в действие основных производственных и непроизводственных фондов

Лимит - ресурсный показатель, определяющий предельную величину капитальных вложений, в том числе на строительно-монтажные работы, на запланированный ввод в действие мощностей, объектов и создание нормативных заделов. Заказчики обязаны укладываться в соответствующие лимиты капитальных вложений.

 Предпосылки для написания приложения

Главным фактором для разработки своего приложения явилось предложение моей супруги, работающей бухгалтером на этом предприятии и выполняющей расчет квартплаты. Усть-Кутская нефтебаза имеет на своем балансе Жилищное хозяйство общей площадью – 22735 кв.метров и количеством проживающих – 14 тысяч 420 человек. Расчет квартплаты производится в бухгалтерии по данным ЖКО. При данном расчете в основном фиксируются те жильцы, с которых взимают и начисляют квартплату за предоставляемые услуги. Моей целью, при написании приложения, было наиболее полно отразить всех проживающих в СУБД и подходящей больше непосредственно для ЖКО как расчетно-информационная база всех проживающих.

Краткая характеристика системы

Для осуществления планирования деятельности любой фирмы на любом этапе работы осуществляются некоторые расчеты для автоматизации которых удобно и рационально применять вычислительную технику. В своей дипломной работе я решил разработать приложение по учету платы за квартиру.

В данном приложении решаются следующие задачи:

Заполнение и хранение информации о всех жильцах. Данная информация упорядочена и выводится на экран, для просмотра или редактирования в зависимости от предполагаемого действия пользователя.

Возможность выборочного начисления на предоставляемые услуги, а также возможно (если это необходимо) установить процент начисления на каждую услугу отдельно для конкретного квартиросъемщика (например в какой то период в квартире никто не проживает (отпуск ,командировка и т.д.)) следовательно за какие-то услуги плата не начисляется или начисляется в меньшем размере (10%,20%,…), или наоборот.

Возможность определения льготы для одного или группы квартиро-съемщиков и процентной ставке для каждой льготы.

Формирование и печать необходимых данных для отчетности и анализа, а также печать квитанции об произведенных начислениях, задолженности по оплате и остатке денежных средств для конкретного жильца.

Возможность обращения к справочным экранам, описывающими возмож-ные действия в данном месте системы.

Предусмотрен входной и выходной контроль для записей, которые являются обязательно уникальными (не повторяемость).

Описание предметной области

Организация работы с данными является одним из основных этапов создания прикладной программы. Именно от этого первого этапа, действительно очень важного и многими недооцениваемого, зачастую зависит создание всей системы автоматизированной обработки данных и успех ее эксплуатации у конечного пользователя.

Разработка программы для учета начислений оплаты за пользование услугами, предоставляемых Жилищно-Коммунальным хозяйством Усть-Кутской нефтебазы.

При расчете квартплаты должны быть предусмотрены начисления по льготам, такие как если гражданин является участником ВОВ, работником тыла, ветеран труда, инвалиды и т.д., пользователь может использовать любые виды льготы и ставки по ним, согласно действующим для данной организации. В процессе работы, можно изменять, дополнять льготы для конкретного жильца или группы жильцов, имеющих одинаковый код и ставок по ней. Также предусмотрено начисление льгот на определенный период, в этом случае пользователь выбирает дату, с которой начинается расчет льгот, для конкретного лица, и дату когда она заканчивается.

 Для каждого жильца, с которого берется квартплата, предусмотреть изменение стоимости услуг, в зависимости от сложившейся ситуации на данный момент времени (отпуск, командировка и т.д.) или от типа жилья (неблагоустроенная квартира, привозная вода и т.д.), как в меньшую, так и в большую сторону, по умолчанию начисляются нормативные ставки, действующие в данной организации.

 После производства всех расчетов, определяется общая сумма к оплате, суммы начислений по всем видам услуг и суммы начислений по льготам по каждому квартиросъемщику.

 По окончании отчетного периода (1 месяц) формируется отчеты по видам оплаты и категориям проживающих, с возможностью их корректировки вручную и распечаткой в любое время.

Определение задач

Регистрация всех проживающих

Определение (указание), используемых услуг

Определение (выбор) льготы и ставок по ней

Определение (изменение) периода начисления льготы (необязательное условие)

Определение норм оплаты за каждую услугу (нормативы)

Определение норм оплаты за каждую услугу, для отдельного жильца или в целом для жильцов, проживающих в одном доме (необязательное условие)

Расчет квартплаты

Расчет льгот

Определение сумм к оплате

Формирование и печать квитанции об оплате для конкретного плательщика

Учет платежей квартиросъемщиков (дата и сумма платежа, остаток или задолженность)

Просмотр, Дополнение и Изменение информации обо всех категориях проживающих

Ежемесячная подготовка отчетов по различным начислениям

Редактирование и печать отчетов (в любое время и в любом количестве)

Возможность обратиться к файлам помощи (F1)

Возможность поиска

Ответы на эти вопросы позволяют подойти к главному в постановке задачи – построении инфологической модели приложения. В простейшем примере такая модель может быть отображена в виде взаимосвязей между компонентами и процессами приложения, как показано на рисунке 2.1. в практике проектирования такие схемы получили название ER-диаграмм (Entity-relationship diagram (ERD) – диаграмма “Сущность-связь”)

Максимально формализованное описание задачи теперь будет выглядеть следующим образом.

*Наименование задачи:*

Автоматизация работы бухгалтера по расчету квартплаты.

*Цель работы бухгалтера:*

Введение учета и расчетов с населением за предоставляемые коммунальные услуги.

*Функции бухгалтера:*

Заключение договоров и регистрация клиентов в Базе Данных.

Ведение информации по льготам и ставкам по ним.

Работа с клиентами : установление льготы, установление тарифных ставок отличных от нормативных по различным причинам, разъяснительная работа и т.д..

Ведение расчетов за используемые услуги (выписка квитанции об оплате).

*Бизнес - правила:*

Ввод информации на основе документов.

Требования к системе

Программа может работать под управлением операционных систем DOS или WINDOWS, при наличии пакета FoxPro 2.6 for DOS.

*Перечень вводимой информации:*

Фамилия, имя и отчество жильца;

Табельный номер;

Год рождения

Адрес;

Квадратные метры;

Телефон;

При наличии льготы – описание льготы, код, период;

Выбор плательщика, указанного в ордере

Нормативные ставки на услуги

Тарифные ставки и описание тарифа

*Перечень печатных отчетов:*

Списки по начислению

Списки льготников

Списки с показаниями счетчика

Оплата электроэнергии

Сальдо по квартиросъемщикам

Свод по квартплате за месяц

Квитанция об оплате

Проектирование базы данных

Существует очень простое понятие базы данных как большого по объему хранилища, в которое организация помещает все используемые ею данные и из которого различные пользователи могут их получать, используя различные приложения. Такая единая база данных представляется идеальным вариантом, хотя на практике это решение по различным причинам трудно достижимо. Поэтому чаще всего под базой данных понимают любой набор хранящихся в компьютере данных.

*В основу проектирования БД должны быть положены представления конечных пользователей организации – концептуальные требования к системе. Потому что, именно конечный пользователь в своей работе принимает решения с учетом получаемой в результате доступа к базе данных информации. Следовательно, от оперативности и качества этой информации будет зависеть эффективность работы организации. Данные, помещаемые в базу данных, также предоставляет конечный пользователь.*

*При рассмотрении требований конечных пользователей принято во внимание следующее:*

*База данных должна удовлетворять актуальным информационным потребностям организации. Получаемая информация должна по структуре и содержанию соответствовать решаемым задачам.*

*База данных должна обеспечивать получение требуемых данных за приемлемое время, то есть отвечать заданным требованиям производительности.*

*База данных должна удовлетворять выявленным и вновь возникающим требованиям конечных пользователей.*

*Загруженные в базу данных корректные данные должны оставаться корректными.*

*Данные до включения в базу данных должны проверяться на достоверность.*

*Доступ к данным, размещаемым в базе данных, должны иметь только лица с соответствующими полномочиями.*

*На основании вышеизложенного рассмотрим этапы проектирования баз данных, которые должны обеспечить необходимую независимость данных и выполнение эксплуатационных требований (пожеланий пользователей).*

Этап 1. Определение сущностей

*Исходя из задачи, описанной выше, выделим следующие сущности:*

Жильцы – БД, содержащая информацию о жильцах (*RABOT.DBF*)

Начисления - БД, содержащая информацию о начислениях (*OPLATA.DBF*)

Льготы - БД, содержащая информацию о льготах (*LGOT.DBF*)

Ставки - БД, содержащая информацию о ставках для конкретного жильца и (или) о жильцах, проживающих в одном доме (*TABLE\_R.DBF*)

Тарифы - БД, содержащая информацию описания тарифов и ставок по ним (*TARIF.DBF*)

Первичный ключ – это атрибут (или группа атрибутов), которые единственным образом идентифицируют каждую строку в таблице.

Альтернативный ключ – это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр объекта.

Для каждой сущности определим атрибуты, которые будем хранить в БД.

Атрибуты, включаемые в состав БД, приведены в таблице 2.1. информационная модель после третьего этапа проектирования приведена на рис. 2.13

Таблица 2.1. Атрибуты и первичные ключи сущностей информационной модели

|  |  |  |
| --- | --- | --- |
| Сущность | Первичный ключ | Атрибуты |
| ЖИЛЬЦЫ | Табельный номерУлицаДомКвартира | Табельный номерУлицаДомКвартираФ.И.ОНомер телефонаКвадратные метрыЛьготаКод льготы№ удостоверения (по льготе)Дата, с которой будет начисляться льготаДата окончания начислений по льготеСтарое значение счетчикаНовое значение счетчикаОпределение плательщикаКоличество проживающих (в 1-й квартире)Дата оплаты за предоставленные услугиОстаток денежных средствВвод оплатыНачисления квартплаты (Есть\Нет)Начисления радио (Есть\Нет)Начисления телефона (Есть\Нет)Начисления горячей воды (Есть\Нет)Начисления холодной воды (Есть\Нет)Начисления коммунальн. услуг (Есть\Нет)Начисления отопления (Есть\Нет)Начисления электроэнергии (Есть\Нет) |
| ЛЬГОТА | Код льготы | Код льготыОписание льготыПроцентная ставка по квартплатеПроцентная ставка по радиоПроцентная ставка по телефонуПроцентная ставка по горячей водеПроцентная ставка по холодной водеПроцентная ставка по ком. услугамПроцентная ставка по отоплению |
| ТАРИФ |  | Описание тарифаСтавка оплатыПорядок начисления (на 1-го чел.\ на 1 м2)Выбор: на 1-го человека или на 1 м2 |
| Сущность | Первичный ключ | Атрибуты |
| СТАВКИ | Табельный номер | Табельный номерОписание льготыСтавка по квартплатеСтавка по радиоСтавка по телефонуСтавка по горячей водеСтавка по холодной водеСтавка по коммунальным услугамСтавка по отоплениюСтавка по электроэнергии |
| НАЧИСЛЕНИЯ | Табельный номерУлицаДомКвартира | Табельный номерУлицаДомКвартираКвадратные метрыКод льготыВыбор плательщикаКоличество проживающихЛьгота (Да\Нет)Начисления квартплаты (Да\Нет)Начисления радио (Да\Нет)Начисления телефона (Да\Нет)Начисления гор. Воды (Да\Нет)Начисления хол. Воды (Да\Нет)Начисления ком. Услуг (Да\Нет)Начисления эл. Энергии (Да\Нет)Начисления отопления (Да\Нет)Сумма начисления квартплатыСумма начисления телефонаСумма начисления радиоСумма начисления ком.услугСумма начисления отопленияСумма начисления эл\энергииСумма начисления гор.водыСумма начисления хол.водыНач. квартплаты (по льготе) на 1-го чел.Нач.телефона (по льготе) на 1-го чел.Нач. радио (по льготе) на 1-го чел.Нач. ком.услуг (по льготе) на 1-го чел.Нач. отопления (по льготе) на 1-го чел.Нач. эл\энергии (по льготе) на 1-го чел.Нач. гор.воды (по льготе) на 1-го чел. Нач. хол.воды (по льготе) на 1-го чел.Сумма нач. квартплаты по льготеСумма нач. гор. Воды по льготеСумма нач. хол. Воды по льготеСумма нач. ком. Услуг по льготеСумма нач. отопления по льготеСумма нач. телефона по льготеСумма нач. радио по льготеИтог сумм начислений по льготеИтог сумм начисленийИтоговая сумма к оплате |

Этап 4. Физическое описание модели

На этом этапе составим базы данных, которые будут в дальнейшем реализовываться в СУБД. Назначения имен БД и их атрибутов отражены в табл. 2.2.

Таблица 2.2. Структура, используемых Баз Данных.RABOT.DBF (Жильцы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | примечание |
| TAB | N | 4 | 0 | Табельный номер |
| FAM | C | 40 | - | Ф.И.О |
| TEL | N | 8 | 0 | Номер телефона |
| YL | C | 15 | - | Улица |
| DOM | С | 3 | 0 | Дом |
| KW\_RA | С | 3 | 0 | Квартира |
| KV\_M | N | 6 | 2 | Квадратные метры |
| N\_LG | N | 1 | 0 | Код льготы |
| N\_YD | С | 9 | 0 | Номер удостоверения (по льготе) |
| DAT\_C | D | 8 | - | Дата, с которой будет начисляться льгота |
| DAT\_PO | D | 8 | - | Дата окончания начислений по льготе |
| ELEC | N | 6 | 0 | Старое значение счетчика |
| ELEC1 | N | 6 | 0 | Новое значение счетчика |
| OR\_R | N | 1 | 0 | Определение плательщика |
| KOL\_VO | N | 2 | 0 | Количество проживающих |
| D\_OPL | D | 8 | - | Дата оплаты |
| OST\_K | N | 8 | 2 | Остаток денежных средств |
| OPL\_TA | N | 8 | 2 | Ввод оплаты |
| LGOT | L | 1 | - | Льгота (Да\Нет) |
| RAD\_L | L | 1 | - | Начисления радио (Да\Нет) |
| TEL\_L | L | 1 | - | Начисления телефона (Да\Нет) |
| KW\_L | L | 1 | - | Начисления квартплаты (Да\Нет) |
| G\_W\_L | L | 1 | - | Начисления гор. воды (Да\Нет) |
| X\_W\_L | L | 1 | - | Начисления хол. воды (Да\Нет) |
| K\_YS\_L | L | 1 | - | Начисления ком. услуг (Да\Нет) |
| EL\_C\_L | L | 1 | - | Начисления эл. энергии (Да\Нет) |
| OTOP\_L | L | 1 | - | Начисления отопления (Да\Нет) |

OPLATA.DBF (Начисления)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | Примечание |
| TAB | N | 4 | 0 | Табельный номер |
| YL | C | 15 | - | Улица |
| DOM | С | 3 | 0 | Дом |
| KW\_RA | С | 3 | 0 | Квартира |
| KV\_M | N | 6 | 2 | Квадратные метры |
| N\_LG | N | 1 | 0 | Код льготы |
| OR\_R | N | 1 | 0 | Выбор плательщика |
| KOL\_VO | N | 2 | 0 | Количество проживающих |
| DAT\_C | D | 8 | - | Дата, с которой будет начисляться льгота |
| DAT\_PO | D | 8 | - | Дата окончания начислений по льготе |
| ELEC | N | 6 | 0 | Старое значение счетчика |
| ELEC1 | N | 6 | 0 | Новое значение счетчика |
| TEL\_L | L | 1 | - | Начисления телефона (Да\Нет) |
| G\_W\_L | L | 1 | - | Начисления гор. Воды (Да\Нет) |
| X\_W\_L | L | 1 | - | Начисления хол. Воды (Да\Нет) |
| K\_YS\_L | L | 1 | - | Начисления ком. Услуг (Да\Нет) |
| EL\_C\_L | L | 1 | - | Начисления эл. Энергии (Да\Нет) |
| OTOPL\_L | L | 1 | - | Начисления отопления (Да\Нет) |
| KW\_PL | N | 8 | 2 | Сумма начисления квартплаты |
| TEL\_R | N | 8 | 2 | Сумма начисления телефона |
| RAD\_R | N | 8 | 2 | Сумма начисления радио |
| K\_YSL | N | 8 | 2 | Сумма начисления ком.услуг |
| OTOPL | N | 8 | 2 | Сумма начисления отопления |
| EL\_C | N | 8 | 2 | Сумма начисления эл\энергии |
| G\_W | N | 8 | 2 | Сумма начисления гор.воды |
| X\_W | N | 8 | 2 | Сумма начисления хол.воды |
| KW\_PLL | N | 8 | 2 | Нач. квартплаты (по льготе) на 1-го чел. |
| TEL\_RL | N | 8 | 2 | Нач.телефона (по льготе) на 1-го чел. |
| RAD\_RL | N | 8 | 2 | Нач. радио (по льготе) на 1-го чел. |
| K\_YSLL | N | 8 | 2 | Нач. ком.услуг (по льготе) на 1-го чел. |
| OTOPLL | N | 8 | 2 | Нач. отопления (по льготе) на 1-го чел. |
| EL\_CL | N | 8 | 2 | Нач. эл\энергии (по льготе) на 1-го чел. |
| G\_WL | N | 8 | 2 | Нач. гор.воды (по льготе) на 1-го чел. |
| X\_WL | N | 8 | 2 |  Нач. хол.воды (по льготе) на 1-го чел. |
| \*SUM\_KW | N | 8 | 2 | Сумма нач. квартплаты по льготе |
| \*SUM\_GW | N | 8 | 2 | Сумма нач. гор. Воды по льготе |
| \*SUM\_XW | N | 8 | 2 | Сумма нач. хол. Воды по льготе |
| \*SUM\_KYSL | N | 8 | 2 | Сумма нач. ком. Услуг по льготе |
| \*SUM\_OT | N | 8 | 2 | Сумма нач. отопления по льготе |
| \*SUM\_TL | N | 8 | 2 | Сумма нач. телефона по льготе |
| \*SUM\_RD | N | 8 | 2 | Сумма нач. радио по льготе |
| \*SUM\_IT | N | 9 | 2 | Итог сумм начислений по льготе |
| ITOG\_N | N | 9 | 2 | Итог сумм начислений |
| ITOG\_L | N | 9 | 2 | Итог сумм начислений по льготе  |
| ITOG | N | 9 | 2 | Итог сумм начислений (к оплате) |

