**Аннотация**

Дипломный проект посвящен разработке автоматизированной информационной системы учета выбывших из стационара. Система базируется на форме №066/у Министерства Здравоохранения России, имеющей название "*Статистическая карта выбывшего из стационара*". Программа предназначена для автоматизации процесса обработки формы №066/у и формирования на ее основе отчетных форм для стационара.

Система предназначена для работы в отделениях медстатистики больниц и не требует от пользователя дополнительных знаний по программированию, она предоставляет ему удобный экранный интерфейс для работы с многофайловой базой данных.

База данных содержит информацию о выбывших больных, адекватную информации в форме №066/у МЗ России. На основании этой информации система позволяет формировать отчетные документы по фиксированным формам. Документы формируются за любой отчетный период (дни, месяцы, годы), по всему стационару или по любому его отделению.

Система прошла опытную эксплуатацию в Детской городской клинической больнице N13 им. Н.Ф. Филатова и передана для внедрения в отделении медстатистики этой больницы.

Программное обеспечение системы написано на языке программирования Clipper.

Экономико-организационная часть включает в себя анализ информации по теме дипломного проекта, расчет годового экономического эффекта и рекламу разработки.

Отчетные материалы к дипломному проекту включают пояснительную записку, 5 приложений и графический материал.

Содержание

Введение 4

1. Анализ существующих методов создания информационных систем в медицине 6

1.1. Классификация МИС 6

1.2. Методология создания автоматизированных больничных информационных систем 7

***1.2.1. Декомпозиция АБИС*** 8

***1.2.2. Интеграция компонент АБИС*** 9

1.3. Архитектура интегрированных АБИС 10

1.4. Эффективность применения АБИС 11

2. Технология работы отдела Медстатистики по обработке данных о выбывших из стационара, выбор технических и программных средств 13

2.1. Технологическая схема работы отдела 13

2.2. Формализованное описание документооборота 15

2.3. Оценка потоков информации, проходящих через отдел Медстатистики 17

3. Разработка структуры базы данных (БД) и алгоритмов обработки 18

3.1. Структура БД 18

***3.1.1. Файловая структура таблиц*** 19

***3.1.2. Сохранность БД*** 21

3.2. Алгоритм программы 22

4. Описание программной реализации 25

4.1. Описание программы 25

***4.1.1. Ввод текущей даты*** 26

***4.1.2. Работа в главном меню*** 26

***4.1.3. Ввод новых данных*** 27

***4.1.4. Редактирование ранее введенной информации*** 28

***4.1.5. Просмотр БД*** 28

***4.1.6. Удаление информации*** 29

***4.1.7. Получение отчетных форм*** 29

***4.1.8. Настройка*** 32

***4.1.9. Выход из программы*** 33

4.2. Результаты опытной эксплуатации 33

5.Экономико-организационная часть 35

5.1.Анализ потока публикаций 35

5.2. Расчет годового экономического эффекта 36

Заключение 40

Литература. 42

Приложение 1 43

“Техническое задание на разработку автоматизированной информационной системы учета выбывших из стационара (форма №066/у)” 43

Приложение 2 77

Описание применения “Автоматизированной системы учета выбывших из  
 стационара” 77

Приложение 3 83

Руководство оператора “Автоматизированной системы учета выбывших из стационара” 83

Приложение 4 92

Руководство системного программиста “Автоматизированной системы учета выбывших из стационара” 92

Приложение 5 107

Текст и описание программы: “Автоматизированная система учета выбывших из стационара” 107

# Введение

В последнее время все более возрастает интерес к использованию средств вычислительной техники в медицине. Это объясняется теми возможностями, которые несут в себе вычислительные средства, в первую очередь современные персональные ЭВМ. Эти ЭВМ являются удобным средством для автоматизации обработки информации в различных областях человеческой деятельности. Габариты персональных компьютеров позволяют разместить их непосредственно на рабочем месте пользователя, а удобство общения с ПЭВМ делает их доступными для широкого круга пользователей-непрограммистов.

Современные ПЭВМ имеют быстродействие и объем оперативной памяти достаточные для обработки больших массивов информации, их применение в таких областях медицины как, например, *Медстатистика*, позволяет значительно сократить затраты ручного труда, повысить качество и правильность работы, сократить время получения конечного результата.

Именно этой проблеме и посвящен дипломный проект. Его целью была разработка автоматизированной системы выбывших из стационара, в основе которой лежит форма №066/у Министерства Здравоохранения России: *"Статистическая карта выбывшего из стационара".*

О важности данной разработки говорит тот факт, что прикладное программное обеспечение информационных систем, необходимое учреждениям здравоохранения, редко может быть закуплено в готовом виде и так как практически все учреждения здравоохранения обладают своей спецификой.

Выполнение дипломного проекта потребовало решения следующих основных задач:

* проведение анализа особенностей и специфики автоматизированных информационных систем в медицине;
* анализ технологии работы автоматизируемого подразделения;
* определение необходимой входной и выходной информации;
* разработка структуры базы данных;
* разработка и отладка программы;
* проведение опытной эксплуатации и внедрение у заказчика;
* определение экономического эффекта от внедрения разработанных программных средств.

Содержание пояснительной записки в значительной степени определяется перечисленными задачами.

# 1. Анализ существующих методов создания информационных систем в медицине

Обеспечение здоровья населения требует постоянного сбора и хранения данных. Затраты на создание и функционирование системы сбора, хранения, обработки и передачи информации составляют значительную и постоянно возрастающую часть бюджета лечебных учреждений. По одним оценкам [1] эти затраты составляют от 10 до 20% от суммарных затрат на лечение, по другим они составляют до 33% бюджета лечебных учреждений. Для снижения этих затрат, дальнейшего совершенствования обработки и представления информации, а также для повышения эффективности управления лечебными учреждениями разрабатываются и внедряются разнообразные медицинские информационные системы (МИС).

## 1.1. Классификация МИС

Согласно [1], общим для всех МИС является рутинный систематический сбор данных, интегрируемых в информацию, обеспечивающую принятие управляющих административных и медицинских решений. В той же работе [1] описывается следующая классификация МИС:

* амбулаторные МИС;
* МИС стационаров;
* МИС социального страхования здоровья населения;
* национальные МИС.

Амбулаторные МИС обеспечивают сбор данных о каждом визите пациента к врачу, включая диагноз, оказанные медицинские услуги и предписанные назначения. Эти данные используются при повторных визитах, планирования работы, выписке счета пациенту и т.д.

МИС стационаров обеспечивают сбор данных для выписного эпикриза, в том числе краткий анамнез, диагнозы, продолжительность госпитализации, состояние при выписке и перечень оказанных медицинский услуг вместе с их стоимостью. Эти данные используются при повторной госпитализации, для учета и планирования работы подразделений и отдельных врачей, для расчетов с пациентами и т.д.

МИС социального страхования здоровья населения собирают данные обо всех медицинских услугах, стоимость которых требуется возместить из бюджета страховой компании, и обеспечивают расчеты с лечебными учреждениями.

Национальные МИС собирают данные о функционировании медицинских учреждений и обобщают эту информацию для руководства национальным сектором здравоохранения.

## 1.2. Методология создания автоматизированных больничных информационных систем

Современная автоматизированная больничная информационная система (АБИС) представляет собой тесно взаимодействующее сочетание МИС стационара и амбулаторной МИС. АБИС - это больничная медицинская и административная информационная система, в функции которой входят сбор, обработка и передача информации в присущих больнице целях.

АБИС обладает высокой сложностью и в целях управляемости, как в техническом, так и в организационном отношениях целесообразнее разделить ее на отдельные системы или подсистемы, которые можно создавать поэтапно. Время показало, что принцип поэтапного создания АБИС значительно облегчил их разработку и внедрение и имел определенные экономические выгоды. Принцип заключается в том, что вначале осуществляется декомпозиция АБИС, т.е. ее подразделение на отдельные компоненты, а затем интеграция этих компонент в единую систему.

АБИС, спроектированная по такому принципу, получила название

*"интегрированная АБИС".*

### ***1.2.1. Декомпозиция АБИС***

Существует много подходов к декомпозиции АБИС. Все они опираются на сочетание двух принципов ее разделения на отдельные взаимосвязанные компоненты: горизонтального и вертикального [1] .

Горизонтальный принцип соответствует созданию отдельных систем, ориентированных на нужды конкретного пользователя или достаточно однородной группы пользователей, например, подразделяют АБИС на административную систему, лабораторную систему и группу специализированных систем (аптечная система, система диетпитания, консультационные системы и пр.).

Вертикальный или функциональный принцип деления соответствует выделению систем, каждая из которых имеет определенный набор функций и может выполнять их по запросу любого пользователя системы. Примером может служить следующее вертикальное деление АБИС на 5 подсистем, каждая из которых наделена специфической функцией:

1. обеспечение выполнения медицинских действий;
2. эксплуатация медицинских данных в целях увеличения объема медицинских знаний в лечебном учреждении;
3. обеспечение материально-технического и медицинского снабжения;
4. оперативное планирование административной деятельности;
5. оперативное планирование лечебной деятельности.

Исторически преобладающим оказался горизонтальный подход, при котором вначале создаются автоматизированные системы отдельных подразделений и служб, которые затем объединяются в интегрированную АБИС. Так, первые интегрированные АБИС появились в результате слияния административно-финансовых систем с независимо разрабатывающимися автоматизированными системами диагностических подразделений больниц, в первую очередь клинических лабораторий и отделений лучевой диагностики.

### ***1.2.2. Интеграция компонент АБИС***

Интеграция компонент АБИС является не менее сложным предприятием, чем ее декомпозиция на отдельные компоненты.

Выделяют три основных требования к интеграции систем АБИС [1]:

* пациент не должен ощущать влияния разделения на подсистемы; назначения, предписанные ему в разных подразделениях, должны быть согласованы между собой (интеграция на уровне пациента);
* пользователь АБИС должен иметь возможность облегченного доступа (при наличии соответствующих полномочий) к функциям различных подсистем, ввод данных должен осуществляться однократно, и при этом система должна обеспечивать целостность и непротиворечивость информации (интеграция на уровне пользователей АБИС);
* информация должна храниться в единой базе данных и предоставляться по мере необходимости любой заинтересованной в ней подсистеме (техническая интеграция).

Интеграция не является самоцелью и должна вводится для улучшения функционирования больницы.

Отдельные системы интегрированной АБИС вводятся в эксплуатацию, как правило, поэтапно. Состав АБИС, сроки и последовательность ввода ее отдельных систем в эксплуатацию зависят от приоритетов, устанавливаемых руководством больницы, и имеющихся финансовых возможностей.

## 1.3. Архитектура интегрированных АБИС

Можно выделить следующие модели архитектуры АБИС:

* модель с централизованной обработкой данных;
* модель с распределенной обработкой данных.

АБИС модели с централизованной обработкой данных выполнены на базе одной или нескольких тесно связанных больших ЭВМ или супер-ЭВМ, которые выполняют централизованную обработку всех потоков информации.

Достоинством централизованной обработки данных является относительная простота организации доступа к данным о пациенте, отсутствие избыточности хранения информации. К недостаткам можно отнести необходимость значительных начальных капиталовложений, в том числе на оборудование машинного зала и стандартное программное обеспечение, а также определенные трудности развития прикладного программного обеспечения АБИС.

АБИС модели с распределенной обработкой данных выполнены на базе сети малых или больших ЭВМ, каждая из которых имеет собственную базу данных.

Основное достоинство этой модели состоит в том, что средства вычислительной техники могут приобретаться постепенно и компоненты АБИС могут быть внедрены поэтапно. Однако распределенность базы данных по нескольким ЭВМ приводит к усложнению разработки прикладного программного обеспечения АБИС и определенным трудностям обеспечения бесперебойной эксплуатации оборудования, а также эксплуатации баз данных.

Архитектура многих АБИС не укладывается в рамки описанных выше моделей, они имеют смешанную архитектуру. Например, в связи с широким распространением ПЭВМ в ряде АБИС они используются в качестве терминалов центральных ЭВМ, для решения локальных задач в нуждах отдельных подразделений больницы или даже отдельных специалистов, а также для предоставления справочно-информационных услуг. Имеются примеры совместного использования центральной ЭВМ и локальной сети ПЭВМ, предназначенной для диспетчеризации амбулаторного приема [1].

## 1.4. Эффективность применения АБИС

Будучи средством организации труда персонала больницы АБИС не оказывает прямого влияния на лечение пациента. Однако система позволяет активно влиять на процессы принятия персоналом больницы как административных, так и медицинских решений. В качестве целей создания АБИС можно привести следующие:

* прямая экономия зарплаты персонала;
* косвенная экономия за счет избавления медицинских сотрудников от рутинной канцелярской работы;
* улучшение качества медицинского обслуживания за счет большей точности, более широкого распространения информации и увеличения ее оперативности;
* уменьшение числа избыточных услуг;
* более полное использование ресурсов;
* лучшее снабжение ресурсами;
* общее улучшение качества обслуживания.

Если раньше основным назначением АБИС было улучшение управления больницей, то теперь - это улучшение качества лечения.

Внедрение АБИС не приводит к снижению бюджетных затрат, а лишь способствует уменьшению их роста. Основной потенциал рационализации функционирования больницы заключается в снижении расходов на зарплату персонала, которая составляет 60-70% бюджета больницы [1]. Такое снижение может быть достигнуто только за счет изменения в организации труда, распределения работ и ответственности за их выполнение. Поскольку наибольшие штаты и средства (за исключением собственно лечения) отвлекает обработка информации, на что расходуется до 15-25% бюджета больницы, то потенциальный экономический эффект внедрения АБИС достаточно высок.

.

# 2. Технология работы отдела Медстатистики по обработке данных о выбывших из стационара, выбор технических и программных средств

## 2.1. Технологическая схема работы отдела

Для обеспечения оперативной информацией различных категорий сотрудников больницы и формирования отчетных материалов о лечебной деятельности отделений стационара в нем существует отделение медстатистики.