\* - Сумма начислений по льготам жильцов, проживающих в одной квартире

LGOT.DBF (Льготы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | Примечание |
| N\_LG | N | 1 | 0 | Код льготы |
| INFO | C | 20 | - | Описание льготы |
| KWP\_L | N | 4 | 2 | Ставка начислений по квартплате |
| RD\_L | N | 4 | 2 | Ставка начислений по радио |
| GW\_L | N | 4 | 2 | Ставка начислений по горячей воде |
| XW\_L | N | 4 | 2 | Ставка начислений по холодной воде |
| K\_L\_L | N | 4 | 2 | Ставка начислений по ком. Услугам |
| OT\_L | N | 4 | 2 | Ставка начислений по отоплению |
| TL\_L | N | 4 | 2 | Ставка начислений по телефону |

TABLE\_R .DBF (Ставки)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | Примечание |
| TAB | N | 4 | 0 | Табельный номер |
| KWP\_L | N | 5 | 2 | Ставка начислений по квартплате |
| RD\_L | N | 5 | 2 | Ставка начислений по радио |
| GW\_L | N | 5 | 2 | Ставка начислений по гор. Воде |
| XW\_L | N | 5 | 2 | Ставка начислений по хол. Воде |
| K\_L\_L | N | 5 | 2 | Ставка начислений по ком. Услугам |
| OT\_L | N | 5 | 2 | Ставка начислений по отоплению |
| EL\_L | N | 5 | 2 | Ставка начислений электроэнергии |
| TL\_L | N | 5 | 2 | Ставка начислений по телефону |

TARIF.DBF (Тарифы)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | Примечание |
| INFO | C | 25 | - | Описание тарифа |
| ST\_KA | N | 5 | 2 | Ставка оплаты |
| K\_INFO | C | 11 | - | Помощь начисления (на 1-го чел.\на 1 кв.м) |
| K\_CH | L | 1 | - | Выбор: .T. – на 1-го человека .F. – на 1 м2 |

HELP.DBF – БД, содержащая описание помощи.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Длина |  | Примечание |
| HLP | M | 10 | - | Описание помощи |

M\_ZAR .MEM - Файл, в котором хранятся переменные стоимости услуг ( \_\* ).

А также структурные файлы-индексы (CDX), для каждой базы, создаваемые программным путем.

Поддержка базы данных

поддержка или ведение базы данных подразумевает наличие средств доступа к данным и средств управления ими. При этом рабочий интерфейс должен быть максимально “комфортным”. Здесь нужно решить, что пользователь будет обычно видеть на экране компьютера.

Удобно в качестве “рабочего стола”, на котором он будет принимать решения, и совершать действия по обработке данных, взять не какое-то меню, а отображение самой базы данных. На экране должны присутствовать и указания на возможные действия пользователя. Поскольку на нем никогда не хватает места, указания на некоторые действия можно поместить в развертывающиеся световые меню. При необходимости также должна быть предусмотрена контекстно-зависимая помощь (HELP – F1).

*Функции поддержки базы данных.*

Для обслуживания базы необходимы следующие функции.

перемещение вперед\назад на одну запись (SKIP\SKIP –1). При этом должны блокироваться попытки выйти за первую\последнюю запись базы. Достижение верхней\нижней границы желательно отображать на экране (READ-окно).

Переход на начало\конец базы (GO TOP\BOTTOM).

Пометка записей, предназначенных для удаления, и снятие пометки (DELETE\RECALL).

Дополнение базы новой записью (APPEND BLANK).

Выход из окна редактирования.

Поиск по ключу (SEEK, LOCATE). В случае неудачного поиска указатель записей должен вернуться на исходную запись.

Вывод\печать данных.

Упаковка данных (PACK).

Переиндексация базы (REINDEX или INDEX).

Полная очистка базы от данных (вспомогательная база oplata.dbf) (ZAP).

*Средства предъявления данных.*

Команда BROWSE\CHANGE удобна тем, что дает панораму данных и не требует программирования перемещения в БД. В данной программе команда BROWSE в основном используется для просмотра данных и результатов вычислений.

Команда READ с командами @…SAY…@…GET:

@…SAY…GET…

@…SAY…

@…GET…

READ …

Эта форма предъявления данных является самой гибкой и дает непосредственный доступ к полям базы. Теперь кроме собственно редактирования, поля базы данных можно анализировать, трансформировать, запоминать в переменных или в других базах и т.д.. Чаще всего такой режим называют READ-окном, хотя использование команд @…GET, READ не обязывает помещать их в специальное окно – они могут проектироваться и непосредственно на экран.

Все операции над базой, предъявляемой командой READ, могут быть совершены либо после завершения команды READ, либо в вызываемых процедурах.

*Обе формы (BROWSE и READ) предъявления данных активно используются в программе.*

Описание работы приложения.

Управление программой осуществляется с помощью меню, так как меню является основной формой диалога в прикладных системах обработки данных. FoxPro обладает развитыми средствами поддержания меню как с объемным световым курсором (световым “зайчиком”), так и с назначаемыми клавишами. Кроме того, в FoxPro реализовано еще и “кнопочное” (Button) меню, которые означают не физическую кнопку на клавиатуре, а некоторую область на экране, которой приданы управляющие свойства. “Кнопочные” меню удобно использовать совместно с мышью.

В FoxPro имеются две альтернативные технологии построения меню – концепция, продолжающая еще из предыдущей версии СУДБ FoxBASE-2,1, и концепция, позаимствованная из СУДБ dBASEIV.

Важнейшее “идейное” различие между Fox-меню и dBASE-меню может быть определено терминами “меню-программа” и “меню-объект”. Fox-меню является только частью программы, где оно создается, используется и “умирает”. dBASE-меню после своего определения остается независимым и “живым” объектом, к которому можно обратиться из любого места прикладной системы.

Поэтому управление программой осуществляется именно с помощью dBASE-меню, а также в процедуре показано использование Fox-меню.

dBASE-меню может не только вырабатывать переменные для анализа, но и непосредственно вызывать процедуры, подпрограммы, команды по обработке выбора. Оно предлагает такую архитектуру построения систем обработки данных, когда программа может вообще не иметь единого ядра, а состоять из процедур, которые связаны друг с другом только через вызовы меню. Кроме того, имеется неоценимая возможность создавать меню из имен файлов и компонентов базы данных, осуществлять множественный выбор.

В общем случае для создания dBASE-меню и работы с ним необходимо предусмотреть следующие элементы:

Определение меню. Здесь описываются содержание, “горячие” клавиши, клавиши быстрого доступа, форма и реакции меню. Определение меню может сделано один раз в начале программы и далее только использоваться.

Активация меню. Команды\клавиши активации предъявляют меню на экране и делают его чувствительным к выбору пользователя.

Деактивация меню. Удаляет его с экрана, сохраняя в памяти для следующего возможного использования.

Удаление меню. Это действие очищает память от определения меню, и более оно не может быть использовано без нового определения.

Имеются два типа элементарных меню:

Вертикальное (“всплывающее”, или POPUP-меню),

Горизонтальное (BAR-меню)

На их основе можно строить иерархические меню практически любой сложности (вложенности). Именно по такой технологии построено меню программы и вид меню показан на рис. 2.1

*Рисунок 2.1* Структура меню (ZAR)

*BAR-пункт “Работа с картотекой”* – при выборе этого пункта открывается READ-окно, в котором представлена информация о квартиросъемщике. В этом окне, возможно, изменять начисления услуг, изменять тарифы для жильцов квартиры или дома, период льготы, количество проживающих, возможность формирования, просмотра и печати квитанции, ввода суммы оплаты, расчета квартплаты, выбора фильтра отображения жильцов (Плательщики, Льготники, Все жильцы) и перемещения по базе дынных.

*BAR-пункт “Квартиросъемщики”* – при его выборе отображаются те жильцы, которые оплачивают квартплату. Вывод осуществляется в табличной форме, где показываются поля с табельным номером и фамилией, а также два дополнительных (вычисляемых) поля, которые не сохраняются в базе данных. Это поле со значком “◼”, который появляется у тех жильцов, которые платят за квартиру и поле со значком “” появляется у жильцов, которые имеют льготу. На свободной области экрана выводится остальная информация о жильце, включая информацию о произведенных начислениях. Если курсор перемещать, находясь в поле со значком “◼”, то автоматически будет обновляться информация на экране. На всех других полях, при нажатии клавиши “Enter” появляется Fox-меню, где можно выбрать дальнейшие действия.

*BAR-пункт “Картотека льготников” -* при его выборе показываются те жильцы которые имеют льготу. Вывод осуществляется в табличной форме, где показываются поля с табельным номером, фамилией, адрес, телефон, категория льготы, номер (название) документа, подтверждающего льготу. В нижней части экрана отображается статус-строка, где показывается дополнительная информация, а именно: если льготник является плательщиком, то показывается количество проживающих и количество льготников, иначе фамилия и таб. № того, кто платит за квартиру.

*BAR-пункт “База жильцов” -* такой же показ как в *“Картотеке льготников”,*только отображаются все жильцы.

*BAR-пункт “Ввод счетчика”* – это табличный вывод всех плательщиков, с данными счетчика, при вводе новых значений счетчика, сумма автоматически вычисляется и корректируется сумма к оплате.

*BAR-пункт “Ввод оплаты” -* это табличный вывод всех плательщиков, с данными о суммах начислений, сумме к оплате и полем для ввода суммы платежа. При вводе суммы платежа автоматически рассчитывается остаток и заносится дата платежа, которые затем отображаются в отчете и квитанции.

PAD-пункт “Отчеты”

*BAR-пункт “Списки по начислению”* - формируется отчет о всех произведенных начислений, и открывается окно с выводом сумм по каждой услуге, отдельно сумм по каждой услуге по льготам, итоговая сумма к уплате, остаток или задолжность, дата платежа.

*BAR-пункт “Списки по льготам” -* формируется отчет и открывается окно с информацией о каждом льготнике, включая суммы по каждой услуге, а также показывается фамилия и таб. № того к кому привязан льготник (т.е. того, кто платит за квартиру).

*BAR-пункт “Списки с показанием счетчика” -* формируется отчет и открывается окно, в котором указаны фамилия, таб.№, адрес и данные старого и нового значений счетчика.

*BAR-пункт “Оплата электроэнергии” -* формируется отчет и открывается окно, где в табличной форме выводятся фамилия, таб.№, адрес, данные старого и нового значений счетчика и сумма к уплате.

*BAR-пункт “Сальдо по квартиросъемщикам” –* формируется отчет и открывается окно, где в табличной форме выводятся фамилия, таб.№, суммы начисления, к оплате и остаток.

 *BAR-пункт “Свод по квартплате за месяц” -* формируется отчет и открывается окно, где в табличной форме выводятся общие суммы по каждой услуге, отдельно по льготам, итоговые суммы начислений по квартплате, по льготам и суммы, которую организация должна получить за месяц.

*После закрытия любого окна с отчетом, открывается окно с предложением распечатки.*

PAD-пункт “Печать” - состоит из имен файлов, которые были созданы при выборе BAR-пунктов “Отчеты” это:

*“Списки по начислению”* - файл NACH-xx.TXT

*“Списки по льготам” -* файл LG\_KW-xx.TXT

*“Списки с показанием счетчика” -* файл ELC\_V-xx.TXT

 *“Оплата электроэнергии” -* файл EL\_PL-xx.TXT

*“Сальдо по квартиросъемщикам” -* файл SALDO-xx.TXT

*“Свод по квартплате за месяц” -* файл SVOD-xx.TXT

Где хх – это номер месяца, за который сформирован отчет. Если установить курсор на имя соответствующего файла и нажать клавишу “Enter” (ВВОД), то откроется окно для просмотра-редактирования отчета. Если установить курсор на имя соответствующего файла и нажать клавишу “Spacebar” (ПРОБЕЛ), то файл распечатается, при условии, если принтер готов, иначе появится сообщение о подготовке принтера.

PAD-пункт “Сервис”

*BAR-пункт “Дополнение”* – открывается окно, в котором можно добавить жильцов в базу данных, с указанием льгот и услуг.

*BAR-пункт “Дополнение льгот” -* открывается окно, в котором добавляются или изменяются код, описание льготы и ставки по ним.

*BAR-пункт “Тарифы”* – открывается окно с dBASE-меню PAD-пункты которого позволяют:

PAD-пункт “Просмотр” – формирует POPUP-меню из полей базы с описанием тарифов, при нажатии клавиши “Enter”, на выбранном BAR-пункте с описанием тарифа, появляются поля редактирования данного тарифа с возможностью удаления или изменения.

PAD-пункт “Добавить” – выводит поля для дополнения тарифа.

PAD-пункт “Выйти” - закрывает окно и меню тарифов.

*BAR-пункт “Нормы оплаты услуг” –* при его выборе открывается окно, в котором нужно указать стоимость каждой услуги. Эти ставки действуют для расчета начисления квартплаты по умолчанию для всех жильцов.

*BAR-пункт “Переиндексация” –* при его выборе происходит процедура переиндексации всех БД.

*BAR-пункт “Упаковка данных” –* запускает процедуру физического удаления данных, которые ранее были помечены на удаление.

PAD-пункт “Расчет”

*BAR-пункт “Расчет квартплаты” -*при его выборе происходит расчет квартплаты за текущий месяц и после выводится окно, где в табличной форме показывается таб.№, фамилия и начисления по каждой услуге, с возможностью их редактирования. В нижней части экрана показывается общие суммы по всем услугам и общая итоговая сумма, также эти суммы выделяются цветом в зависимости от нахождения курсора в таблице.

*BAR-пункт “Расчет по льготам” –* при его выборе происходит расчет для тех жильцов, которые имеют льготу и удовлетворяют требованию начала и конца действия льготы, если эти поля у жильца отсутствуют предполагается, что он не имеет срока ограничения. Вывод данных и информация в нижней части аналогична как в пункте “*Расчет квартплаты”.*

*BAR-пункт “Слияние квартплаты с льготами”* – при его выборе происходит расчет суммы к оплате т.е. “*начисления по квартплате”* минус “*начисления по льготе”.* После этого выводится окно с полями: таб.№, фамилия, итоговая сумма начислений, итоговая сумма начислений по льготе и сумма к оплате. В нижней части экрана показывается адрес плательщика, количество проживающих и количество льготников.

PAD-пункт “Помощь”

*BAR-пункт “Помощь” -*выводит окно с описанием назначений PAD- и BAR-пунктов.

*BAR-пункт “Справка” –* выводит окно с кратким описанием предназначения программы.

PAD-пункт “Выход” – позволяет выйти из программы.

Работа окнами.

В приложении используются два вида окон это:

Окна, предназначены для обработки данных.

Окна, предназначены для выбора действия или отказа, либо выводят дополнительную поясняющую информацию. Такие окна еще называют *диалоговые окна.*

При начальном запуске приложения, когда база данных пуста, любой выбор пункта меню, для работы которого необходимы данные о жильцах, вызовет диалоговое окно с предложением начать работу с дополнения. Это окно содержит поясняющий текст и две кнопки <Добавить> и <Отмена>.

Окно “Работа с картотекой” (INS)

При выборе пункта меню “КАДРЫ”-“*Работа с картотекой”* открывается окно размером во весь экран, в котором размещается вся основная информация о жильце. Окно визуально делится по вертикали на две части, где в первой части (около 2\3 окна) располагается сведения о жильце, а в оставшейся части, которая выделена цветом, располагаются кнопки для выбора определенных действий. Рисунок окна показан на *рисунке 2 приложения 2*. Рассмотрим эти части поподробней.

1 часть “Сведения о жильце”

В первых строках расположена личная информация: фамилия, таб.№, адрес, телефон, площадь квартиры, которые недоступны для редактирования, дата рождения, а также автоматически вычисляемая информация о количество лет.

Количество проживающих и количество льготников вычисляются в зависимости от записей в БД и в ней не сохраняются, напротив этих полей находится поле с количеством проживающих, которое сохраняется в БД и служит основанием для расчета квартплаты.

Далее идет надпись “СОСТАВ СЕМЬИ”, которая накрыта невидимой кнопкой, при выборе которой выпадает POPUP-меню с фамилиями проживающих в данной квартире с указанием кто за неё платит и кто является льготниками. При выборе элемента меню, окно обновляется информацией об этом жильце.

Далее идет строка с информацией о льготе: код, наименование льготы, а также с какого и по какой период она действует. Если период не указан, то предполагается, что льгота не имеет ограничения.

Кнопки выбора начислений услуг ( [ ]-не начисляется, [x]-начисляется)

Поясняющий текст к кнопки (для какой услуги, которую нужно выбирать)

Ставки за услуги, эти надписи накрыты невидимыми кнопками, выбор которых открывает окно и меню, где можно изменить ставки для жильцов квартиры (дома). Описания тарифов отображается в окне в зависимости от выбора тарифа: так если расчет услуги производится в зависимости от количества проживающих в квартире, то отображаются только тарифы на одного человека, если рассчитывается от занимаемой жилплощади, то на 1 квадратный метр. (*см. рис. 3 прилож. №2,* а также см. описание пунктов меню “Сервис”-“Тарифы”).

Показываются суммы начислений по услугам

Показываются суммы начислений по услугам с учетом льготы.

Заголовки столбцов.

В столбце “Льготы” предоставлена общая сумма начислений по льготам для данной квартиры и чтобы получить информацию о начислениях по льготе жильца, информация по которому представлена в окне, предназначена кнопка “[]”, которая доступна только в том случае если жилец имеет льготу и при его выборе появляется мини окно, где показаны его начисления по льготе.

На следующей строке представлены итоговые суммы по начислению, льготам и к оплате.

Далее дата оплаты, сумма уплаты и остаток или задолженность.

И на последней строке расположены две кнопки “По льготам” и “По оплате”, которые доступны в том случае если данный жилец имеет льготу либо имеет ордер, либо и то и другое.

“По льготе” – при ее выборе появляется окно, которое показано на *рис. 5 прилож. 2*, в котором указан код, ставки и описание льготы для данного жильца. Вся информация может корректироваться, важно только помнить, что вся измененная информация повлияет на всех жильцов, имеющих этот код.

“По оплате” – при ее выборе появляется аналогичное окно, описанное выше, которое показано на *рис. 6 прилож. 2*, в котором содержится информация о ставках стоимости услуг для данного жильца. Если расчет ведется по нормативным ставкам, то все поля имеют нулевое значение, а также если нужно данному жильцу установить нормативные ставки, то нужно обнулить соответствующее поле. После выхода из полей редактирования появится предупреждение:

“ Установить всем жильцам дома Y/N ”,

которое будет ждать нажатие клавиш. Если нажать клавишу “Y” то ставки, которые отображались в окне будут установлены всем жильцам, проживающих в данном доме, при нажатии любой другой клавиши ставки будут установлены только жильцам данной квартиры.

На этом информация в первой части окна заканчивается, и перейдем ко второй части.

2 часть “Кнопки - меню”

Во второй части расположены кнопки, при выборе которых выполняется определенные действия по обработке данных (*см. рис 2 приложения. 2 стр. 2).* Разберем их по порядку расположения.

Кнопка “Изменить” – при ее выборе открывается окно, которое показано *на рис 20* *приложения 2 стр. 20*, где можно изменить данные о жильце. Если изменяется адрес жильца, то автоматически пересчитывается количество проживающих по старому и новому адресах. Чтобы сохранить измененную информацию необходимо выйти из окна по кнопке “Подтверждаю”, если выйти по кнопке “Отмена”, внесенные изменения не сохраняются.

Кнопка “Добавить” – открывается аналогичное окно, описанное выше, которое показано *на рис 19* *приложения 2 стр. 19*,только все поля имеют пустые значения. Также некоторые поля, при редактировании проверяются на корректность вносимых данных. Подробнее это будет описано при рассмотрении этого окна. Выход также должен происходить по кнопке “Сохранить” иначе жилец и его данные не будут внесены в БД.

Кнопка “Удалить” - помечает на удаление жильца, находящегося в окне и информация в окне обновляется.

Кнопка “Печать…” - формирует квитанцию о начислениях и оплате, и открывает окно, которое показано *на рис 7 приложения 2 стр. 7*. После его закрытия появится окно с просьбой выбрать решение распечатки данного документа (*рис 8 приложения 2 стр. 8).*

Кнопка “Ввод оплаты” - открывает окно, которое показано *на рис 4* *приложения 2 стр. 4*, где возможно вести сумму оплаты для квартиросъемщика, который платит за квартиру.

Следующие кнопки позволяют фильтровать данные для просмотра:

Кнопка “Плательщики” – для просмотра будут доступны жильцы которые платят за квартиру;

Кнопка “Льготники” – для просмотра будут доступны жильцы, которые имеют льготу, независимо от периода ее действия;

Кнопка “Все жильцы” – для просмотра будут доступны все жильцы, кроме помеченных на удаление.

При этом под этими кнопками расположена подсказка, какой фильтр выбран.

Следующая группа кнопок предназначена для перемещения в БД:

Кнопка “Вверх” - позволяет перейти на одну запись вверх;

Кнопка “Вниз” - позволяет перейти на одну запись вниз;

Кнопка “Начало” - позволяет перейти в начало БД;

Кнопка “Вверх” - позволяет перейти в конец БД.

При этом если достигнуто начало (конец) БД, то соответствующие кнопки становятся недоступными, а в правом углу окна появляется предупреждение, что достигнут конец (начало) БД.

Кнопка “Выйти” – позволяет выйти и закрыть окно.

Кнопка “Расчет” – позволяет рассчитать квартплату для текущего жильца, с учетом сделанных изменений, а также расчет происходит, если в окне находится плательщик, иначе кнопка просто игнорируется.

Окно “Квартиросъемщики” (KD)

При выборе пункта меню “КАДРЫ”-“Квартиросъемщики” открывается BROWSE-окно с полями: табельный номер, фамилия и два вычисляемых поля *(см. рис 9 приложения 2 стр.9)*, которые показывают плательщика и льготников. При этом отображаются только те жильцы кто платит за квартиру. На свободной части экрана с правой стороны выводится информация по жильцу, на котором находится курсор, при этом она автоматически обновляется, если при перемещении курсор находится в поле, которое указывает, является ли жилец плательщиком. При нажатии в любом поле, кроме поля со значком (◼), появляется Fox-меню, где можно выбрать дальнейшие действия (см. рис 10 приложения 2 стр.10):

“Постоянная часть” – открывает окно “Работа с картотекой”, которое было описано выше

“Начисления” - обновляют информацию на экране в правой части относительно того квартиросъемщика, на котором находится курсор.

“Жильцы” – при выборе этого пункта в BROWSE-окне отображаются только жильцы квартиры того квартиросъемщика, на котором находился курсор.

“Плательщики” – возвращает окно с отображением только тех жильцов, которые платят за квартиру.

“Поиск” – при выборе этого пункта активируется POPUP-меню с просьбой выбора критерия поиска (*см. рис 13 приложения 2 стр. 13),* это может быть по фамилии, табельному номеру или по адресу. После выбора соответствующего пункта, кроме пункта “Отмена”, предъявляется окно для ввода ключа. При этом если это по фамилии, то возможен поиск и по неполному ключу, например по ключу “ПЕТР” будут удовлетворять поиску фамилии: “ПЕТРОВ”, “ПЕТРОВСКИЙ”, “ПЕТРИК” и т.д., если по адресу то предъявляются три поля для заполнения улицы, дома и квартиры. Здесь также возможен неполный ключ, то есть обязательным заполненным полем, должно быть, поле “улица” которое можно редактировать вручную или выбрать из выпадающего списка всех улиц, когда курсор попадает в это поле, а остальные поля могут быть пустыми. В этом случае указатель записи в БД установится на первого жильца в БД проживающего по этой улице.

Эта система поиска вызывается во всех окнах, где содержится информация о жильцах.

Остальные пункты были рассмотрены выше (окно “Работа с картотекой”).

Окно нормативов (M\_ZAR)

При выборе пункта меню “СЕРВИС”-“Нормы оплаты услуг” – открывается окно, где нужно вести ставки стоимости каждой услуги и фамилию того, кто занимается расчетом и эта фамилия будет подставляться в отчеты и квитанцию. Ставки, веденные в этом окне являются базовыми при расчете квартплаты для всех жильцов, кроме тех, для кого ставки определены отдельно *(см. рис 11 приложения 2 стр. 11).*

Окно дополнения льгот (INS)

При выборе пункта меню “СЕРВИС”-“Дополнение льгот” – открывается окно, в котором можно добавлять или изменять информацию по льготам. Первое поле в которое попадает курсор после открытия окна - это код льготы, если вести цифру кода, которая уже описана раннее, то информация в окне автоматически обновится, то есть поля будут заполнены в зависимости от веденного кода и возможно изменить информацию по этому коду. Для того чтобы сохранить веденные изменения или добавить льготу нужно нажать кнопку “Сохранить” *(см. рис 17 приложения 2 стр. 17)*

Окно тарифов

При выборе пункта меню “СЕРВИС”-“Тарифы” – открывается окно с меню для выбора действий.

*(см. рис 18 приложения 2 стр. 18).* Пункт меню “Добавить” предъявляет в окне поля для занесения данных по тарифу. Для изменения или удаления уже существующего тарифа нужно его выбрать в POPUP-меню, которое раскрывается при выборе пункта “Просмотр” *(см. рис 22 приложения 2 стр. 22).*

Окно дополнения жильцов

При выборе пункта меню “СЕРВИС”-“Дополнение” – открывается окно, в котором нужно занести информацию о жильце *(см. рис 19 приложения 2 стр. 19).* Работа с этим окном построена следующим образом:

Когда курсор попадает в поле ввода фамилии, то автоматически включается клавиша CAPSLOCK, для ввода фамилии прописными буквами.

После ввода табельного номера и выхода из этого поля, проверяется существование такого номера в БД и если такой уже есть, то появляется окно предупреждение с фамилией того жильца, который имеет этот табельный номер и выход из поля невозможен, пока не будет изменен табельный номер.

При выборе кнопки “[ ] Ордер” также проверяется БД и если за данную квартиру уже определен плательщик появляется окно с его фамилией и выбор этой кнопки отменяется.

при выборе кнопки “[ ] Льгота” появляется окно и меню выбора кода. Можно выбрать уже существующий код либо ввести новый, который добавится в базу льгот. Если код не будет выбран (код=0), то выбор льготы автоматически отменяется. После выбора льготы становятся доступны кнопки выбора услуг и верху над ставками льготы, появляется информация о номере льготы и что возможно его изменить нажав клавишу F7.

Период начисления льготы. В том случае если нужно определить период льготы то нужно заполнить поля с какой и по какую дату она действительна, здесь также проверяется корректность. Так если дата по которую должна начисляться льгота, будет меньше даты начала начисления, то выход из поля не возможен.

После ввода всех необходимых данных нужно нажать кнопку сохранить, информация будет занесена в БД и поля примут пустые значения для дальнейшего заполнения.

BROWSE-окна

Следующая группа окон имеет ряд общих характеристик это BROWSE-окна, открывающиеся в окне размером во весь экран, кроме двух последних строк, которые предназначены для вывода дополнительной информации - статус-строка. Все поля недоступны для редактирования, при комбинации клавиш: <Ctrl+T>-выполняется процедура удаления, <Ctrl+N>-процедура дополнения, <F5>-процедура поиска. Различие состоит только в отборе данных и выводимыми полями это:

Пункт меню “КАДРЫ”-“Картотека льготников” - предъявляет только тех жильцов, которые имеют льготу и выводит следующие поля: табельный номер, фамилия, период действия льготы, улица, дом, квартира, телефон, категория, номер удостоверения.

Пункт меню “КАДРЫ”-“База жильцов” – предъявляет всех жильцов и выводит следующие поля: табельный номер, фамилия, улица, дом, квартира, телефон, категория, номер удостоверения, количество проживающих.

Пункт меню “КАДРЫ”-“Ввод данных счетчика” - предъявляет только плательщиков и выводит следующие поля: табельный номер, фамилия, улица, дом, квартира, телефон, старое и новое значение счетчика и вычисляемое поле к оплате, которое автоматически рассчитывается при изменении данных счетчика и прибавляется к общей сумме платы за квартиру.

Пункт меню “КАДРЫ”-“Ввод оплаты” - предъявляет только плательщиков и следующие поля: табельный номер, фамилия, ввод оплаты, начисленную сумму за услуги, сумму по льготе, сумму к оплате, дату оплаты, остаток (задолженность) а также вычисляемое поле льготник *(см. рис 12 приложения 2 стр.12).*

Пункт меню “РАСЧЕТ”-“Слияние квартплаты с льготами” – предъявляет только плательщиков и выводит следующие поля: табельный номер, фамилия, сумма начислений, сумма по льготе и сумму к оплате *(см. рис 16 приложения 2 стр.16).*

Пункт меню “РАСЧЕТ”-“Расчет квартплаты” - предъявляет только плательщиков и выводит следующие поля: табельный номер, фамилия и суммы начислений по каждой услуге отдельно *(см. рис 14 приложения 2 стр.14).*

Пункт меню “РАСЧЕТ”-“Расчет по льготам” - предъявляет только льготников с учетом периода начислений и поля аналогичные описанных выше *(см. рис 15 приложения 2 стр.15).*

Окно помощи (HELP)

Это окно предназначено для вывода помощи для определенных окон при работе с данными по нажатию клавиши F1, а также при выборе пункта меню “ПОМОЩЬ”-“Помощь”, где дается помощь по пунктам меню и “Справка”, где дается информация для чего предназначена программа и ее краткая характеристика. *(см. рис 21 приложения 2 стр. 21).*

Краткий обзор Процедурного языка - FoxPro

Программа в FoxPro – это текстовый файл, содержащий набор команд, написанных в соответствии с синтаксическими правилами языка. Программа может иметь подпрограмму (процедуры), в которых помещаются часто повторяющиеся фрагменты кода, размещаемые после основного текста программы или в отдельном файле.

В FoxPro предусмотрена широкая возможность использования процедур, которые могу быть как внешними (в виде отдельных программных файлов), так и внутренними (внутри программы).

Эти средства удобно использовать для реализации некоторых одинаковых процессов обработки данных, а также вся программа может состоять из обращений к процедурам и функциям.

Кроме того, модульное программирование позволяет разработчику гораздо лучше ориентироваться в своей системе, а также сделать более эффективным процесс ее написания и отладки.

Разберем описание процедуры, функции и процедурного файла.

Подпрограмма начинается с ключевого слова

PROCEDURE <имя процедуры>

И выполняется пока не будет выполнено одно из следующих условий:

Еще раз встретят слово PROCEDURE.

Будет обноружено команда RETURN – возращение в предыдущую программу.

Будет выдана команда CANCEL –прерывание работы программы.

Будет выдана команда QUIT- выход из СУБД.

Встретится новая команда DO для запуска другой программы.

Будет достигнут конец файла.

FoxPro аналогично подпрограмме трактуется понятие пользовательской функции, которая начинается с ключевого слова

FUNCTION <имя функции>

и в отличии от процедуры может вернуть необходимые значения вызываемую программу.

Имеются 4 способа вызвать функцию:

Присвоить возвращаемое значение переменной. Например, следующая строка кода запоминает текущую системную дату в переменой dToday :

dToday=DATE()

Включить вызов функцию в команду. Например, следующая команда устанавливает по умолчанию каталог, имя которого возвращает функция GETDIR():

SET DEFAULT TO GETDIR()

Напечатать возвращаемое значение в активное окно:

? TIME()

Вызвать функцию без запоминания где либо возвращаемого значения:

= SYS(2002)

Для прерывания выполнения программы служит оператор

RETURN [Expression | TO MASTER | TO ProgramName]

Который возвращает управление вызывающую программу, и в ней выполняется следующая команда после вызывающей; если указана опция TO MASTER, то управление возвращается на самый верхний уровень вызывающей программы, а эта же программа с опцией TO ProgramName передает управление в указанную программу. При использовании функции команда автоматически возвращает .Т. (истина), если не указано другое выражение на месте Expression.