Рассмотрим технологию работы медстатистики при поступлении информации из отделений стационара. На *рис.1* представлена структура функциональной связи отдела с другими подразделениями больницы при поступлении информации о выбывших больных.



*Рис. 1. Технологическая схема работы отдела медстатистики.*

В медстатистику поступают истории болезни (ИБ) с заполненными врачами *статистическими картами на выбывших из стационара* (форма №066/у). Форма №066/у заполняется на каждого выбывшего. На каждого иногороднего выбывшего заполняется дополнительно еще одна форма №066/у. В подразделении медстатистики производится сверка поступивших ИБ на выбывших с данными журнала на выбывших, умерших и переведенных (ЖВ). В ЖВ вносятся отметки о наличии ИБ на выбывших. Далее составляется справка о непоступивших ИБ на выбывших больных, которая передается обратно в отделения. Поступившие ИБ отправляются в архив на хранение. Формы №066/у складываются в папки отделений для составления отчетов о деятельности стационара. Второй экземпляр формы №066/у на иногородних больных отправляется в Боткинскую больницу для дальнейшей статобработки. Далее, на основании формы №066/у в подразделении медстатистики готовятся следующие отчетные формы для отделений стационара, бухгалтерии и администрации больницы:

* ежемесячный отчет по каждому отделению;
* годовые отчеты по всему стационару (см. Приложение 5):

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Таблицы 1-3,16-19 формируются для написания отчетов о деятельности стационара для администрации больницы и в вышестоящие организации, 4-15 - только для администрации больницы.

## 2.2. Формализованное описание документооборота

Форма №066/у является обязательным документом для любого стационара Российской Федерации и имеет полное название "Статистическая карта выбывшего из стационара". Она заполняется на каждого выбывшего и заключает в себе следующую информацию:

1. Номер истории болезни больного;
2. Фамилия И.О. больного;
3. Пол больного;
4. Дата рождения;
5. Время рождения;
6. Возраст;
7. Вес;
8. Адрес постоянного местожительства;
9. Район проживания;
10. Горожанин / селянин;
11. Государство, гражданином которого является больной;
12. Направляющая организация;
13. Причина направления;
14. Отделение стационара;
15. Профиль койки;
16. Доставлен в стационар по экстренным показаниям или в плановом порядке;
17. Через сколько часов после заболевания (получения травмы);
18. Дата поступления в стационар;
19. Время поступления в стационар;
20. Исход заболевания;
21. Дата выписки / перевода /смерти;
22. Время выписки / перевода /смерти;
23. Количество, проведенных в стационаре дней;
24. Диагноз направившего учреждения;
25. Госпитализирован впервые / повторно;
26. Диагноз стационара;
27. Хирургические операции;
28. Обследование на реакцию Вассермана;
29. Фамилия И.О. лечащего врача.

## 2.3. Оценка потоков информации, проходящих через отдел Медстатистики

В 1998 году план на пролеченных больных в ДГКБ составил 22 400 человек. На каждого пролеченного больного должна быть сформирована ф.№066/у. Объем одной этой формы составляет около 1 КБ, тогда объем поступающей информации по ф.№066/у в год составит порядка 19 МБ. При этом в медстатистику поступает и другая информация: о движении больных и коечного фонда (ф.7), а также информация из хирургического центра, районной поликлиники и травмопункта. Самый большой объем здесь имеет форма 7. Если считать, что объем одной ф.7 также равен 1 КБ, то в день объем поступившей информации по ф.7 составит 15 КБ. Если полагать, что число рабочих дней в году 273, то годовой объем информации, поступившей в медстатистику по ф.7 составит порядка 4 МБ. Тогда общий объем информации составит 23 МБ. При этом объем информации, поступающий в день, составит порядка 0.1 МБ. В стационар из медстатистики поступает порядка 0.003 МБ информации, в том случае, если администрации передается информация, сформированная на основе ф.7.

Учитывая характер работы стационара, следует предположить, что основная информация будет проходить в сети с 8 до 14 часов. При этом интенсивность потока информации составляет порядка 0.036 Кбит/с. При пропускной способности стандартной сети Ethernet 10 Мбит/с этот поток не представляет большой нагрузки на сеть. Следует ожидать, что не рассмотренные оценки потоков информации из хирургического центра, районной поликлиники и травмопункта не окажут существенного влияния на загрузку сети.

# 3. Разработка структуры базы данных (БД) и алгоритмов обработки

## 3.1. Структура БД

Для хранения информации по выбывшим больным была разработана БД, концептуальная модель которой представлена на *рис.2.*



*Рис. 2. Концептуальная модель базы данных.*

БД является распределенной - она состоит из трех таблиц:

* ТАБЛИЦЫ "DIA66" (содержащей диагнозы заболеваний больного);
* ТАБЛИЦЫ "OP66" (содержащей названия операций больного);
* ТАБЛИЦЫ "KARTA" (содержащей всю остальную информацию).

Связь осуществляется по номеру истории болезни. Каждому выбывшему соответствует одна запись в таблице "KARTA" и такое количество записей в таблицах "DIA66" и "OP66", сколько диагнозов ему проставлено и сколько операций ему было проведено соответственно. Другими словами между таблицей "KARTA" и таблицей "DIA66", а также между таблицей "KARTA" и таблицей "OP66" существует связь "один ко многим".

Для уменьшения объема, отводимого для хранения одной записи в таблице "KARTA", используется следующее: заполнение ф. №066/у в программе идет в автоматизированном режиме, т.е. все пункты ф.№066/у (где это возможно) заполняются с помощью выбора одной из нескольких альтернатив, представленных в виде меню. Совокупность всех альтернатив называется справочником. Заполнение справочника происходит на самом раннем этапе эксплуатации программы. Каждой альтернативе в справочнике соответствует уникальный номер или код (поэтому справочники иногда называют кодификаторами), т.е. между альтернативой и ее кодом существует взаимооднозначное соответствие. В виду этого в БД хранятся не сами альтернативы, а их коды. Существуют отработанные процедуры работы со справочниками, в частности разработанные сотрудниками кафедры 29.

### ***3.1.1. Файловая структура таблиц***

Каждая таблица включает в себя файл(ы) данных и файл(ы) индексов. Таблица для хранения данных по пациентам содержит файлы:

* КАRТА66.DBF и КАRТА66.NTX;
* DIA66.DBF и DIA66.NTX;
* OP66.DBF и OP66.NTX.

Структура файла KARTA66.DBF:

|  |  |  |  |
| --- | --- | --- | --- |
| ИМЯ ПОЛЯ | ТИП | ДЛИНА | СОДЕРЖАНИЕ |
| NUM\_IB | C | 7 | номер истории болезни |
| FAM | C | 30 | фамилия больного |
| F\_S\_NAME | C | 30 | имя, отчество больного |
| DATE\_B | D | 8 | дата рождения |
| HOUR\_B | N | 2 | часы рождения |
| MINS\_B | N | 2 | минуты рождения |
| OLD | N | 2 | код возраста |
| POL | N | 1 | код пола |
| MASSA | C | 6 | масса |
| PLACE\_LIV | C | 60 | адрес |
| RAION | N | 2 | код района |
| CITY\_VIL | N | 1 | код: горожанин/селянин |
| DIRECT1 | N | 2 | код направляющего учреждения |
| DIRECT2 | N | 2 | код стационара |
| STATE | N | 2 | код государства |
| WHY | N | 1 | код причины направления |
| DEPARTMENT | N | 2 | код отделения стационара |
| KOIKA | N | 2 | код профиля койки |
| PASS | N | 1 | код состояния поступления |
| TIME | N | 1 | код поступления после болезни |
| DATE\_IN | D | 8 | дата поступления в стационар |
| HOUR\_IN | N | 2 | часы поступления |
| MINS\_IN | N | 2 | минуты поступления |
| END1 | N | 1 | код исхода |
| END2 | N | 1 | код причины исхода |
| END3 | N | 2 | код учреждения для перевода |
| DATE\_END | D | 8 | дата исхода |
| HOUR\_END | N | 2 | часы исхода |
| MINS\_END | N | 2 | минуты исхода |
| OLD\_D | N | 2 | возраст на момент смерти |
| ALL\_DAY | N | 2 | количество дней в стационаре |
| DIA\_DIRECT | C | 4 | направляющий диагноз |
| NUM\_COME | N | 1 | код числа поступлений |
| RW\_DATE | D | 8 | дата анализа на RW |
| RW\_REZ | N | 1 | код результата RW |
| FAM\_DOCTOR | N | 2 | код фамилии лечащего врача |

Индексное выражение для файла: NUM\_IB.

Структура файла DIA66.DBF:

|  |  |  |  |
| --- | --- | --- | --- |
| ИМЯ | ТИП | ДЛИНА | СОДЕРЖАНИЕ |
| NUM\_IB | C | 7 | номер истории болезни |
| KOD1 | C | 1 | код заболевания |
| KOD2 | C | 1 | код диагноза |
| SHIFR | C | 4 | шифр заболевания |
| COMM1 | M | 10 | названия заболеваний |

Индексное выражение для файла: NUM\_IB+KOD1.

Структура файла OP66.DBF:

|  |  |  |  |
| --- | --- | --- | --- |
| ИМЯ | ТИП | ДЛИНА | СОДЕРЖАНИЕ |
| NUM\_IB | C | 7 | номер истории болезни |
| SHIFR | C | 4 | шифр операции |
| DATA | D | 8 | дата операции |
| COMM | M | 10 | название операции |

Индексное выражение для файла: NUM\_IB.

### ***3.1.2. Сохранность БД***

В целях поддержания сохранности и целостности БД были разработаны организационно - технические мероприятия, заключающиеся в создании программы сохранения на дискеты основных баз данных автоматизированной системы и обучения работы с ней пользователей, а также рекомендации по ее применению, заключающиеся в следующем: по окончанию сеанса работы с системой желательно сохранить вновь введенную информацию на дискеты для возможности ее восстановления, если жесткий диск выйдет из строя. При сохранении информации на дискеты переносятся все основные базы данных полностью, причем каждая база данных на свою дискету. Если объема одной дискета будет не достаточно для сохранения какой-либо базы данных, то эта база данных будет сохранена на нескольких дискетах. Программа сохранения не требует от персонала специальных навыков: необходимо только запустить ее и вставлять/вынимать дискеты, реагируя на сообщения, которые она будет выдавать на экран. Частоту запусков программы сохранения определяет сам пользователь.

## 3.2. Алгоритм программы

Система предназначена для автоматизации заполнения формы №066/у Минздрава России и генерации на ее основе типовых отчетов для стационаров.

В соответствии с ТЗ на разработку системы взаимодействие программы с пользователем должно быть реализовано по следующему алгоритму:

1) Ввод текущей даты;

2) Выбор из главного меню одной из следующих функций:

а) Ввод данных (перечень данных см. пункт 2.2) и печать;

б) Коррекция ранее введенных данных;

в) Поиск данных больного по N ИБ или по фамилии.

г) Получение отчетных форм (за отчетный период с <число, месяц, год> по <число, месяц, год>);

д) Выход из программы.

После выполнения каждой функции необходимо обеспечить возврат в главное меню и выбор любой другой функции.

При выборе функции "Выход из программы" работа программы завершается.

При выборе функции "Коррекция ранее введенных данных" пользователь может откорректировать данные, введенные ранее.

Предусмотрен также режим "Настройка программы", который должен обеспечивать адаптацию программы к изменяющимся справочникам, используемым в программе. Режим настройки должен быть скрыт от конечного пользователя программы и доступен лишь администратору БД.

Разработанный алгоритм программы приведен на *рис.3.* Алгоритм процедуры формирования отчетных документов приведен на *рис.4.*



*Рис. 3.Алгоритм программы.*



*Рис. 4. Алгоритм формирования отчетов.*

# 4. Описание программной реализации

## 4.1. Описание программы

Как уже отмечалось ранее, автоматизированная система выбывшего из стационара предназначена для автоматизации процесса заполнения формы №066/у Минздрава России и формирования на ее основе отчетных форм для стационара.

Программа обеспечивает:

* ввод, хранение, коррекцию данных по выбывшим из стационара больным;
* оформление печатных документов по данным о выбывших больных за отчетный период;
* настройку на структуру конкретного стационара.

Она ориентирована на пользователя, не являющегося профессионалом в области вычислительной техники.

Система реализована для персональных ЭВМ IBM PC XT/AT со следующим набором периферийных устройств: принтер; дисплей с платой адаптера EGA/VGA, накопитель на жестком диске объемом не менее 80 Мб. Минимальный объем свободной оперативной памяти 540 Кб.

Программа написана на языке программирования Clipper и работает под управлением операционной системой MS DOS версии 3.1 и выше.

Для запуска программы необходимо набрать в командной строке => karta и нажать клавишу ENTER.

Об успешном запуске программы свидетельствует приглашение ввести дату (см. пункт 4.1.1.).

Процесс работы с системой разбивается на несколько этапов:

* ввод текущей даты;
* работа в главном меню;
* ввод новых данных;
* коррекция введенных данных;
* формирование и печать отчетов за выбранный период;
* настройка программы на структуру конкретного стационара;
* выход из программы

### ***4.1.1. Ввод текущей даты***

При запуске программы на экране появляется заставка с надписью “СЕГОДНЯ: ” и некоторой датой. На машинах, снабженных автономными встроенными часами, будет выведена текущая дата, на других машинах выводится дата, установленная последней системной командой "DATE".

Для подтверждения даты нажмите клавишу ENTER, для изменения - наберите нужную дату и нажмите клавишу ENTER.

Можно также ввести любую дату, а не только текущую.

### ***4.1.2. Работа в главном меню***

Главное меню программы содержит следующие пункты:

* создание;
* удаление;
* редактирование;
* навигация;
* отчеты;
* выход.

Перемещение по пунктам главного меню осуществляется клавишами перемещения курсора: стрелка вправо, стрелка вниз и стрелка влево, стрелка вверх. Выбор и отказ от выбора пункта меню осуществляется клавишами ENTER и ESC соответственно. В нижней части экрана имеется подсказка о тех действиях, которые будут производиться при выборе соответствующего пункта главного меню.