RETRY

Действует подобно команде RETURN, но при возвращении управления в вызывающую программу повторяется выполнение последней команды.

Встречая обращение в процедуре, Fox Pro ищет ее в следующей последовательности:

1.в текущей процедуре;

2.в процедурном файле, если он подключен;

3.снизу вверх в старших процедурах относительно текущей, если они есть;

4.на диске в виде отельной программы.

Обращение к процедуре выполняется команды

DO <имя командного файла> [WITH <список параметров>][IN <файл>]

В процедуру могут быть переданы и из нее получены некоторые величины, указанные в списке параметров после слова WITH. Этими величинами могут быть не только переменные и константы, но любые разрешенные выражения. В таком случае первой командой в вызываемой процедуре должна быть команда, воспринимающая их:

PARAMETERS <список параметров>

Процедурный файл

Множество внешних, обычно “родственных” процедур иногда удобно объединить в один процедурный файл, (также типа PRG), который вызывается (загружается в память) командой

SET PROCEDURE TO <имя процедурного файла>

Только после этого можно командой DO вызывать из него отдельные процедуры *(в данной программе имя процедурного файла FUNC.PRG)*

В каждый момент может быть открыт только один процедурный файл. Чтобы его закрыть, нужно использовать команду SET PROCEDURE TO без параметров.

Техническое описание работы программы

В приложении используются семь внешних процедур. т.е. 7 файлов типа .PRG из них один процедурный. Краткая характеристик каждого из них:

MENU.PRG-главный программный файл *(приложение 1.1)*

FUNC.PRG-процедурный файл *(приложение 1.2)*

OPEN.PRG-файл открытия БД *(приложение 1.3)*

BAZES.PRG-файл BROWSE-окон *(приложение 1.4)*

ADD\_DEL.PRG-файл дополнение и изменения данных *(приложение 1.5)*

RAS.PRG-файл расчетов квартплаты, льгот и их слияния *(приложение 1.6)*

OTCHET.PRG-файл формирование отчетов *(приложение 1.7)*

При старте программы запускается главный файл, т.е. файл MENU.PRG, который будет запускать работу всей информационной системы, он состоит из следующих блоков:

Блок установочных команд SET определяющих параметры конфигурации рабочей среды;

Открытие баз данных и необходимых индексных файлов;

Определение глобальных переменных, массивов и их инициализация;

Определение и описание окон;

Описание и активизация работы главного меню для выбора основных вариантов работы системы и передача управления соответствующим программным файлам или подпрограммам;

Закрытие баз данных и выход из СУБД.

Сейчас рассмотрим работу приложения в той последовательности, как она выполняется по шаговом выполнении каждого кода в окне трассировки. Для чего, при чтении данной главы целеобразно запустить среду FoxPro, открыть в окне трассировки файл MENU.PRG в режиме пошагового выполнения, либо обращаться к приложению 1, где прилагаются распечатки файлов программ

В первых строках файла MENU.PRG с помощью команд SET устанавливается операционная среда:

Отключение макросов

Установление даты

Запрет отображения на экране записей помеченных на удаление

Отключение статус-строки и т.д.

Далее определяются и загружаются в память цветовые схемы, которые используются для раскраски окон и некоторых областей внутри них.(SET COLOR OF SCHEME). Подключается процедурный файл (FUNC.PRG), командой ON ERROR и функцией ERROR() определяется анализ возможных ошибок в программе.

Далее вызывается процедура находящаяся в файле OPEN.PRG (*см. прилож. 1.3 стр. 1)* для открытия БД. Здесь проверяется наличие баз на диске и если их нет, то они создаются с помощью языка SQL, при этом в БД HELP, которая состоит из одного MEMO-поля, копируются текстовые файлы помощи. Если базы уже созданы ранее, то они открываются в заданных областях вместе со структурными индексами, которые содержат TAG’и для связи БД и систематизированного предъявления данных:

БД <RABOT.DBF> (База жильцов)

TAG-tab – индексирование по полю табельного номера, для связи с БД ставок (TABLE\_R.DBF).

TAG-fam – индексирование по полю фамилии, для поиска командой SEEK.

TAG-n\_lg – индексирование по полю номера льготы (код), для связи с БД льгот (LGOT.DBF).

TAG-date – индексирование по полям периода действия льготы (dat\_c,dat\_po), для расчета сумм по льготникам, рассчитываются только те льготники, у кого период входит в текущую дату и кто не имеет периода действия льготы.

TAG-lgt – индексирование по полям адреса, с условием, что предъявляться будут только жильцы, имеющие льготу.

TAG-ord – индексирование по полям адреса, с условием, что предъявляться будут только те жильцы, кто платит за квартиру.

TAG-adrr – индексирование по полям адреса и табельного номера, для связи с БД начислений (OPLATA.DBF), а также это главный TAG при просмотре данных.

БД <OPLATA.DBF (База начислений)

 TAG-tab –индексирование по полю табельного номера.

 TAG-adr – индексирование по адресу, для связи с БД жильцов (RABOT.DBF)

БД <LGOT> (БД льгот)

TAG-n\_lg – индексирование по полю номера льготы (код), для связи с БД жильцов (RABOT.DBF).

БД <TABLE\_R (БД ставок)

TAG-tab – индексирование по полю табельного номера, для связи с БД жильцов (RABOT.DBF).

 В том случае, если индексных файлов не обнаружено, то они создаются. Здесь также устанавливается связь между базами по ключевым полям. Связь между базами имеет структуру ОДИН КО МНОГИМ, то есть БД жильцов является родительской по отношению к другим.

Далее объявляются глобальные перемены (PUBLIC)

\_PAD\_OTCH - которое служит для анализа формирования отчета если ее значение .T. то отчет формируется, если ее значение .F., которая присваивается в процедуре дополнения, то при выборе формирования отчета программа просит провести слияние квартплаты со льготами, где данной переменной присваивается значение .Т.

\_REC- запоминает номер текущей записи в БД

\_FILTR- имеет числовое значение и в зависимости от значения устанавливает фильтр предъявления данных в окне “Работа с картотекой”.

Объявляется массив, содержащий имена месяцев и определяются переменные для хранения нормативных ставок, которые сохраняются в файле M\_ZAR.MEM и в последующем загружаются в память из этого файла.

После этого командой DEFINE WINDOW определяются и описываются окна. В основном в программе окна создаются в сомой процедуре, для того чтобы можно было использовать опцию CLOSE то есть возможность закрытия окна с помощью мыши. А здесь в основном определены окна для вывода подсказок-предупреждений и выбора дальнейших действий пользователя.

Команды DEFINE MENU, DEFINE POPUP и DEFINE BAR определяют и описывают расположение на экране и работу основного меню. Командой ON SELECTION PAD (BAR) определяется реакция пунктов меню при их выборе. Здесь нужно сказать, что команда ON SELECTION PAD может использоваться в виде ON PAD в этом случае, при попадании курсора на такой PAD-пункт сразу вызовет действия определенные при описании этих пунктов, а с добавлением слова SELECTION, чтобы выполнилось действие нужно непосредственно его выбрать, то есть на соответствующем пункте нажать клавишу ввода (Enter).

Последней командой в файле MENI.PRG является команда активации меню и вся дальнейшая работа приложения будет зависеть от выбора пунктов меню.

Любая система работы с базами данных предполагает наличия этих данных, поэтому работа с программой нужно начинать с дополнения БД информацией. В программе предусмотрено, что при выборе пункта меню, который предназначен для работы с данными, а база пуста появится окно с предупреждением дополнить. Рассмотрим подробней эту процедуру. (см прилож. 1.6 стр.1).

Здесь функцией RECCOUNT() проверяется наличие записей в БД и если значение равно нулю, то активируется окно VIB в котором при помощи команд ввода-вывода @…SAY…GET выводится поясняющий текст (SAY) и кнопки для выбора дальнейших действий (GET), которые активируются командой READ с опцией CYCLE, которая запрещает выход из окна по достижении курсора последнего GET-объекта . GET–объекты часто используется в программе, поэтому сейчас разберем их подробнее.

В данном случае используются кнопки меню, которые определяются в команде FUNCTION и обработка выбранной кнопки с помощью команды VALID. В процедуре INS2(), которая описана в приложении 1.2 стр. Эти объекты еще называют как средства управления в стиле WINDOWS все переменные, определенные командой GET имеют или числовые значения, тогда переменной присваивается номер объекта в порядке его расположения в команде, или символьные, тогда переменной присваивается символьное описание кнопки, а также для кнопок-переключателей и радио-кнопок возможно логическое значение переменной.

Затем в процедуре с именем определенной командой VALID или WHEN осуществляется в структуре DO CASE выбор дальнейших действий в зависимости от значения переменной. Например, в данном случае в окне определены две кнопки *Дополнить* и *Отмена* (FUNCTION ‘ \* …’), которые при их выборе присваивают числовое значение переменной ins1 и которая при выборе кнопки анализируется в структуре DO CASE в процедуре INS2. Eсли выбрана кнопка *Дополнить,* то переменная ins1 принимает значение 1 (потому что эта кнопка описана первой в команде FUNCTION ‘\* …’), выполняется условие CASE ins1=1 и следовательно выполняются команды которые следуют после этого условия (DO INS WITH 1). Если выбор не происходит вообще, то и процедура игнорируется.

Теперь еще раз пройдемся по пунктам меню, и разберем его работу с точки зрения языка программирования. Начнем с пункта “СЕРВИС”

“СЕРВИС”-“Дополнение”

Итак чтобы начать с дополнения БД нужно выбрать в меню “СЕРВИС”-“Дополнение” или в окне кнопку Дополнить. Выбор пункта меню или кнопки вызывает выполнение процедуры дополнения, которая определяется командой DO INS WITH 1. Здесь параметр 1 определяет что это процедура дополнения. В программе предусмотрена процедура Изменения с помощью выполнения этой же процедуры, но с параметром 2. Разберем эту процедуру. *(см. прилож. 1.5 и прилож. 1.2).*

В первых строках процедуры определяется имя для параметра, в переменной "ord\_a" запоминается название тега индекса, чтобы после выхода из процедуры восстановить параметры системы.

Далее закрепляются за клавишами "F1" и "F7" действия вызова помощи и изменения кода по льготе, глобальной переменной "\_PAD\_OTCH" присваивается значение ".F.", определяется POPUP-меню улиц и массив элементы которого в цикле DO WHILE преобразуются в BAR-пункты меню и с помощью переменной "m" массив переопределяется, чтобы в меню не было пустых строк. Определяется окно "hp" для вывода в нем меню кодов и описания льгот их выбора и занесения номера или описания документа льготы.

Далее в структуре DO CASE анализируется номер параметра. Если параметр равен 1, то это дополнение и с помощью команды SCATTER MEMVAR BLANK создается массив, содержащий переменные аналогичные полям БД с пустыми значениями, если параметр равен 2, то это изменение и массив будет содержать переменные с данными текущей записи БД (SCATTER MEMVAR) и здесь также определяются переменные, которым присваиваются значения полей, которые имеют логический формат и отображаются на экране с помощью GET-объектов это кнопки-переключатели для определения начислений услуг, кнопка-переключатель выбора льготы и кнопка-переключатель выбора плательщика. Также здесь в переменных запоминается текущий адрес плательщика (yl\_ins=yl, dom\_ins=dom, k\_ins=kw\_ra), для анализа изменения адреса.

После того как параметр определен активируется окно INS в котором располагаются переменные из созданного массива для занесения (изменения) данных (*см. рис.19 и рис 20 прилож. 2)*

Схема отображения данных реализовано следующим образом: в первой строке после активации окна вызывается функция POS\_CH2(), в которой содержатся SAY-объекты комментарии к GET-полям (*прилож. 1.2 стр.29).* Далее располагаются GET-объекты полей и выбора кнопок для большинства которых предусмотрена проверка при выходе (VALID) и входе (WHEN).

Поле-переменная *фамилия –* при входе выполняется функция =CAPSLOCK(.T.), для переключения ввода фамилии прописными буквами.

Поле-переменная *табельный номер -* здесь при выходе из поля выполняется процедура unic(), которая проверяет уникальность введенного номера (*прилож. 1.2 стр. 32).* То есть в структуре DO CASE определяется параметр если он равен 2 то возвращается значение .Т. и курсор входит в следующее поле, если он равен 1, то с помощью команды поиска проверяется наличие такого номера в БД и если такой номер существует, то функция возвращает значение .F., что означает, что выход из поля невозможен, а также появляется окно vib, в котором с помощью команды SAY выводится фамилия того жильца, который имеет этот табельный номер.

Поле-переменная *улица -* при входе выполняется функция yliz\_s() *(см. прилож. 1.2 стр.29),* где вначале выключается клавиша CapLock, а затем проверяется условие, что записей в БД больше нуля и параметр равен 1 и если это выполняется то активируется POPUP-меню улиц, для выбора нужной (*см. прилож. 1.2 стр. 29),* или закрыть меню и вести улицу вручную.

Переменная or1 - имеет числовое значение и соответствует полю OR\_R: 1 – если выбрана ([x], 0 – не выбрана ([ ]) и служит для выбора жильца плательщиком. Здесь также предусмотрена проверка при выходе. (*см. прилож. 1.2 стр. 31 FUNC O\_R())* в структуре DO CASE анализируется выбор кнопки, если она выбрана, то сначала объявляются переменные из полей-переменных с удалением из них пробелов, затем ведется поиск в базе жильцов с таким адресом и имеющих поле OR\_R со значением 1. Если поиск удачный и это не текущая запись (например при изменении), то появляется окно-предупреждение с фамилией жильца, который уже определен плательщиком за эту квартиру. Переменной or1 присваивается значение 0 и GET-объект обновляется. Если при выборе кнопки (or1=1) поиск неудачный, то полю-переменной M.OR\_R присваивается значение 1 и становятся доступными кнопки выбора услуг.

Переменная lg\_ta – имеет логическое значение и соответствует полю LGOT: .F.- если не выбрана ([ ]), .T.- если выбрана [x]. Здесь при выходе, если кнопка выбрана, что проверяется в структуре DO CASE *(см. прилож. 2 стр. 30 FUNC VIB\_LG),* то активируется окно с полями-переменными кода льготы (M.N\_LG) и описанием (номером) документа (M.N\_YD), при этом курсор попадает в поле с выбором кода, для которого предусмотрено проверка входа, то есть при входе в это поле выполняется функция LG1 *(см. прилож 1.2 стр. 29),* в которой определяется наличие данных в БД льгот и если они имеются определяется и активируется меню из полей БД льгот. Можно выбрать имеющиеся льготы или добавить новую. Если при выходе поле кода льготы будет иметь нулевое значение, то переменной lg\_ta и M.LGOT присваивается значение .F., иначе становятся доступные для редактирования поля с периодом, ставками и описанием льготы, а также становятся доступными кнопки выбора услуг.

Переменные kw, gw, ks, ot, elc, tl, rd – все эти поля имеют логический тип и соответствуют полям базы выбора для начисления по услугам. Если это процедура изменения то они имеют значения этих полей, как описано выше, если дополнение то все имеют значение .F..При этом они доступны для редактирования, только в том случае, если жилец является (выбран) плательщиком или (и) льготником. При выходе выполняется процедура <*имя переменной*>\_INS (*см. прилож. 1.2 стр. 30-31)*, где полю-переменной присваивается значение переменной.

Период льготы, ставки и ее описание – данная группа полей выделена цветом в зависимости от того доступны, (M.LGOT=.T.) или не доступны (M.LGOT=.F.) для редактирования. Здесь контроль данных проверяется только в поле-переменной, по какую дату должна начисляться льгота (M.DAT\_PO) то есть она должна быть больше даты начала даты начисления *(см. прилож. 1.5 стр. 2)*, остальные поля – это поля БД льгот.

Следующей командой определяющие кнопки, которые имеют числовое значение, в зависимости их расположения, при описании. 1-“Сохранить\Подтверждаю”, 2-“Выход”, 3-“Удалить”. Обработка выбора этих кнопок осуществляется в процедуре ad\_in() *(см. прилож. 1.2 стр. 32-35).* В первых строках выполняются действия по удалению возможных начальных пробелов (LTRIM()) в полях фамилии и адреса, далее определяется категория жильца и если это не квартиросъемщик, то удаляется не нужная информация (телефон и кв. метры), и наоборот. После этого определяется номер выбранной кнопки (pod) и номер параметра входа в это окно.

Кнопка “Сохранить”, параметр 1 (pod=1, red=1) – добавляется новая запись (APPEND BLANK) и все поля-переменные переносятся в эту запись (GATHER MEMVAR). Далее выполняются действия по заполнению поля с количеством жильцов. Ищется первая запись в БД с адресом нового жильца, в любом случае поиск будет удачным, но это может быть либо добавленная запись, либо другая. Если кроме новой записи больше не обнаружено, то полю количества жильцов (KOL\_VO), присваивается значение 1, иначе значение найденной записи, и в этом случае сканируется записи с этим адресом, чтобы увеличить поле-количество на 1. Затем открывается БД, в которой хранятся информация о ставках по услугам для каждого жильца (TABLE\_R) и в нее добавляется новая запись с копированием табельного номера новой записи из БД жильцов. Затем ищется в БД жильцов запись с одинаковым адресом и если такая найдена, то копируются в переменные значения полей из БД ставок (TABLE\_R), которые затем копируются в поля БД ставок новой записи, иначе они имеют нулевые значения. После этого командой SCATTER MEMVAR BLANK создаются новые переменные полей, переменным для выбора услуг присваиваются значения .F. GET-поля обновляются и можно редактировать следующую новую запись.

Кнопка “Подтверждаю”, параметр 2 (pod=1, red=2) – первой командой переменные переносятся в БД. После этого проверяется изменялся ли адрес, с помощью переменных определенных ранее (см. выше *yl\_ins ,dom\_ins, k\_ins).* Если адрес не изменился, то процедура заканчивается и возвращается в окно изменения, иначе устанавливается фильтр работы только с жильцами, имеющими текущий (измененный) адрес, и корректируется поле-количество. Затем аналогичная операция происходит с жильцами старого адреса.

Кнопка “Выход” – завершает работу процедуры, и выполняются команды следующие за командой READ, это закрытие окна, переиндексация данных и выход с процедуры.

Кнопка “Удалить” – доступна только в том случае, если это процедура изменения *(см. прилож. 1.2 стр. 35)* и будет рассмотрена ниже.

Теперь рассмотрим последнюю команду обновления GET-объектов – это команда-опция SHOW red(), входящая в команду READ. Она выполняется всегда если в какой-либо процедуре встретится выражение SHOW GETS, в данном случае в процедурах, вызываемых в окне Дополнения\Изменения данных, если встречается это выражение выполняется процедура red() *(см. прилож. 1.5 стр.3).* здесь в структуре DO CASE анализируются переменные и в зависимости от их значения выполняются следующие действия:

если параметр равен 1 (red=1), то кнопка описанная для сохранения новых данных будет иметь название “Сохранить”, а кнопка удалить будет не доступна

если параметр равен 2 (red=2), то кнопка “Сохранить” изменит свое название на “Потверждаю”.

Если переменная M.OR\_R=0, то кнопки-переключатели выбора услуг будут недоступны. И наоборот M.OR\_R=1, то доступны.

Если переменная M.LGOT=.F., то то кнопки-переключатели выбора услуг и поля редактирования по льготе будут недоступны. И наоборот.

“СЕРВИС”-“Дополнение льгот”

при выборе данного пункта запускается процедура INS\_LG, которая расположена в файле ADD\_DEL.PRG *(см. прилож. 1.5 стр. 5)* и показана на рис. 17 прилож. 2.

В первых строках выбирается БД, где содержится информация по льготам, затем определяется окно для работы процедуры и обращение к помощи. Открывается окно для редактирования льгот, в окне также используются не сами поля, а их переменные, созданные командой SCATTER MEMEVAR.

После ввода в поле “Код” номера, при его выходе выполняется процедура UNIC\_LG *(см. прилож. 1.2 стр. 36),* в которой в БД льгот ищется код с веденным кодом и если найден то переменные обновляются информацией по этому коду, и доступны для изменения или удаления, если нет то остаются пустыми и готовы к заполнению.

Так как работа происходит с переменными, то для сохранения сделанных изменений или дополнения, нужно подтвердить, нажав на кнопку “Сохранить”.

Работа кнопок осуществляется в процедуре LG\_INS, где в структуре DO CASE выполняются действия в зависимости от числового значения переменной LG\_INS:

LG\_INS=1. Из переменной M.INFO (Категория) удаляются возможные начальные пробелы, затем ищется код-переменная в БД льгот, если найден, то это изменение и поля переносятся в БД. Если не найден то дополнение и здесь сначала добавляется запись в БД, а затем в нее переносятся поля. Так как работа на этом не заканчивается, то создаются новые поля переменные и поля обновляются.

LG\_INS=2. Команда CLEAR READ завершает работу с этим окном.

LG\_INS=3. Запись помечается на удаление (DELETE) и физически удаляется (PACK), после этого создаются новые поля, поля обновляются и работа продолжается.

“СЕРВИС”-“Тарифы”

при выборе этого пункта выполняется процедура TARIFS\_ZAR, расположенная в файле FUNC.PRG *(см. прилож. 1.2 стрю 55)* и показано окно на рис. 18 прилож. 2.

В первых строках командой HIDE POPUP убирается с экрана меню “СЕРВИС”, затем закрепляется за клавишей Esc процедура выхода по клавише Esc, в переменной \_REC запоминается номер записи в БД жильцов. Выбирается активной БД тарифов и определяется окно для работы данной процедуры и горизонтальное меню, которое будет расположено в этом окне и с его помощью будут выполняться действия по обработке данных.

Данное меню имеет три PAD-пункта.:

PAD-пункт “Просмотр” – активируется POPUP-меню, которое состоит из полей БД тарифов, и выбор любого BAR-пункта выполняется в процедуре INS\_REC, в которую передаются два параметра это названия тарифа (PROMPT()-mprompt), и номер записи в БД тарифов (RECNO()-mrecno) *(см. прилож. 1.2 стр. 56),* в которой в зависимости от параметра выполняются следующие действия:

если описание пустое (empty(mprompt)), то эта запись помечается на удаление.

Иначе создаются поля-переменные для изменения данной записи (рис. 22 прилож. 2) так как редактируются переменные, то в окне присутствуют кнопки для сохранения сделанных изменений кнопка-“Сохранить” *(процедура del\_rec1, ras\_on1=1),* где сначала, в зависимости от значения переменной M.K\_CH (.T.,.F.) заполняется переменная M.INFO и затем переменные переносятся в БД; выхода без изменения кнопка “Отказ” *(процедура del\_rec1, ras\_on1=2)*, где командой CLEAR READ завершается работа с переменными-полями без перенесения их в БД;и удаления текущей записи *(процедура del\_rec),* текущая запись командой DELETE помечается на удаление.

PAD-пункт “Дополнить” и PAD-пункт “Выйти” - обрабатываются в процедуре INS\_ST, в которую передаются параметры, содержащие текст-приглашение данных пунктов.

Если выбран пункт “Дополнить”, то в процедуре INS\_ST создаются поля-переменные с пустыми значениями и с помощью кнопок дополняются записи.

Если выбран пункт “Выйти”, то в процедуре INS\_ST закрывается окно и деактивируется меню.

“СЕРВИС”-“Нормы оплаты услуг”

Данный пункт открывает окно M\_ZAR, где с помощью команд @…SAY…GET *(см. прилож. 1.4 стр. 4)* выводятся переменные ставок, определенные в начале программы в файле MENI.PRG и после их редактирования они сохраняются в файле M\_ZAR.MEM. чтобы их сохранить необходимо выйти из окна по кнопке “Сохранить”, которая запустит процедуру сохранения *(см. прилож. 1.2 стр. 20).*

“СЕРВИС”-“Упаковка данных”

Выбор этого пункта запускает процедуру физического удаления данных, помеченных ранее на удаление *(см. прилож. 1.2 стр. 22).* В первых строках сканируются записи, помеченные на удаление в БД жильцов и помечаются записи во вспомогательной БД ставок, которая связана с основной БД по полю табельного номера (TAB). Затем снова сканируется БД жильцов с целью проверки недопущения наличия жильцов без плательщика. Здесь перебираются помеченные записи, и в переменные заносятся значения адреса для поиска, и фамилии для возможного предупреждения.

После этого ищется жильцы, проживающие по этому адресу, если такие существуют, то ищется плательщик и если такого нет выводится окно, показанное на рисунке, где нужно выбрать одно из трех действий *(см. прилож. 1.2 стр. 23, процедура D\_IN).*

“Удалить всех” – будут помечены все записи по этому адресу.

“Изменить” – запустится процедура изменения, для найденного жильца.

“Восстановить” - восстановить плательщика, ранее помеченного на удаление, то есть снимет пометку командой RECALL.

“СЕРВИС”-“Переиндексация”

В этой команде я не пользовался командой REINDEX, а просто удалял все индексы и создавал новые. Это и проще и времени уходит меньше *(см. прилож. 1.2 стр. 23)*. В первых строках закрываются все БД и вместе с ними структурные индексы, далее с помощью команды DOS удаляются все файлы, имеющие расширение .CDX то есть структурные индексы, потому что другие индексы в программе не используются, и затем выполняется процедура открытия баз данных *(см. прилож. 1.3)*, где индексы создаются автоматически. Здесь нужно сказать, что при таком подходе FoxPro может выдать ошибку, и требование явно указать на БД, которую нужно открыть. Эта проблема решается с помощью функции ON ERROR DO EROR WITH ERROR(), которая передает код ошибки в процедуру EROR как параметр *(см. прилож. 1.2 стр. 53)*, и в ней в зависимости от области открываются нужные базы данных.

Пункт меню – “КАДРЫ”

После того как в базе жильцов имеются записи, можно работать с процедурами, осуществляющих действия по их обработке. Эти процедуры в основном находятся в первом пункте меню “КАДРЫ”.

“КАДРЫ”-“Работа с картотекой”

При выборе данного пункта открывается окно, представленное как личная карточка жильца, в которой отображается необходимая информация и средства по ее обработке *(см. прилож. 1.2 стр. 1-20).*

Как уже описывалось в главе 2 с работой в этом окне, вся информация, расположенная в этом окне делится на две части, поэтому каждую часть разберем по отдельности. Но прежде рассмотрим заполнение окна информацией и другими объектами *(см. прилож. 1.2 стр. 1-2 и прилож. 2 рис. 2 ).*

Итак в первых строках выполняются некоторые технические действия, а именно это проверка наличия данных, которая описывалась выше, определение и присвоение переменным “e” и ”b” логических значений для анализа конца и начала БД, и в зависимости от этого доступность или недоступность кнопок перемещения, клавише F1 назначается процедура открытия соответствующего окна с помощью, клавише F5 процедура поиска. После этого активируется окно INS и в него выводится следующая информация: это поля БД, имеющие символьные и числовые данные с помощью команды GET с опцией, запрещающей редактирование. Это поля фамилии (FAM), табельного номера (TAB), телефона (TEL), адреса (YL,DOM,KW\_RA), квадратных метров (KV\_M), суммы начисленной квартплаты из БД, в которой хранятся все начисления, и доступные для редактирования GET-поля количества (KOL\_VO), код и описание льготы из БД льгот, периода действия льготы (DAT\_C,DAT\_PO) и поля выбора услуг, имеющие логические значения, представленные с помощью переменных: kw (KW\_L)-квартплата, gw (G\_W\_L)-горячая вода, xw (X\_W\_L)-холодная вода, ks (K\_YS\_L)-коммунальные услуги, ot (OTOP\_L)-отопление, elc (EL\_C\_L)-електричество, tl (TEL\_L)-телефон, rd (RAD\_L)-радио, которые затем выводятся на экран с помощью GET-объектов (кнопки-переключатели). А также другие GET-объекты, которые будут рассмотренны позже.

Далее идут SAY-объекты, это подписи к GET-полям, а также результаты возвращаемые функциями, которые выводятся с помощью выполнения функции POS\_CH1 (=POS\_CH1( )) *(см.прилож. 1.2 стр.3).* разберем только выводимые значения, возвращаемые функциями.

Количество жильцов – выполняется функция KOL (прилож. 1.2 стр. 19), где путем сканирования формируется переменная k и возвращается ее значение (RETURN k), так как эта переменная числового типа, а команда SAY выводит только символьный тип данных, то необходимо преобразовать числовой тип в символьный и убрать первые пробелы, что позволяет следующие команды: LTRIM(STR(KOL(0))), где KOL(0) – полученное значение, STR – функция преобразования числовых данных в символьные, LTRIM – удаление ведущих пробелов.

Количество льготников – аналогичная функция, описанная выше только считает жильцов, если поле LGOT имеет значение истины (.Т.).

 Ставки по которым производится расчет для данного жильца. *(см. прилож. 1.2 стр 16-17).* Здесь сначала проверяется поле ставок БД ставок, так как связь одна ко многим то при перемещении курсора в основной базе он перемещается во всех вспомогательных базах и если поле в БД равно нулю, то параметру, который является результатом работы функции, присваивается значения нормативной ставки, иначе значения этого поля. Затем в команде SAY тип возвращаемого результата преобразуется в символьную форму и после вывода их на экран они накрываются невидимыми кнопками.

Числовые поля из БД начислений. Здесь эти данные просто преобразуются в нужный формат и выводятся на экран.

И в этой же процедуре раскрашиваются области внутри окна, для их более выразительности и выделения. (@ … FILL TO …).

Также необходимость заполнения окна с помощью выполнения функции вызвано тем, что информация по каждому жильцу является не всегда одинаковой, в окне предусмотрено перемещение по БД, а значит и обновление выводимой информации. Это достигается с помощью этой функции и опции обновления GET-объектов SHOW <имя процедуры>, которая входит в команду READ, которая будет рассмотрена ниже.

Вернемся к окну. Как уже описывалось выше (глава 2 стр. 26) вся основная информация о жильце содержится в первой части. Рассмотрим работу доступных GET-объектов в этой части.

Первый GET-объект это поле количества проживающих (@ 5,39 GET kol\_vo), при зтом как бы для проверки рядом выводится SAY-объект количества, рассчитанный программным путем (см. выше).

Далее SAY-объект “СОСТАВ СЕМЬИ” накрыт невидимой кнопкой (@ 6,27 GET family FUNCTION ‘\*I ‘ VALID FAMILY( ) …).Невидимой она называется потому, что не формируют никаких текстовых сообщений и только выделяются цветом, и при нажатии клавиш Ehter/Space могут вызывать функцию через опцию VALID. Здесь при нажатии этих клавиш происходит процедура FAMILY *(прилож. 1.2 стр. 8)* в которой в переменные заносятся данные о текущем TAG’e, и адресе, устанавливается фильтр доступа к данным, имеющим этот адрес и активируется меню, из полей БД жильцов, которое определено ранее в файле-процедуре MENI (прилож. 1.1) и пункты которого состоят из фамилии и двух вспомогательных полей, указывающих на категорию жильцов (плательщик, льготник). Выбор пункта этого меню отобразит в окне этого жильца и обновится информация относительно этого жильца, также в зависимости от категории жильца переменной \_FILTR присваивается значение 1-если жилец является плательщиком, 2- если льготник и 3- если ни тот ни другой (просто проживающий). Это нужно для того, чтобы при выполнении процедуры обновления (SHOW) в зависимости от значения этой переменной, будет установлен соответствующий TAG, влияющий на предъявление данных.

Следующая группа GET-объектов выводит поля выбора услуг с помощью кнопок-перек-лючателей (@ 10-17,2 GET kw(gw,xw,ks,ot,elc,tl,rd) FUNCTION ‘\*C <имя услуги> ‘VALID KW( )(GW(),XW()…)). Здесь переменные и поля имеют один и тот же тип данных, а именно логический и в процедурах, вызываемых при выходе по опции VALID просто полю присваивается значение переменной *(см. прилож. 1.2 стр. 17-18).*

Чутьиначе обстоит дела с выбором телефона и радио. Это сделано для того, что предполагается, что в квартире установлена одна телефонная и(или) одна радиоточка, поэтому если в квартире живут два или более льготников, которые пользуются льготными ставками на эти услуги, то при расчете им обоим (или более) будет убавляться плата по этим услугам, что в конечном итого телефон и(или) радио будет бесплатно или ЖКХ еще будет доплачивать за пользование этими услугами, что разумеется недопустимо, вот для этого осуществляется контроль выбора услуг. По своему содержанию процедуры одинаковы, поэтому разберем одну из них (см. прилож. 1.2 стр. 18).

Итак при выборе телефона (радио) вызывается процедура TL1(RD1) в которую передаются три параметра: 1-OR\_R (OR-в процедуре), 2-LGOT (LG-в процедуре), которые имеют значения полей плательщика (0,1) и льготы (.T.,.F.), относительно жильца которому выбирают услугу, 3-RECNO( ) (R-в процедуре), который имеет значение номера записи в БД этого жильца и 4-ORDER( ) (ORD-в процедуре), который имеет символьное значение активного индекса (TAG’a). Далее в процедуре в переменные заносятся значения адреса, для включения их в SCAN-условие, *а также, только для телефона, в условие поиска, где ищется плательщик и у него проверяется поле телефона и если оно пустое, то предполагается, что телефона нет и выбрать эту услугу невозможно.*

После этого в структуре DO CASE анализируются переданные параметры:

CASE OR=1.AND.LG=.T. то есть если это плательщик и он также льготник. В первой строке активируется TAG, при котором доступны все записи в БД, затем сканируется БД и у всех жильцов, имеющих этот адрес полю выбора телефона(радио) присваивается значение “Ложь” (.F.) и затем этому жильцу полю выбора телефона(радио) присваивается значение “Истина” (.Т.). то есть по этому условию плательщик имеющий льготу имеет преимущество перед другими жильцами, имеющими льготу, по выбору этих услуг.