### ***4.1.3. Ввод новых данных***

Если выбран пункт главного меню "*Создание*", то можно ввести данные в новую карту, выбывшего из стационара (другими словами, заполнить новую форму №066/у).

Высвечивается приглашение для введения номера истории болезни вновь выбывшего из стационара больного. Новая карта будет создана только в том случае, если информации о больном, с указанным номером истории болезни, в базе данных нет.

После этого на экране появляется текст, который содержит основные пункты формы №066/у, подлежащие заполнению (см. пункт 2.2). Движение по этому тексту осуществляется клавишами перемещения курсора, при этом по экрану перемещается "световое пятно". Для заполнения нужного пункта надо подвести "световое пятно" к данному пункту текста и нажать клавишу ENTER. Если данный пункт формы №066/у может содержать одно из наперед заданных значений, то на экран выводятся все эти значения и пользователь, используя клавиши перемещения курсора и клавишу ENTER, осуществляет выбор (используя клавишу ESC, можно выйти из данного пункта без выбора какого-либо значения). Все эти, "наперед заданные" значения хранятся в справочниках. Если данный пункт может содержать любое значение, то пользователю предлагается ввести его самому.

Возврат в основное меню происходит по нажатию клавиши F10.

Если некоторые пункты (а именно, те которые нужны для составления отчетных документов) не будут заполнены, то вновь введенная форма не будет сохранена. При попытке выхода, в этом случае, пользователю будет предложено заполнить нужные пункты.

### ***4.1.4. Редактирование ранее введенной информации***

Выбрав пункт главного меню "*Редактирование*", пользователь может произвести коррекцию любой ранее введенной формы, хранящейся в БД. Для этого ему предлагается либо ввести номер истории болезни, либо фамилию интересующего его больного.

Коррекция осуществляется по тем же правилам, что о при "Создании" (см. пункт 4.1.3.).

Для поиска №066/у формы интересующего больного, пользователь может использовать "Навигатор" БД (см. пункт 4.1.5.).

Пользователь имеет возможность получить твердую копию отредактированной им формы. Для этого ему надо, находясь в режиме "Редактирование" или "Создание", сделать следующее:

* включить принтер;
* вставить в принтер лист бумаги;
* нажать клавишу F5 .

Перед пользователем появится меню, пунктами которого являются:

1. "Печатать текущий раздел документа", т.е. тот пункт, на котором в данный момент находится курсор ("световое пятно");
2. "Печатать весь документ", т.е. всю данную №066/у форму целиком;
3. "Отказ от печати".

Клавишами перемещения курсора и клавишей ENTER пользователь осуществляет выбор.

### ***4.1.5. Просмотр БД***

Выбрав пункт "*Навигатор*", пользователь может просмотреть всю информацию, которая хранится в БД. Для этого ему предлагается меню второго уровня, используя пункты "*Текущая карта*", "*Следующая карта*" и "*Предыдущая карта*" которого, пользователь может последовательно просмотреть всю БД. При этом в нижней части экрана будет представлены номер истории болезни пациента, его фамилия и дата выписки, соответствующие очередной №066/у форме. Кроме того, пользователь может ввести условия просмотра:

* либо номер истории болезни интересующего его больного (в этом случае, пользователь будет просматривать БД, начиная с формы с введенным номером истории болезни (если она есть в БД); если формы с введенным номером истории болезни в БД; нет, то будет представлена первая форма с номером большим, чем у заданной);
* либо фамилию больного;
* либо дату выписки больного из стационара (в этом случае, пользователь будет просматривать №066/у формы больных, выписанных во введенную дату и во все последующие дни).

Возврат в основное меню - по F10 или по ESC.

После выхода из "Навигатора" текущим номер истории болезни становится тот номер, который был у последней из просмотренных форм. Это можно использовать для поиска и последующего редактирования интересующей пользователя формы или для её удаления ( см. пункт 4.1.6.).

### ***4.1.6. Удаление информации***

Выбрав этот пункт основного меню, пользователь получает возможность, путем последовательного просмотра БД, помечать интересующие его №066/у формы на удаление.

Помеченные пользователем на удаление формы после выполнения пункта "*Удаление*" физически удаляются из БД.

### ***4.1.7. Получение отчетных форм***

Блок- схема процедуры формирования отчетов была представлена ранее на рис.4.

Выбрав пункт главного меню "Отчеты", можно получить отчеты фиксированной формы на основе введенной информации. Предлагается меню со следующими пунктами:

* Месячные отчеты;
* Квартальные отчеты.

Выбрав первый пункт, можно получить месячный отчет по любому отделению стационара. Для этого пользователю предлагается указать названия отделения, по которому будет формироваться отчет. Если отделение не будет указано, то отчет формироваться не будет.

Если выбран второй пункт, то необходимо указать конкретный отчет, который надо сформировать. Выбор осуществляется с помощью следующего меню:

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Можно получить квартальные отчеты, как по всему стационару, так и по любому его отделению. Для этого пользователю предлагается определить отделение, по которому будет формироваться отчет. Если отделение не будет указано (нажата клавиша ESC), то отчет будет формироваться по всему стационару.

После выбора отчета необходимо ввести дату начала и окончания отчетного периода. Для ввода даты надо набрать чч/мм/гг на клавиатуре и нажать клавишу ENTER. Для отказа от ввода - ESC.

После обработки получается необходимая отчетная форма, которая изображается на экране и сохраняется в файле. Можно получить твердую копию отчетной формы на принтере.

Чтобы получить твердую копию на принтере необходимо вы-

полнить следующие действия:

* включить принтер;
* вставить в принтер лист бумаги;
* выбрать в меню альтернативу "Выводить отчет на печать".

Всюду для возврата в предыдущее меню используется клавиша ESC.

Некоторые отчетные документы имеют большую ширину, поэтому для их распечатки необходим широкий принтер. Ширину отчета можно определить при его просмотре после того, как он появится на экране.

### ***4.1.8. Настройка***

Настройка программы на режим конкретного использования заключается в корректировке справочников.

Режим настройки скрыт от конечного пользователя программы и доступен лишь администратору БД.

Для корректировки справочников необходимо, находясь в главном меню программы, нажать Control-F10. При этом пользователь попадает в меню со следующими альтернативами:

* выбор и изменение списка отделений стационара;
* выбор и изменение списка профилей коек;
* выбор и изменение списка направляющих организаций;
* выбор и изменение списка причин смерти;
* выбор и изменение списка районов;
* выбор и изменение списка возрастов;
* выбор и изменение списка причин перевода;
* выбор и изменение списка причин направлений;
* выбор и изменение списка стран содружества.

При выборе любого пункта меню пользователь переходит к меню добавления, удаления и переименования выбранного списка.

В меню, для добавления нового элемента необходимо нажать клавишу INS. После этого будет предложено ввести новое название. Закончив ввод, надо нажать клавишу ENTER. Для удаления элемента используется клавиша DEL.

При нажатии клавиши ENTER осуществляется выбор текущего элемента в списке с целью его переименования. Вводится новое название элемента и нажимается клавиша ENTER. ESC означает отказ от переименования.

При нажатии ESC происходит возврат в предыдущее меню.

### ***4.1.9. Выход из программы***

Для окончания работы с системой следует выбрать пункт "Выход" главного меню.

В программе не предусмотрено специальных сообщений для системного программиста. При работе программы возможна выдача сообщений исполнительной системой Clipper. Типичными сообщениями могут быть: "Not enought memory", "Memory fault" – по этим сообщениям следует удалить из оперативной памяти все резидентные программы, кроме русификаторов.

"Unable to open file <имя\_файла>" - следует проверить наличие в каталоге указанного файла и при его отсутствии скопировать с резервной дискеты.

В более сложных случаях следует записать сообщение и ситуацию, в которой оно возникло, и обратиться к системному программисту, сопровождающему программу.

## 4.2. Результаты опытной эксплуатации

Опытная эксплуатация проходила в Детской городской клинической больнице N13 им. Н. Ф. Филатова (ДГКБ N 13). В систему было заложено порядка 900 форм №066/у действительно выбывших из ДГКБ N13 больных. Ввод информации проводился сотрудниками отдела медстатистики этой больницы. В процессе работы были определенны некоторые особенности ввода информации и формирования отчетных документов. Все они были учтены в окончательном варианте программы.

На основании этих 900 форм были получены все возможные отчетные документы. Сверка их с отчетами отдела медстатистики, полученными в ручную, подтвердила правильность работы системы.

Система передана для внедрения в отделение медстатистики ДГКБ N13.

# 5.Экономико-организационная часть

## 5.1.Анализ потока публикаций

В соответствии с заданием на дипломное проектирование проводилось изучение изменения потока публикаций по теме дипломного проекта. Анализ проводился с помощью автоматизированной поисковой системы, действующей в ГПНТБ.

Для поиска использовались следующие ключевые слова:

* медицинская статистика;
* медицинская информатика;
* СУБД.

Для оценки потока публикаций был взят отрезок времени, начинающийся 1993 годом.

******Полученные результаты представлены на гистограмме изменения потока публикаций (*рис.5*).

***Рис. 5. Гистограмма изменения потока публикаций.***

Из данной гистограммы видно, что наблюдается непрерывное увеличение числа публикаций, говорящее, вероятно, о возрастании интереса к проблеме использования средств вычислительной техники в работе отдела медицинской статистики. Это, скорее всего, связано со значительным увеличением парка ЭВМ, в первую очередь - персональных, их относительной дешевизной и увеличением их производительности, что играет не последнюю роль при обработке больших объемов статистической информации.

Неослабевающий интерес к этой теме связан с теми большими возможностями, которые представляют такие средства, как СУБД и пакеты прикладных программ.

## 5.2. Расчет годового экономического эффекта

Годовой экономический эффект будем рассчитывать согласно [7]:

Э = (Эб – Эн) \* Ан, где

* Эб = Сб + Ен \* Кб - приведенные затраты на единицу работы базового образца;
* Эн = Сн + Ен \* Кн - приведенные затраты на единицу работы нового образца;
* Ен - нормативный коэффициент эффективности (Ен=0.15);
* Кб,Кн - капитальные затраты;
* Сб,Сн - себестоимость единицы работы;
* Ан - годовой объем работ.

В качестве единицы работы возьмем работу, проводимую отделом Медстатистики больницы за год. В этом случае Ан=1. В качестве Эб возмем затраты труда этого отдела без использования ЭВМ (т.о. Кб=0).

Капитальные затраты Кн представляют собой затраты на выполнение дипломного проекта Сдп, которые отражаются в величине его себестоимости [6]:

Кн = Сдп = Спр + Снакл, где

* Спр - прямые расходы на выполнение дипломного проекта;
* Снакл - накладные расходы.

Для МИФИ Снакл составляют примерно 25% от величины Спр:

Спр = Зосн + Зэвм + Осс + Оп, где

* Зосн - зарплата и стипендия основных исполнителей ДП;
* Зэвм - затраты на использования машинного времени;
* Осс - отчисления на социальное страхование;
* Оп - отчисления в пенсионный фонд.

Зосн = Зстуд \* Тдп + Зрук \* Трук + Зконс \* Тконс, где

* Зстуд, Зрук, Зконс - заработная плата студента-дипломника, его руководителя и консультанта соответственно;
* Тдп, Трук, Тконс - время, затраченное на участие в выполнении ДП этими лицами соответственно (*см. табл. 1.*).

*Таблица 1. Расчет заработной платы.*

|  |  |  |
| --- | --- | --- |
|  | Зарплата | Время работы |
| Руководитель (к. т. н., доцент) | 62 руб/час | 24 часа |
| Консультант (к. т. н. доцент) | 62 руб/час | 2 часа |
| Дипломник | XI: 800 руб. XII: 800 руб. I: 1600 руб. II: 1600 руб. | 4 месяца |

Таким образом:

Зосн=(800+800+1600+1600) + 62\*24 + 62\*2=4800+1488+124=6412.

Выполнение ДП потребовало 320 часов машинного времени. Стоимость 1 часа на PC XT составляла 40 рублей. Таким образом, Зэвм = 320\*40 = 12800 рублей.

Отчисления на социальное страхование и в пенсионный фонд составляют 37% от Зосн, следовательно:

(Осс + Оп)= 0.37 \* Зосн = 0.37\*6412 = 2372.44 рублей

Таким образом:

Спр = Зосн + Зэвм + Осс + Оп =6412+12800+2372.44=21584.44;

Спр=21584.44 рублей;

Снакл=0.25\*Спр=0.25\*21584.44=5396.11 рублей;

Кн = Сдп = Спр + Снакл = 21584.44 + 5396.11 = 26980.55;

Кн=26981 рубль.

Себестоимость эксплуатации Сн представляет собой стоимость единицы работы, сделанной новым образцом. Эта стоимость включает в себя зарплату оператора, работающего с программой (новым образцом) и стоимость машинного времени.

Зарплата оператора Зоп вычисляется как произведение времени работы с программой в часах Траб с почасовой оплатой оператора (35 руб./час).

Стоимость машинного времени определяется как произведение Траб со стоимостью одного часа машинного времени (40руб/час).

Траб определяется из содержания самой работы, которую выполняет программа:

* формирование ежемесячных отчетов по 14 отделениям больницы (среднее время формирования одного отчета 15 минут);
* формирование 19 различных годовых отчетов (среднее время формирования одного годового отчета 45 минут).

Следовательно, Траб= 12\*(14\*15/60) + 19\*45/60 = 42+14.25 = 56.25 часов.

Таким образом, Зоп= 56.25\*35 = 1968.75 рублей, а стоимость машинного времени 56.25\*40=2250 рублей. Отчисления в фонд социального страхования и в пенсионный фонд составят 37% от Зоп, т.е. 1968.75 \* 0.37 = 728.44 рублей.

С учетом накладных расходов получаем себестоимость работы программы:

Сн = (1968.75+2250+728.44) + 0.25\*(1968.75+2250+728.44) =

= 6183.99 рублей.

Сн = 6184 рублей.