CASE OR=1.AND.LG=.F. то есть если это плательщик, не имеющий льготу. Здесь просто полю выбора телефона(радио) присваивается значение “Истина”.

CASE OR=0.AND.LG=.T. то есть если это не плательщик, а льготник. В этом случае сканируются записи, имеющие этот адрес и льготу, и если у кого-либо поле выбора телефона(радио) имеет значение “истины”, то переменной TL(GET-объект) присваивается значение “Ложь” и после завершения сканирования значение переменной присваисвается полю выбора телефона(радио).

Следующие GET-объекты, это группа невидимых кнопок, которые накрывают области, где выводятся значения стоимости услуг (@ 10,28 GET tar\_s FUNCTION ‘\*I ;;;;;;;’ VALID TARIFS( )…), при выборе которых выполняется процедура TARIFS (*см. прилож. 1.2 стр. 11, прилож. 2 рис. 3),* где можно выбрать тарифную ставку для услуги на которой был сделан выбор, из имеющихся в БД тарифов или если нет, то добавить. При этом можно установить эту ставку всем жильцам дома, или можно установить норматив по этой услуге жильцу или всему дому. Работа, в окне и с меню во многом схожа с работой описанной ранее, при описании “СЕРВИС”-“Тарифы”. Рассмотрим отличия. В первых строках определяется окно и меню для работы процедуры.

Далее с помощью структуры DO CASE определяется на какой услуге был сделан выбор, и в зависимости от этого устанавливается ограничение на предъявление данных, например если это горячая вода, то появятся тарифы, имеющие значение .Т. поля K\_CH, (расчет на одного человека) и наоборот, переменной VIB\_STAVOK присваивается имя поля в БД ставок, для того, что в дальнейшем с помощью макроподстановки (&VIB\_STAVOK) этим полям будут присвоены соответствующие значения. И после этого активируется окно и меню для выбора ставки по этой услуге, при этом в правом углу появляется системное окно-подсказка, с помощью команды WAIT, с текстом выбранной услугой.

Пункт “Выбрать” предъявляет POPUP-меню с BAR-пунктами, состоящими из полей БД тарифов, и при выборе любого пункта появляется меню с выбором установить всем жильцам дома или квартиры. После этого в процедуре V\_ST1 куда передаются параметры номера BAR-пункта (BAR( )) и номера текущей записи в БД (RECNO( )) *(см. прилож. 1.2 стр.14-15).* Здесь в структуре DO CASE определяется номер выбранного BAR-пункта и для всех жильцов дома (параметр=1(“Установить всем жильцам дома”) или для данных жильцов квартиры (параметр=2 “Установить данному жильцу”), в скан- условии, полям БД ставок этих жильцов по выбранной услуге присваивается значения, которые имеются в поле ST\_KA в БД тарифов (REPLACE &VIB\_STAVOK WITH ST, где VIB\_STAVOK-это имя поля в БД ставок (см. выше) ST-ставка из поля ST\_KA в БД тарифов (ST=ST\_KA))

Здесь также добавлен повый пункт меню “Установить норматив”, при выборе которого появляется POPUP-меню с двумя BAR-пунктами “Установить всем жильцам дома” и “Установить данному жильцу” выбор этих пунктов осуществляется в процедуре V\_ST2 куда передаются параметры номера BAR-пункта (BAR( )) *(см. прилож. 1.2 стр.15).* Здесь в структуре DO CASE определяется номер выбранного BAR-пункта и для всех жильцов дома (параметр=1(“Установить всем жильцам дома”) или для данных жильцов квартиры (параметр=2 “Установить данному жильцу”), в скан- условии, полям БД ставок этих жильцов по выбранной услуге присваивается нулевые значения.

Последние GET-объекты в этой части экрана это кнопки “По льготе” и “По оплате”, которые доступны если жилец является плательщиком (“По оплате”) и(или) льготником (“По льготе”).

Кнопка “По льготе” – открывает окно (см. прилож. 2 рис 5), в котором с помощью команды @…SAY…GET, выводятся поля из базы льгот, по тому коду, который имеет данный жилец, доступные для редактирования. *(см. прилож. 1.2 стр. 9).*

Кнопка “По оплате” – открывает окно (см. прилож. 2 стр. 6), в котором с помощью команды @…SAY…GET, выводятся поля из базы ставок для данного жильца, доступные для редактирования *(см. прилож. 1.2 стр.10).* при этом после закрытия этого окна или после выхода из последнего (первого) GET-объекта все значения полей заносятся в переменные, появляется это же окно с информацией о количестве квартир в доме, по адресу которому живет плательщик, и с помощью команды WAIT будет предложено установить такие ставки всем жильцам дома (WAIT ‘Установить всем жильцам (Y/N)’ TO Y ) и если будет нажата клавиша Y (Н), то с помощью фильтра и сканирования эти ставки будут присвоены всем жильцам, проживающих по этой улице и дому, иначе только жильцам данной квартиры. Это достигается с помощью команды REPLACE <имя поля> WITH <имя переменной>, так как связь одна ко многим, то перемещения указателя в основной БД вызывает перемещение во вспомогательной.

В части второй этого окна расположены триггерные кнопки (Push Buttons), которые при их выборе выполняют определенные действия по обработке данных.

“Изменить”, “Добавить” - запускает процедуру изменения для жильца находящегося в окне, или процедуру дополнения нового жильца, которая описана выше (см. “СЕРВИС”-“Добавить” стр. 36).

“Удалить” – запускает процедуру удаления текущей записи *(см. прилож. 1.2 стр.35)* здесь вначале проверяется помечена ли запись на удаление и если помечена то пометка снимается и процедура заканчивается. Это нужно для того, что в окнах “База жильцов”, “Картотека льготников” также можно удалять записи комбинацией клавиш <Ctrl+T>, и этой же комбинацией клавиш снимать пометку на удаление. Далее идет команда переноса переменных в БД, если вызов этой процедуры пришел из функции изменения, затем в переменные заносится текущий адрес, устанавливается фильтр по этому адресу и в переменную kol заносится количество записей (COUNT TO kol), после этого переменная kol уменьшается на одну единицу, полученный результат присваивается всем записям по полю количества жильцов, затем эта запись помечается на удаление, и если вызов был с окна INS (“Работа с картотекой”), то информация и GET-объекты в этом окне обновляются.

“Печать …” – формирует квитанцию, с данными о всех произведенных начислениях, суммах к оплате, по льготам, дату когда оплачено и остаток (задолженность)*.*выбор этой кнопки запускает процедуру PRINT1 (*см. прилож. 1.2 стр. 26; прилож. 3 стр. 1; прилож. 2 рис. 7 )*, в которой выполняются следующие действия:

сначала переопределяется назначение клавиши F1, затем открывается альтернативный файл, в который будут заноситься формируемые данные и в последующем он будет сохранен на диске для его распечатки. В переменную “Т” заносится будущее имя файла (‘tab’+’.’+’txt’=tab.txt), затем объявляются два массива NACH(12,1)-для данных по начислениям, LG(9)-для данных по льготным начислениям. После этого в сканируется БД с условием только жильцов с одинаковым адресом и в переменные и элементы массивов заносятся необходимые данные, которые будут занесены в файл с помощью команды “?”, “??”. после окончания формирования квитанции этот файл будет открыт в окне VEDOM для его просмотра (редактирования) *(см. прилож. 2 рис. 7).*и после его закрытия будет предложено распечатать или отказаться *(см. прилож. 2 рис. 8)*

“Ввод оплаты” – открывает окно для ввода оплаты для одного квартиросъемщика. Выполняется процедура VVV расположенная в файле BAZES.PRG *(см. прилож. 1.4 стр. 5).*

Вначале переопределяется назначение на клавишу помощи, затем определяется окно для работы, затем анализируется жилец, на котором был вызов этой процедуры и если это не плательщик, то ищется плательщик квартиры и указатель записи переходит на эту запись, и работа будет производится с этим жильцом. Затем определяются переменные для работы это адрес (y,d,kv), текущая дата (dat), переменная для хранения веденой суммы (opl), погсле этого активируется окно (см. прилож. 2 рис.4) и в него выводятся поясняющие записи такие как фамилия, табельный номер, сумма начислений, сумма по льготам, сумма к оплате, остаток и GET-поле для ввода суммы оплаты, которая имеет контроль выхода *(см. прилож. 1.2 стр. 40)* здесь проверяется поле OPL\_TA , содержащая сумму оплаты жильцами и если она не пуста, то появится дополнительное окно с пунктами-кнопками “Дописать”, “Переписать”.

Выход с сохранением осуществляется выбором кнопки <Ok> в этом окне, которая описана в процедуре SV3 *(см. прилож. 1.2 стр. 40).*

данные кнопки предназначены для выбора фильтра предъявления данных в окне. Их выбор обрабатывается в процедуре FILTR *(см. прилож. 1.2 стр. 7),* где переменной \_FILTR присваиваются числовые значения, в зависимости от которых в процедуре обновления будет установлен соответствующий TAG.

“Вверх”, “Вниз”– данная группа кнопок предназначена для перемещения по БД жильцов. При этом по достижении конца(начала) БД соответствующие кнопки блокируются.. Обработка этих кнопок осуществляется в процедуре PER *(см. прилож. 1.2 стр. 6-7)* с помощью переменных “e” и “b”, которые определены ранее (см.выше), запоминается факт достижения конца/начала файла. Первоначально они имеют значение .F., если была предпринита попытка продвинуться за пределы БД то одной из них присваивается значение .Т.. В остальных случаях они равны .F. это нужно для восстановления в процедуре tb\_l() активности кнопок если указатель записей находиться не на последней/первой записи. Также здесь используется системная переменная \_CUROBJ, которая содержит номер GET-объекта. Ее значение изменяется при выборе кнопки и курсор переходит на другой объект. Она служит для того, что бы при перемещении с помощью клавиш Enter/Space курсор не покидал эти кнопки.

“Начало”, “Конец” - служат для быстрого перехода в начало/конец БД, при этом соответствующие кнопки блокируются.

“Расчет” - позволяет рассчитать квартплату на одного жильца, который находится в окне.выбор этой кнопки вызывает процедуру RAS\_ON\_ONE *(см .прилож. 1.2 стр. 53).* Здесь определяются переменные для работы это R-номер текущей записи в БД жильцов, Т-табельный номер, ORD\_R-активный TAG в БД жильцов, Y- улица, D-дом, K-квартира.

После этого активной выбирается БД начислений и в ней, по табельному номеру, ищется запись, для которой делается расчет (LOCATE FOR t=tab). Если поиск не удачный, то для БД жильцов устанавливается фильтр, при котором доступны только записи с текущим адресом и командой APPEND FROM <БД жильцов> FIELDS <список одинаковых полей в БД жильцов и БД начислений> добавляются записи в БД начислений. Если поик удачный, то разрывается связь между базами, для работы команды обновления, с помощью которой одинаковые поля в БД начислений принимают значения такие какие в БД жильцов: UPDATE ON <ключевое поле> FROM <область БД жильцов> REPLACE <поле БД начислений> WITH <поле БД жильцов>.

Далее текущей выбирается БД начислений и для нее устанавливается связь ОДНА-КО-МНОГИМ с БД ставок и БД льгот, а также фильтр для обработке данных имеющим только адрес, который ранее был сохранен в переменных.

Далее обнуляются все поля начислений по льготам, это сделано для того, что если в окне изменен период или льгота отменена, то при расчете льгот эта запись игнорируется, но старые данные расчета сохраняются и чтобы этого не допустить они обнуляются.

После этого в SCAN-цикле рассчитывается квартплата, с помощью команды REPLACE <поле> WITH <выражение>, то есть <полю> присваивается значение <выражения>. Здесь в выражении используется расчет (REPLACE kw\_pl WITH IIF(g.kwp\_l=0, kv\_pl, g.kwp\_l)\*IIF(kw\_l=.t.,kv\_m, 0)..).

Функция IIF(<условие>,<выражение1>,<выражение2>) – выдает значение <выражение1>, если <условие> истинно, и <выражение2> - если ложно. В нашем случае первый множитель – это ставка, и если поле ставки для этого жильца равно нулю, то применяется нормативная ставка, если же оно имеет значение, то применяется эта ставка. Второй множитель определяет выбор этой услуги, если поле фиксирующие выбор услуги равен .Т., то есть она выбрана для начислений, то умножение идет на квадратные метры(количество человек), иначе на 0.

После расчета квартплаты рассчитываются льготы. Здесь в SCAN-цикле предусмотрено условие при котором в расчет попадают только те жильцы, у которых либо не имеется периода действия льготы, либо текущая дата лежит в промежутке между началом и концом действия льготы. Расчет аналогичный описанному выше, различие только в том, что если по услуге расчет идет по квадратным метрам, то но высчитывается на одного человека (kv\_m/kol\_vo), и все значения умножаются на (-1), чтобы показать льготы с отрицательным знаком.

После окончания расчетов суммируются все начисления по льготам и заносятся в поля жильца, который платит за квартиру, для просмотра их в окнах, и расчета суммы к оплате. Затем разрываются все связи, отменяются фильтры, выполняется процедура открытия баз данных *(см. прилож. 1.3 стр. 1),* обновляются данные в окне и процедура завершается.

Выйти” – заканчивает работу с этим окном.

И теперь рассмотрим процедуру обновления окна tb\_l(), которая включена в команду READ, с помощью опции SHOW *(см. прилож. 1.2 стр.5-6).* Здесь в нескольких циклах DO CASE анализируются значения переменных, которые в процессе выполнения принимают различные значения, и в зависимости от этих значений выполняются определенные действия.

Анализ переменной \_FILTR – имеет числовое значение и служит для фильтра доступа к данным при их просмотре и редактирования. Значение 1 – устанавливается главным TAG ord, при котором доступны жильцы, которые платят за квартиру, 2 - – устанавливается главным TAG lgt, при котором доступны жильцы, которые имеют льготу, 3 - – устанавливается главным TAG adrr, при котором доступны все жильцы. А также на экран выводится строка-подсказка с выбранным фильтром

Далее анализируются категория жильца, и если это плательщик или льготник, то кнопки для выбора услуг – доступны для редактирования, если просто жилец то недоступны, при этом кнопки и область их расположения выделяются цветом.

Далее также анализируются категория жильца, если льготник – доступна кнопка “По льготе”, если плательщик – доступна кнопка “По оплате” и не доступны, если это не выполняется. Анализ выполняется в двух циклах по каждой кнопки, для того, что если плательщик имеет льготу, то ему доступны обе кнопки. Здесь также при анализе категории плательщика выводится разная информация по начислениям льгот *(см. прилож. 1.2 стр. 4).* Это – если плательщик выполняется процедура OB\_NACH, которая выводит на экран общие (суммированные) начисления по льготам, если это жилец то выполняется процедура LG\_NACH, которая выводит на экран начисления по льготе для данного жильца.

И последнее это анализ переменных “e” и “b”, которые служат для блокирования кнопок перемещений по БД в процедурах перемещений, при попытке выйти за пределы БД, а здесь они раз блокируются при удовлетворяющем значениии, которые присваиваются в процедурах перемещений.

“КАДРЫ”-“Квартиросъемщики”

При выборе этого пункта выводятся данные из БД жильцов в табличной форме с помощью команды BROWSE. Записи выводятся горизонтально в окне определенном ранее. Формат полей может настраиваться с помощью опции FIELDS <список полей>, а также иметь так называемые вычисляемые поля. Эти поля фактически не являются полями БД, но могут быть их функциями и отображаются на экране наравне с настоящими полями.

В данной процедуре KDR\_R *(см. прилож. 1.4 стр. 1)* в окне KD открывается BROWSE-окно с полями табельного номера (tab), фамилии (fam), вычисляемым полем со значком ‘⏹’, который появляется у жильца, который является плательщиком и вычисляемым полем со значком ‘✓’, который появляется у жильцов, имеющих льготу – эти поля являются функциями от фактических полей, выполняемые с помощью функции IIF(<условие>,<выр1>,<выр2>). Предъявление данных часть экрана, а на свободной части, выводятся на экран информация с помощью команд @…SAY…GET остальная интересующая информация *(см. прилож. 2 рис. 9),* которая формируется в процедуре NACH *(см. прилож. 1.2 стр. 37)* , при этом если при перемещении по БД курсор находится в поле табельного номера, то эта информация автоматически обновляется в зависимости от активной записи БД, во всех других полях эту информацию необходимо выбрать из меню, которое появляется после нажатия клавиши Enter в любом поле.