Для оценки Эб рассчитаем затраты отдела медстатистики стационара без использования ЭВМ. В отделе работает пять человек:

* 4 медрегистратора (ставка 4000 руб./месяц);
* 1 врач (ставка 6000 руб./месяц).

Следовательно, затраты за год составляют:

12 \* (4 \* 4000 + 6000) \* (1 + 0.37) = 361680 рублей.

Разработанная система покрывает около 40% всей работы отдела, следовательно, Эб= 0.40\*361680 = 144672 рублей.

В итоге получаем:

Э = (Эб – Эн)\*Ан = (144672 - (6184 + 0.15\*26981)) \* 1= 134440.85 рублей.

***Годовой экономический эффект*** от использования разработанной системы составит ***134440 рублей.***

Лимитная цена разработанной системы:

Ц = 0.15 \* Кн + П \* Кн + НДС \* Кн + Сн, где

* Кн - себестоимость разработки;
* П - прибыль (0.30);
* НДС- налог на добавленную стоимость (0.20);
* Сн - себестоимость эксплуатации.

Ц= 0.15\*26981 + 0.30\*26981 + 0.20\*26981 + 6184 =23721.65

Следовательно, ***Лимитная цена разработки*** составляет ***23 721 рубль***.

.

# Заключение

В результате проделанной работы были получены следующие результаты:

* проведен анализ существующих методов создания информационных систем в медицине;
* проведен анализ работы отдела медстатистики стационара и анализ потоков информации, проходящей через него, на основании чего сделан вывод о возможности использования в работе отдела стандартной сети Ethernet, с пропускной способностью 10 Мбит/с;
* учитывая объем хранимой информации, потоки запросов на ввод/вывод, допустимую скорость обработки, в качестве технических средств выбраны ПЭВМ IBM PC XT/АТ со следующим набором периферийных устройств: принтер, дисплей с платой адаптера EGA/VGA, накопитель на жестком диске объемом не менее 40 МБ, и минимальным объемом свободной оперативной памяти не менее 540 КБ; а в качестве программных средств – язык программирования Clipper;
* составлено формализованное описание документооборота (форма №066/у МЗ России);
* разработана концептуальная модель базы данных, содержащая в себе информацию, адекватную форме №066/у МЗ России и состоящую из трех баз данных, связанных по номеру истории болезни;
* разработаны алгоритмы обработки;
* разработанная с использование языка Clipper Summer'87 программа занимает в виде загрузочного модуля 310 КБ ОП;
* разработаны организационно- технические мероприятия, направленные на поддержание сохранности и целостности БД;
* определен годовой экономический эффект от внедрения разработанной автоматизированной системы, который составил 134 440 рублей.

Система прошла опытную эксплуатацию в Детской городской клинической больнице N13 им. Н.Ф.Филатова и передана для внедрения в отделение медстатистики этой больницы.

# Литература

1. Clipper Summer'87. Комплект документации.
2. Дейт К.Дж.”Введение в системы баз данных”:  
   Пер. с англ. - К: Диалектика,1998. - 784с.: ил.
3. Каррабис Дж. - Д. “Программирование на dBase III plus”:  
   М: Финансы и статистика, 1991. - 240с.: ил.
4. Мартин Дж. “Организация баз данных в вычислительных системах”:  
   пер. с англ. - М:Мир, 1990. - 660с.:ил.
5. Серия "Обзоры по важнейшим проблемам медицины":  
   выпуск N 5: "Использование автоматизированных больничных информационных систем за рубежом":   
   М: Медицина и здравоохранение, 1998.
6. Тригер Д.Я. “Технико-экономическое обоснование и информационное обеспечение дипломных проектов (в области физики и вычислительной техники)”:  
   Учебное пособие - М: МИФИ, 1982. - 44с.
7. Эриванский Ю.А. “Экономическое обоснование эффективности программных средств вычислительной техники”:  
   М: МИФИ, 1990. - 67с.

# Приложение 1

## “Техническое задание на разработку автоматизированной информационной системы учета выбывших из стационара (форма №066/у)”

***1. Введение.***

Автоматизированная система учета выбывших из стационара (условное обозначение КАРТА) предназначена для автоматизации заполнения формы №066/у Минздрава России и генерации на ее основе типовых отчетов для стационаров.

***2. Основание для разработки.***

Основанием для разработки является договор N 12-1-99 на создание научно-технической продукции между ДГКБ N13 и ИТЦ ИА.

***3. Назначение разработки.***

Указано во введении.

***4. Требования к программе.***

*4.1. Функциональные характеристики.*

Программа должна выполнять функции ввода, хранения и обработки информации о выбывшем больном и формирования ежедневных, месячных и годовых отчетов по фиксированным формам.

В программе должен быть обеспечен:

* ввод непротиворечивой и целостной информации;
* удобство пользователя, не имеющего специальных навыков работы с вычислительной техникой при заполнении формы №066/у, высокая скорость заполнения за счет повышения наглядности и упрощения процесса ввода отдельных элементов информации;
* получение фиксированных отчетных форм за время, приемлемое для пользователя (не более 5 минут для ежедневного отчета, не более 1 часа для месячного отчета и не более 1 рабочего дня для годового отчета).

Контроль непротиворечивости и целостности информации должен состоять из двух этапов:

1. контроля непосредственно при вводе (проверка ограничений на элемент информации, вводимый в данный момент);
2. контроля информации после завершения ввода (проверка ограничений на совокупность элементов информации).

Для удобства пользователя в качестве базы для экранного интерфейса необходимо использовать полноэкранное редактирование полей БД.

Входной информацией программы является информация из формы №066/у (см. Приложение 1.1).

Выходной информацией программы являются таблицы:

* месячного отчета (см. Приложения 1.2 и 1.3);
* годового отчета, включающие в себя:

1. Состав больных в стационаре, сроки и исход лечения (см. табл.1 ф.№14);
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения (см. табл.2 ф.№14);
3. Хирургическая работа учреждений (см. табл.3 ф.№14);
4. Распределение больных по возрасту и району (см. Приложение 4);
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни (см. Приложение 1.5);
6. Больные, переведенные в другие лечебные учреждения (см. Приложение 1.6);
7. Нозология больных, переведенных из других стационаров (см. Приложение 1.7);
8. Нозологическая таблица умерших (см. Приложение 1.8);
9. Нозология больных умерших по возрастам (см. Приложение 1.9);
10. Нозология умерших в возрасте от 0 до 6 дней жизни (см. Приложение 1.10);
11. Операции умерших (см. Приложение 1.11);
12. Нозология умерших до суток (см. Приложение 1.12);
13. Нозология инфекционных заболеваний (см. Приложение 1.13);
14. Распределение инфекционных заболеваний по отделениям (см. Приложение 1.14);
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы (см. Приложение 1.15);
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Таблицы 1-3,16-19 формируются для написания отчетов о деятельности стационара для администрации больницы и в вышестоящие организации, 4-15 - только для администрации больницы.

Взаимодействие программы с пользователем должно быть реализовано по следующему алгоритму:

1. Ввод текущей даты.
2. Выбор из главного меню одной из следующих функций:
   1. Ввод данных (перечень данных см. в Приложении 1.1) и печать;
   2. Коррекция ранее введенных данных;
   3. Поиск данных больного по №ИБ или по Фамилии И.О;
   4. Получение отчетных форм (за отчетный период с <число, месяц> по <число, месяц>);
   5. Выход из программы.