Эти действия достигаются путем включения в имена полей ключей, которые выполняют контроль входа (:W), контроль выхода (:V), а также проверку постоянно (:F) здесь:

 Для поля табельного номера с помощью ключа - :W=NACH() установлен входной контроль, при котором обязательно - :F выполняется процедура вывода информации по жильцу (выполняется процедура NACH ).

Для остальных полей контроль входа - :W=EN(), которая назначает на выбор клавиши Enter выполнение процедуры POP\_VIB и при выходе из поля - :V=NE() это назначение отменяется *(см. прилож. 1.2 стр. 38)*, при чем это выполняется всегда - :F.

Выполнение процедуры POP\_VIB предъявляет FOX-меню (см. отличия в описании меню), которое показано на рис. 10 прилож. 2 и описано в прилож. 1.2 стр. 39. Рассмотрим ее.

Сначала объявляется массив, элементы которого и будут пунктами меню, затем каждому элементу присваивается символьный тип с описанием названия, которое будет отображаться в меню. Далее с помощью команды READ оно активируется. После выбора из меню выполняется действие закрепленное за каждым пунктом в цикле DO CASE и оно “умирает”. Все пункты меню рассматривались выше.

“КАДРЫ”-“Картотека льготников”

Выбор зтого пункта также вызывает BROWSE-окно, открытое в окне KD1, которое открывается на весь экран, кроме двух последних строк, и предъявляет тех жильцов, которые имеют льготу (BROWSE FOR LGOT=.T.) Здесь выводятся поля фамилии, табельного номера, адреса, телефона, периода действия льготы, категории и номера удостоверения. При входе в поле выполняется процедура INFO *(см. прилож. 1.2 стр. 41)* с помощью ключей - :W=INFO() и - :F. Эта процедура выводит в нижних строках следующую информацию: если это плательщик, то суммы к оплате, остатка, количество льготников и количество жильцов, иначе фамилию и табельный номер того жильца, кто является плательщиком.

“КАДРЫ”-“База жильцов”

Работа процедуры, при выборе этого пункта аналогична описанной выше, только предъявляются все жильцы.

“КАДРЫ”-“Ввод данных счетчика”

Открывает в окне KD1 BROWSE-окно с плательщиками и полями табельного номера, фамилии, адреса, данными о старом и новом значениях счетчика и полем суммы начисленной за пользование электроэнергией из БД начислений *(см. прилож. 1.4 стр. 3).* Поля данных счетчика доступны к редактированию, остальные открыты только для просмотра, что достигается включением в имена полей ключа - :R. Также все поля имеют входной контроль - :W=INFO(), который описан выше, а поля данных счетчика и выходной контроль - :V=R() *(см. прилож. 1.2 стр. 42)*, где при выходе рассчитывается сумма к оплате за электроэнергию и корректируется итоговые суммы.

“КАДРЫ”-“Ввод оплаты”

здесь в начале определяются две переменные, которые нужны для работы, это dat, которая содержит текущую дату (dat=DATE()) и opl, которая имеет нулевое значение и служит для того, чтобы в программе не было ошибки при обновлении полей, потому что используется одна процедура обновления полей с процедурой для ввода оплаты для одного жильца. Затем открывается в окне KD1 BROWSE-окно, в котором предъявляются поля из разных баз, это табельный номер, фамилия, поле для хранения суммы оплаты (OPL\_TA), поле для хранения даты оплаты (D\_OPL), и поле для хранения остатка(задолженности) (OST\_K) из БД жильцов и поле итоговой суммы начислений (ITOG\_N), итога начислений по льготам (SUM\_IT) и суммы к оплате (ITOG) из БД начислений. *(см. прилож. 1.4 стр. 3-4, прилож. 2 рис. 12).* Доступные к редактированию является поле суммы оплаты которое имеет выходной контроль - :V=SV3() *(см. прилож. 1.2 стр.40)* , в которой полю остатка присваивается рассчитанное значение (введенная сумма – сумма к оплате), полю даты значение переменной dat, то есть текущая дата и здесь не используется ключ - :F, это значит, что если в поле, для которого предусмотрен контроль выхода (opl - :V=SV3), данные не изменялись то процедура игнорируется, а выполняется она только в том случае если данные изменялись. Также здесь используется другая информация в статус-строке :W=INFO3() *(см. прилож. 1.2 стр. 42),* где показывается адрес жильца, количество проживающих и количество льготников.

Пункт меню “РАСЧЕТ”

“РАСЧЕТ”-“Расчет квартплаты”

При выборе этого пункта появляется окно, где нужно подтвердить начало расчета выбором кнопки “Ок” *(см. прилож. 1.6 стр.1).* В окне выводится текущая дата и строка подсказка за какой месяц будет рассчитываться квартплата. Название месяца достигается с помощью массива и переменной mess, которые объявлены в файле MENI.PRG это массив mes(4,3), элементы которого содержат названия месяца и переменная mess содержит номер текущего месяца (=MONTH(DATE())) и выражение mes(mess) – выведет элемент массива по номеру, который содержится в переменной mess.

При подтверждении расчета выполняется процедура RAS\_1 *(см. прилож. 1.2 стр. 43-44).* В первых строках деактивируется окно, выбирается активной БД начислений и из нее удаляются все записи командой ZAP, затем командой APPEND FROM <БД жильцов> FIELDS <список полей> добавляются все записи из БД жильцов. После этого все БД закрываются (CLOSE DATABASE) и в области “а” открывается БД жильцов, для которой устанавливается фильтр по полю OR\_R то есть доступны только плательщики, в области “b” открывается БД начислений. После этого командой JOIN WITH <область> TO <имя новой БД> FOR <условие> FIELDS <список полей>, создается новая БД-RACH.DBF содержащая поля, которые перечислены после опции FIELDS и количеством равным в БД жильцов, что выполняется с помощью условия. Другими словами новая БД расчета содержит только плательщиков, с полями необходимыми для проведения расчета и хранения рассчитанных данных.

Далее снова закрываются все базы данных и в области “а” уже открывается новая БД расчетов, для которой создается структурный индекс для связи с другими БД. Затем устанавливается связь с БД жильцов и БД ставок и производится расчет, который аналогичен описанному, при рассмотрении кнопки “Расчет” в окне “Работа с картотекой” (“КАДРЫ”-“Работа с картотекой”). После завершения расчета, с помощью команды CALCULATE SUM(поле в котором храняться данные расчета) ТО <переменные в которые заносится результат работы команды> создаются переменные со значениями общих сумм начислений для каждой услуге, для формирования статус-строки при просмотре выполненного расчета. Далее с помощью команды @ …SAY они (переменные) выводятся в нижней части экрана и в окне KDR открывается BROWSE-окно с данными расчета и фамилией жильцов.

С помощью ключа входа в поле - :W=INFO1() внизу изменяет цвет сумма, которая относится к данной услуге (к примеру если курсор в поле горячая вода, то внизу окрашена область с общей сумме по горячей воде, как показано на рис. 14 прилож. 2), а с помощью контроля выхода - :V=INFO2() цвет принимает прежний цвет и с помощью ключа - :F выполняется всегда, независимо от изменения, это в основном нужно для ключа :V.

Работа этой процедуры показана на стр. 51 прилож. 1.2. здесь в структуре DO CASE анализируется в какое поле входит курсор (INFO1), и из какого выходит (INFO2) это достигается с помощью применения функции VARREAD()-которая возвращает прописными буквами имя поля(переменной), а с использованием с командой BROWSE/CHANGE возвращает имя поля строчными (кроме первой) буквами (см. также SYS(18)), и в зависимости от названия поля при входе, окрашивается определенная область внизу экрана, с помощью команды @…FILL TO … и цветовой схемы 12, а при выходе с помощью этих же команд этой области возвращается прежний цвет из цветовой схемы 1.

Перед открытием окна просмотра по начислениям, назначаются на клавиши выхода из BROWSE-окна, процедуры выхода (ON KEY LABEL ESC(CTRL+W,CTRL+Q) DO vib8), где активируется окно с кнопками “Сохранить” и “Отмена” *(см. прилож. 1.2 стр. 45)* и при выборе кнопки “Сохранить” открывается БД начислений, с помощью команды - UPDATE <ключевое поле> FROM <область> REPLACE <поле активной БД начислений> WITH <поле из указанной области (БД расчетов)>, переносятся рассчитанные данные в БД начислений. После этого выбирается активной БД расчетов, закрывается и удаляется с диска вместе со структурным индексом.

“РАСЧЕТ”-“Расчет по льготам”

Все действия, выполняемые при выборе этого пункта аналогичны как описано выше в расчете квартплаты за исключением некоторых отличий, которые не влияют на суть расчета описанного выше *(см. прилож. 1.2 стр. 46).*

Здесь имеющиеся записи не удаляются из БД начислений, так как предпологается, что расчет льгот производится после расчета квартплаты и все записи находящиеся в БД жильцов имеются в БД начислений (это достигается командой APPEND FROM описанной выше).

Также используется другой TAG при создании новой БД расчетов, а именно TAG date (см. прилож. 1.3 стр. 2) который делает доступными жильцов у которых текущая дата попадает в промежуток периода действия льготы и тех у кого период не определен.

При определении связи новая БД расчетов также имеет связь с БД льгот, для применения в расчете установленные проценты льготы по кодам.

Расчет льгот аналогичен описанному, при рассмотрению расчета на одного жильца в окне “Работа с картотекой” кнопка “Расчет”. *(см. прилож. 1.2 стр.53)*

И так как имена полей имеют разные имена, то для окрашивания общих сумм внизу экрана, при просмотре расчетов, применяется процедура для входа – INFO4, для выхода – INFO5 *(см. прилож. 1.2 стр. 52 и рис. 15 прилож. 2)*

“РАСЧЕТ”-“Слияние квартплаты с льготами”

Так как в программе отдельно насчитываются начисления по квартплате и льготам, то необходимо определить сумму к оплате, для чего и служит этот пункт. После выбора кнопки “Ок”, в окне описанном выше, выполняется процедура RAS\_3 *(см. прилож. 1.2 стр. 49).*

Здесь после закрытия всех баз данных в области “а” открывается БД начислений, в области “b” БД жильцов, затем объявляется переменная “m”, для хранения количества записей в БД (m=RECCOUNT()), что необходимо для выхода из цикла. Далее организовываются четыре цикла, из которых только самый внутренний SCAN-цикл осуществляет перемещение указателя записей. Остальные WHILE-циклы контролируют заданные условия, инициализируют, формируют и заносят данные в поля БД.

Первый WHILE-цикл имеет условие !EOF(), то есть цикл выполняется до тех пор пока не будет достигнут конец БД. Для этого служит функция EOF(), котороя принимает значение .F. при попытке указателя записи выйти за предел последней записи БД. Следующей командой служит определения переменной содержащей имя улицы, для работы следующего WHILE-цикла.

Также при расчете для баз данных применяются TAG’и, которые позволяют предъявлять записи в систематизированном виде по адресу, что позволяет использовать WHILE-циклы, а также SCAN-цикл с включением в него WHILE-условия.

Второй WHILE-цикл предъявляет те записи у которых улица совпадает с переменной определенной в первом WHILE-цикле, то есть жильцов с одинаковой улицей, а также здесь определяется переменная, содержащая номер дома, для работы следующего WHILE-цикла.

Третий WHILE-цикл предъявляет те записи, у которых улица и дом совпадают, то есть жильцов проживающих в одном доме. И здесь определяется переменная, содержащая номер квартиры, для работы следующего SCAN-цикла и переменные помогающие формировать данные, числового типа, которым присваивается значение 0.

Четвертый SCAN-цикл обрабатывает записи для одной квартиры. Внутри этого цикла используются команды управления (IF…[ELSE]…ENDIF), которые позволяют от удовлетворяющих условий присваивать переменным нужные значения и затем переносить их в поля БД. Так в первой команде выявляется, является ли жилец плательщиком и при истинном значении, переменной it присваивается значение итоговой суммы по начислению квартплаты, переменной r-номер записи, для занесения в нее, после окончания цикла, в поля плательщика нужных результатов. Во второй команде анализируются льготники и с помощью переменных (определенных в третьем цикле) присваиваются результаты сумм начислений по льготам.

После завершения цикла по квартире указатель записей устанавливается на следующую запись, которая запоминается в переменной n, затем в переменную os заносится сумма к оплате, командой GO r указатель переходит на запись плательщика и в поля для этой записи переносятся полученные результаты. Затем активной делается БД жильцов и в ней ищется эта запись и в зависимости от условий корректируются поля по остатку и оплате. Так если оплата не производилась (поле OPL пусто), то остаток будет рассчитанная сумма к оплате с отрицательным значением, иначе результат остатка, полученный при вычитании из внесенной суммы начисления, рассчитанной суммы по оплате. После этого выбирается снова БД начислений, проверяется условие, при котором если переменная n (номер следующей записи) больше переменной m (количество записей) то выполняется процедура просмотра и слияние расчетов заканчивается, иначе расчет продолжается для следующих записей. Для чего нужно это условие?

Дело в том, что для просмотра и формирования отчетов целеобразно хранить все данные, относящиеся к начисленным суммам по начислениям квартплаты и льготам в записи одного жильца, являющегося плательщиком, для чего в цикле осуществляется переход на эту запись. А так как при начале цикла в первом WHILE-цикле определено условие окончания, если переменная !EOF() примет значение .F., то при таком подходе (возврате на нужную запись в процессе цикла) эта переменная никогда не примет этого значения, так как она изменяет свое значение не тогда когда указатель записи находится на последней записи, а когда указатель записи попытается выйти за пределы последней записи, вот для прекращения цикла и используется это условие. А именно: переменная m хранит число всех записей а n хранит номер последней записи. И если n будет больше m, то это значит, что все записи рассчитаны. Единственным недостатком (по моему мнению) такого подхода является то, что переменная m ранит все записи, включая записи помеченные на удаление, и в зтом случае расчет для последней записи производится столько раз, сколько в БД записей на удаление, что не влияет на результат, а только на скорость обработки, но это ограничение не касается моего расчета, по той причине, что записи помечаются только в БД жильцов, а в БД начислений их не существует, так как БД начислений формируется из БД жильцов, для которой определено игнорирование записей на удаление (SET DELETE ON).

И последняя команда, которая выполняется при истинном условии описанном выше, это просмотр расчетов. Он осуществляется с помощью команды BROWSE *(см. прилож. 2 рис. 16)* со следующими полями: фамилия, табельный номер из БД жильцов и результаты начислений по квартплате, льготам и сумме к оплате из БД начислений, также используются три вычисляемых поля. Это поле-функция от полей, со значком “✓”,показывающая жильцов, имеющих льготу и поля со значками: “+”, “=”,показывающих суть расчета. В нижней части в статус-строке показывается адрес жильца, количество проживающих и льготников *(см. прилож. 1.2 стр. 42)*. А также здесь переменной \_PAD\_OTCH, присваивается значение .Т., которая позволяет формировать отчеты по выполненным данным.

Функция ПОИСКА

 На этом пункты меню по работе с данными заканчиваются и сейчас рассмотрим функцию поиска, которая выполняется во всех процедурах, предназначенных для обработки и просмотра данных. Эта процедура закреплена за клавишей F5, а в окне, которое вызывается при выборе “КАДРЫ”-“Квартиросъемщики”, закреплена за пунктом в Fox-меню “Поиск”.

Для выполнения этой функции используется:

меню, с пунктами определяющим по каким значениям будет производиться поиск (по фамилии, по табельному номеру, по адресу и отмена поиска), которое определено в файле MENI.PRG *(см. прилож. 1.1 стр. 4).*

Окно для задания критерия поиска, с помощью команды @…GET и переменной “а”, имеющей тип, в зависимости от выбора пункта меню. Окно также определено в начальном файле *(см. прилож. 1.1 стр. 2).*

А также используются разные процедуры. Для поиска по фамилии и табельному номеру – POISK, по адресу – PO\_ADR. Это сделано для того, чтобы при поиске по адресу возможно оставлять пустыми значения номера дома и(или) квартиры.

Выполнение функции поиска построена таким образом, что при нажатии клавиши (пункта меню) активируется меню поиска (ON KEY LABEL F5 ACTIVATE POPUP POISK (ACTIVATE POPUP POISK)) , где нужно выбрать необходимый пункт, как показано на рис. 13 прилож. 2, где при работе с картотекой льготников нажата клавиша F5 и выбран пункт “По адресу”. Для пунктов меню POISK’a командой ON SELECTION BAR определено выполнение процедуры POISK *(см. прилож. 1.2 стр. 24)*

Здесь в переменной \_REC запоминается номер текущей записи, для возврата при неудачном поиске. Далее в структуре DO CASE анализируется выбранный пункт.

Если PROMPT()=’По фамилии’(‘По табелю’), то главным выбирается TAG fam (tab), для поиска с помощью команды SEEK(), в команде @…GET определяется переменная “а” имеющая символьный тип (DEFAULT SPASE(25)) а для табеля числовой тип из четырех знаков (PICTURE ‘9999’), предъявляется окно POISK в котором расположено поле GET для занесения ключа поиска. После ввода искомого значения из него удаляются возможные пробелы (ALLTRIM(a)) для поиска по фамилии, что позволяет вести поиск и по не полному ключу (фамилии). Также значение переменной “а” запоминается в переменной “d”, а для табельного номера переменная “а” преобразуется в символьный тип (d=STR(a,4)), которая нужна для включения ключа поиска в сообщение WAIT о неудачном поиске. Сам поиск осуществляется после окончания команды DO CASE в команде IF…ENDIF - IF BAR()#4.AND.!EMPTY(a).AND.!SEEK(a) – если пункт не отмена и если переменная “а” не пуста и поиск неудачный, то выводится системное окно с помощью команды WAIT–WAIT ‘Поиск’+PROMPT+’:’+d+’ НЕУДАЧНЫЙ’ WINDOW – которое показано на рисунке:

Которое затем удаляется нажатием любой клавиши, и указатель записей возвращается на (GO \_REC) старое место. Потом происходит возврат в команду из которой был вызов. Если в результате поиска была найдена нужная запись, то в команде BROWSE, курсор будет стоять на ней, если в команде READ, то будет переход на эту запись и обновление информации и GET-объектов для этой записи.

Если PROMPT()=’По адресу’, то выполняется процедура PO\_ADR *(см. прилож. 1.2 стр. 25).* Здесь вначале определяется меню, массив и переменные с помощью которых в WHILE-цикле будут формироваться BAR-пункты из названий улиц. Затем массив переопределяется и его элементы – это пункты меню, которое активируется при входе в поле для ввода улицы и его выбор осуществляется в процедуре YLIZ (*см. прилож. 1.2 стр. 29),* где переменной m.yl присваивается значение параметра-mprompt. (ON SELECTION POPUP YL DO YLIZ WITH PROMPT()), где передаваемый параметр (PROMPT()) это название BAR-пункта, то есть название улицы. Работа с окном POISK осуществляется с помощью переменных полей адреса, которые создаются с помощью команды – SCATTER FIELDS yl, dom, kw\_ra MEMVAR BLANK. После заполнения полей и нажатии клавиши Enter (начать поиск) в структуре DO CASE анализируется сколько полей заполнено и с помощью команды LOCATE c FOR условием ведется поиск. Если он удачный то осуществляется переход на найденную запись, и если это READ-окно то информация о жильце обновляется, иначе указатель записи возвращается на прежнюю запись.

Пункт меню “ОТЧЕТЫ”

Подготовка и оформление отчетных данных без сомнения являются кульминацией любого делового приложения. Так что стоит приложить усилия для того, чтобы все многообразие данных, записанных в БД приложения, нашло достойное отображение в отчетах, напечатанных на принтере.

FoxPro имеет мощные средства построения отчетов для вывода данных в желательном для пользователя виде на принтер, экран или в текстовый файл, но в своей дипломной работе я не пользовался этими средствами, а попытался программированием вручную создать нужные выводимые данные. Это связано с тем, что я не преследовал коммерческой цели, создавая приложение, а также для того, что бы лучше освоить язык программирования и задействовать как можно больше команд и функций FoxPro.

Все отчеты формируемые с помощью данных баз построены по одной схеме, поэтому я разберу одну из них более подробно, а в остальных только кратко дам описание.

“ОТЧЕТЫ”-“Списки по начислению”

При выборе этого пункта формируется отчет для каждого жильца с суммами начислений по каждой услуге и отдельно начисления по льготам, показывается сколько оплачено и остаток если остаток положительный, то это сумму необходимо заплатить плательщику за пользование услугами, если отрицательный то эта сумма должна перейти на следующий месяц в счет будующих платежей за услуги. Также в шапке по каждому жильцу выводится дата оплаты, количество проживающих, из них количество льготников, занимаемая площадь. (см. прилож. 3 стр.1). формирование отчета выполняется в процедуре SP\_NACH *(см. прилож. 1.7 стр. 1).* В первых строках убирается меню отчетов с экрана (HIDE POPUP VIEV), затем проверяется значение переменной \_PAD\_OTCH, она должна иметь значение “Истины” (.T.), если ее значение “Ложь” (.F.), то значит выполнялась процедура “Дополнения\Изменения” и появится системное окно, с просьбой сделать слияния расчетов. (см. выше).

Затем переопределяется клавиша F1 для открытия в окне помощь подсказку по этой процедуре, в переменную “i” заносится номер текущего месяца (MONTH(DATE())), для включения его в имя файла, чтобы в последующем было легко опознать данные файла по его имени. Например в переменной формируется имя за март месяц (T=’NACH-’+ALLTRIM(STR(i)))+’.’+’txt’, где “i” это цифра три и в результате получится файл NACH-3.TXT, что означает отчет за март (3) по начислениям (NACH). Эта переменная нужна чтобы после завершения формирования отчета его, с помощью макроподстановки (EVALUATE(‘T’)или(&T)) , можно было открыть в окне для просмотра и(или) редактирования. Ниже определяется это окно, оно не определяется заранее, для того чтобы его можно было закрыть с помощью мыши, щелкнув в левом углу по прямоугольнику (Опции “SYSTEM CLOSE”).

Далее с помощью команды SET ALTERNATE TO <имя файла> создается альтернативный файл и командой SET ALTERNATE ON он открывается для внесения в него данных, одновременно командой SET CONSOLE OFF выдача на экран формируемых данных не производится.

Формирование отчета выполняется с помощью циклов, а так как некоторые данные хранятся только в полях плательщика, то используются переменные для хранения данных которые будут выводится в файл и переменные позволяющие организовать циклы это:

 Определяются два массива и переменные для вывода данных в файл:

NACH(12,1)-для хранения сумм начислений по квартплате каждой услуге, итоговых сумм и оплаты, LG(9) – для хранения сумм по начислениям по льготам каждой услуге, PL- содержит количество жильцов, L – количество льготников, FM-фамилия плательщика, OS-значение поля-OST\_K (остатка), TB-табельный номер плательщика, KV\_MET – значение поля KV\_M (квадратные метры), DAT\_OP – значения поля D\_OPL (дата оплаты).

В циклах определяются переменные для систематизированного предъявления данных:

Y-значение поля YL (улицы), D-значение поля DOM (дома) и K- значение поля KW\_RA (квартира).

В процедуре используются четыре цикла, аналогичных описанным в процедуре слияния квартплаты с льготами. То есть используются четыре цикла, где первый WHILE-цикл определяет, что циклы будут работать пока не будет попытки выйти за последнюю запись в БД (DO WHILE !EOF()). Второй предъявляет записи жильцов, живущих на одной улице (DO WHILE y=yl). Третий предъявляет записи жильцов (DO WHILE y=yl AND d=dom). И четвертый SCAN-цикл предъявляет жильцов одной квартиры (SCAN WHILE yl=y.and.dom=d.and.kw\_ra=k), внутри которого, с помощью команды управления IF…ENDIF в зависимости от категории жильцов, переменным присваиваются соответствующие значения, которые затем с помощью команды ?\?? [<*выр1*>[PICTURE<символы шаблона предъявления данных>][FUNCTION<формат выводимых данных>]][АТ<номер столбца, с которого должен начинаться вывод>][,<*выр2*>]…] заполняют открытый альтернативный файл.

После обработки всех записей БД вывод в альтернативный файл запрещается (SET ALTERNATE OFF) и он закрывается (SET ALTERNATE TO). И созданный файл открывается в окне VEDOM для просмотра(редактирования) (MODIFY COMMAND EVALUATE(‘T’) WINDOW vedom).

После закрытия окна просмотра отчета появляется окно с предложением отправить документ на печать. В случае согласия (кнопка “Да”) опрашивается готовность принтера (PRITSTATUS=.T.) и документ распечатывается (TYPE (T) TO PRINT) иначе (кнопка “Нет”) работа процедуры заканчивается и печать отчета возможна из пункта меню “Печать”.

“ОТЧЕТ”-“Списки по льготам”

Формирует отчет по льготникам не зависимо от периода действия льготы *(см. прилож. 1.7 стр 3).* Здесь переменные почти не используются, в переменные заносится только фамилия и табельный номер плательщика и переменные для предъявления данных в циклах. Данные выводятся с помощью полей льготников в БД жильцов и БД начислений, где хранятся нужные данные. Распечатка отчета показана в прилож. 3 стр. 4.

“ОТЧЕТ”-“Оплата э\энергии”

Формируется отчет по оплате за электроэнергию, исходя из данных счетчика *(см. прилож 1.7 стр.5).* Здесь используются переменные только для систематизированного предъявления данных в циклах. Данные выводятся из полей плательщиков, так как льгота по этой услуге не предусмотрена. Распечатка показана в прилож. 3 стр. 5.

“ОТЧЕТ”-“Сальдо по квартиросъемщикам”

 Формируется отчет по плательщикам с указанием суммы начислений по квартплате, сумме к оплате и суммы остатка *(см.прилож. 1.7 стр. 7)*. Здесь также используются переменные только для систематизированного предъявления данных в циклах. Распечатка показана в прилож. 3 стр. 6.

“ОТЧЕТ”-“Свод по квартплате за месяц”

Формируется отчет по общим суммам всех начислений по каждой услуге и итоговые суммы за месяц *(см. прилож. 1.7 стр. 8)*. В этой процедуре не используются циклы. Данные получаются с помощью команды математической обработки БД: CALCULATE [<границы>][WHILE<условие>][FOR<условие>]<список выражений>[TO <переменные>/TO ARRAY <массив>]. Эта команда позволяет вести математические расчеты в БД. <Список выражений> может содержать любую комбинацию внутренних для данной команды функций, которые также выполняют матаматическую обработку (AVG(<вырN>), CNT(), MAX(<выр>), MIN(<выр>), STD(<вырN>), SUM(<вырN>), VAR(<вырN>), NPV(…)). Здесь приведена полный перечень возможности для данной команды, а жирным шрифтом выделены опции, которые используются в процедуре. Так первая команда CALCULATE подчитывает общие начисления по квартплате (SUM<поле с суммой начислений квартплаты по услуге>), вторая общие начисления по льготам (SUM<поле с суммой начислений льготы по услуге>) и полученные результаты заносятся в переменные. Затем в переменные заносятся общие суммы начислений для получения организацией за предоставления услуг квартиросъемщикам за месяц. Они рассчитывается путем вычета из общих начислений квартплаты по услугам начисления льгот по услугам и полученные результаты выводятся в файл. Распечатка показана в прилож. 3 стр.7).

Пункт меню “ПЕЧАТЬ”

При выборе данного пункта активируется POPUP-меню описанное с опцией PROMPT FILES [LIKE<маска>] – элементами которого будут названия файлов, ограниченных маской (DEFINE POPUP prin\_t FROM 1,31 PROMPT FILES LIKE \*-1.TXT *см.прилож. 1.1 стр.3)*, которое формирует меню из имен файлов, созданных при выборе соответствующего пункта “ОТЧЕТА” и определение реакции меню выполняется в процедуре print3 (ON SELECTION POPUP prin\_t DO print3 WITH PROMPT(),LASTKEY()), куда передаются два параметра это-имя файла (vv - PROMPT()) и код нажатия клавиши (lk - LASTKEY()).

В процедуре *(см. прилож. 1.2 стр. 28)* в структуре DO CASE анализируются переданные параметры. Если нажата клавиша “Enter” (ВВОД), то код 13 (CASE lk=13) и осуществляется вызов соответствующего текстового файла в окно VEDOM для просмотра с возможностью редактирования (MODIFY COMMAND (vv) WINDOW VEDOM).

Если нажата клавиша “Spacebar” (ПРОБЕЛ), тол код 32 (CASE lk=32), происходит опрос готовности принтера (PRINTSTATUS()) и если он готов, файл печатается (TYPE (vv) TO PRINT), если нет – выдается соответствующее сообщение. Ниже приводятся мена файлов:

файл LG\_KW-xx.TXT*“Списки по льготам”*

файл EL\_PL-xx.TXT*“Оплата электроэнергии”*

файл SVOD-xx.TXT*“Свод по квартплате за месяц”*

*Где хх – это номер месяца, за который сформирован отчет.*

Пункт меню “ПОМОЩЬ”

Все файлы помощи хранятся в базе данных HELP.DBF, которая имеет только одно MEMO-поле. И по умолчанию имеет восемь, заполненных информацией, полей.

“ПОМОЩЬ”-“Помощь”

при выборе этого пункта командой ON SELECTION … DO HELP WIHT 6 выполняется процедура, в которой в окне открывается текст подсказки записанной в 6-ой записи MEMO-поля. Здесь описывается помощь по пунктам меню. *(см прилож. 2 рис. 21)*

“ПОМОЩЬ”-“Справка”

при выборе этого пункта командой ON SELECTION … DO HELP WIHT 3 выполняется процедура, в которой в окне открывается текст подсказки записанной в 3-ей записи MEMO-поля. Здесь описывается общее назначение и характеристика приложения.

*Работа процедуры помощи будет рассмотрена ниже.*

Создание контексно-зависимой экранной подсказки.

Каждое MEMO-поле в БД помощи (HELP.DBF), содержит конкретное описание помощи по работе с отдельными окнами. В начале работы программы, когда баз данных еще нет, описание помощи хранится в текстовых файлах. И при первом запуске приложения, с помощью языка SQL *(см. прилож. 1.3 стр. 2),* они создаются и в БД помощи последовательно в MEMO-поля переносятся содержимое текстовых файлов (APPEND MEMO HLP FROM HELP1.TXT).

Для экрана помощи определяется окно также с именем HELP, а клавиша F1 и два пункта данного меню закрепляются за процедурой с именем HELP, в которую передается один параметр К (ON KEY LABEL f1 DO help WITH k)

Этот параметр в дальнейшем будет указывать номер записи в файле HELP.DBF, где содержится нужная подсказка. В самой процедуре *(см. прилож. 1.2 стр. 21)* осуществляется перевод указателя записей в области I (в которой открыта БД помощи) на запись номер К.

Здесь же определяется окно HELP, в которое с помощью команды MODIFY MEMO I.HLP WINDOW HELP NOEDIT выводится текст подсказки. Чтобы исключить случайную возможность порчи содержимого HELP пользователем, возможность редактирования исключена опцией NOEDIT.

В процедурах работы с приложением вначале переопределяется клавиша F1 для открытия соответствующей подсказки (ON KEY LABEL f1 DO help WITH 1(2,4,5,7,8)), а по завершении работы с любым окном клавише F1 определяется параметр 6 (…WITH 6). Это позволяет из любого места программы можно вызвать свою помощь нажатием клавиши F1. Доступ к нужной записи в файле помощи происходит очень быстро, поскольку ее не приходиться искать. Для этого нужно в соответствующих местах программы делать присвоения переменной К, то есть указывать номер нужной записи из файла HELP.DBF. В процедурах, где помощь не предусмотрена, переменной К задается значение 6, для вывода общей помощи по программе, которая также закреплена за пунктом меню “Помощь”.

Пункт меню “Выход”

Для корректного закрытия БД и завершения работы необходимо проконтролировать запросом попытку пользователя выйти из программы. Данная функция будет содержать обычный запрос на подтверждение выхода, и при положительном ответе (кнопка “Да”) осуществит закрытие всех активных файлов БД, сохранит в файле M\_ZAR.MEM все переменные начинающиеся с символа “\_” (SAVE TO m\_zar ALL LIKE \_\*). В приложении зти переменные определены как глобальные и переменные нормативных ставок и после этого происходит выход из приложения (QUIT). При выборе кнопки “Нет” происходит возврат в приложение.

Заключение

В процессе работы над дипломным проектом был создан комплекс программных средств для обеспечения работы бухгалтера-расчетчика. Кроме того была разработана концепция проектирования информационно - справочных систем с применением функций экранного интерфейса, разработанных в рамках дипломного проекта. Применение специализированных функций позволило сократить время разработки основного модуля программы и обеспечило высвобождение времени на создание процедур обработки данных. В дипломном проекте представлен один из множества подходов к проектированию программного обеспечения информационного плана, дающий возможность на ранних этапах разработки учесть все нюансы будущей программы, необходимый набор функций, состав и структуру баз данных, что в дальнейшем исключает необходимость переработки уже написанных компонентов программы.

Данный дипломный проект разрабатывался не с коммерческой целью, а с целью показать возможности СУБД FoxPro, используя в приложении как можно больше команд и функций данной СУБД. Поэтому работа приложения не предусматривает сохранения и просмотра раннее сделанных расчетов и работа осуществляется только с текущим месяцем.

Хотя принцип работы приложения и язык программирования не исключают и коммерческое распространение. Достаточно лишь добавить модули сохранения данных при переходе на следующий месяц и просмотра из архивов этих данных.