После выполнения каждой функции необходимо обеспечить возврат в главное меню и выбор любой другой функции. При выборе функции "Выход из программы" работа программы завершается.

При выборе функции "Коррекция ранее введенных данных" пользователь может откорректировать данные, вводимые ранее.

Предусмотрен также режим "Настройка программы", который должен обеспечивать адаптацию программы к изменяющимся справочникам, используемым в программе, а именно:

1. Добавление и удаление списка отделений стационара;
2. Добавление и удаление списка профилей коек;
3. Добавление и удаление списка направляющих организаций;
4. Добавление и удаление списка причин смерти;
5. Добавление и удаление списка операций;
6. Добавление и удаление списка районов;
7. Добавление и удаление списка возрастов;
8. Добавление и удаление списка учреждений, откуда и куда переводятся больные ДГКБ №13;
9. Добавление и удаление причин перевода больного;
10. Добавление и удаление причин направлений больного;
11. Добавление и удаление стран содружеств и их шифров.

Режим настройки должен быть скрыт от конечного пользователя программы и доступен лишь администратору БД.

Концептуальная модель БД ф. №066/у представлена в Приложении 14.

*4.2. Требования к надежности.*

Разработанная программа должна обладать средствами защиты от ошибочных действий персонала.

*4.3. Условия эксплуатации.*

Программа должна быть ориентирована на пользователя, не имеющего специальных навыков работы в области вычислительной техники и программирования.

*4.4. Требования к составу и параметрам технических средств.*

Программу необходимо реализовать на базе ПЭВМ типа IBM PC/XT/AT с набором периферийных устройств (принтер, дисплей с платой адаптера EGA/VGA, накопителем на жестком диске с объемом не менее 40 МБ).

***5. Требования к программной документации.***

Документация на программу должна включать объединенный документ: описание программы и текст программы, эксплуатационные документы в согласованном с пользователем объеме.

**Приложение 1.1.**

Данные о больном, фиксируемые в форме №066/у:

1. Номер истории болезни;
2. Фамилия И.О;
3. Пол:
   1. Мужской;
   2. Женский;
4. Дата и время рождения (год, месяц, число, время);
5. Возраст:
   1. 0-6 дней.
   2. 7-28 дней;
   3. 29 дней-3 мес.;
   4. 4-6 мес.;
   5. 7-9 мес.;
   6. 10-12 мес.;
   7. 1год;
   8. 2-3 года;
   9. 4-7 лет;
   10. 8-14 лет;
   11. 15 лет;
6. Вес (кг/г);
7. Проживает:
   1. Адрес постоянного местожительства;
   2. Район:
      1. Москва;
      2. Московская область;
      3. Иногородний;
      4. Сельский житель;
      5. Иностранец;
8. Житель:
   1. Города;
   2. Села;
9. Государство
10. Кем направлен:
    1. Род. дома, род. отделения больниц;
    2. Клиники, больницы:
       1. ДГКБ №32;
       2. ДГКБ №67;
       3. ДГКБ №1;
       4. ДКБ №11;
       5. ДГКБ №3;
       6. ГКБ №13;
       7. ДКБ №2;
       8. ДГКБ №9;
       9. 5 ДИБ;
       10. ДИБ 12;
       11. ДИБ 6;
    3. Поликлиники;
    4. Консультации;
    5. Минздрав России;
    6. Скорая;
    7. Самотек;
    8. ГУЗМ;
    9. Военная часть;
    10. Милиция;
    11. Повторные больные;
    12. ГУЗМО;
    13. РЗГВ;
    14. Прочее;
11. Причина:
    1. Оперативное вмешательство;
    2. На гемодиализ;
    3. В реанимационное отделение;
    4. Прочее;
12. Отделение:
    1. Микрохирургия;
    2. Хирургия новорожденных;
    3. Урология;
    4. Травматология;
    5. Гнойной хирургии;
    6. Торакальное;
    7. Искусственная почка;
    8. Нефрология;
    9. Терапия старшего возраста;
    10. Токсикология;
    11. Патология новорожденных;
    12. Лор;
    13. Реанимация;
    14. Неонатальный центр;
13. Профиль коек:
    1. Хирургический;
    2. Терапевтический;
    3. Реологический;
    4. Реанимационный;
14. Доставлен в стационар по экстренным показаниям:
    1. Да;
    2. Нет;
15. Через сколько часов после заболевания (получения травмы):
    1. в первые 6 часов;
    2. 7-24 ч.;
    3. позднее 24 ч.;
16. Дата и время поступления в стационар (год, месяц, число, время)
17. Исход заболевания:
    1. выписан;
    2. умер;
    3. переведен;
18. Если переведен:
    1. Куда:
       1. ДГКБ №32;
       2. ДГКБ №67;
       3. ДГКБ №1;
       4. ДКБ №11;
       5. ДГКБ №3;
       6. ГКБ №13;
       7. ДКБ №2;
       8. ДГКБ №9;
       9. 5 ДИБ;
       10. ДИБ 12;
       11. ДИБ 6;
    2. Причины перевода:
       1. по профилю;
       2. для дальнейшего лечения;
       3. оперативное вмешательство;
19. Дата и время выписки, перевода, либо смерти (год, месяц, число, час);
20. Проведено дней (вычисляется автоматически как разность между датой выписки, перевода, либо смерти и датой поступления);
21. Диагноз направившего учреждения по МКБ;
22. Госпитализирован в данном году по поводу данного заболевания:
    1. Впервые;
    2. Повторно;
23. Диагноз стационара по МКБ:
    1. Основной;
       1. клинический;
       2. патологоанатомический;
    2. осложнения;
       1. клинический;
       2. патологоанатомический;
    3. сопутствующие заболевания;
       1. клинический;
       2. патологоанатомический;
24. Причина смерти:
    1. основное заболевание;
    2. осложнения основного заболевания;
    3. другие важные заболевания, способствовавшие смертельному исходу, но не связанные с заболеванием или его осложнениями, послужившим непосредственной причиной смерти;
25. Хирургические операции:
    1. Дата (число, месяц, год);
    2. Название операции (берется из каталога операций в ф.№14 - отчет о деятельности стационара и каталога операций из ф.№30-здрав - отчет ЛПУ)
26. Обследование на реакцию Вассермана:
    1. Число, месяц, год;
    2. Результат:
       1. Положительный;
       2. Отрицательный;
27. Фамилия, имя, отчество лечащего врача.

***Приложение 1.2.***

Месячная таблица для отделений “Хирургия новорожденных” и “Патология новорожденных” должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе.

По каждому названию болезни:

1. Возраст:
   1. 0 - 6 дней;
   2. 7 - 14 дней;
   3. 15- 21 дня;
   4. 22- 28 дней;
   5. 29 дней - 3 мес.;
   6. 4 - 6 мес.;
   7. 7 - 9 мес.;
   8. 10 - 12 мес.;
2. Всего до одного года:
   1. Всего;
   2. В т.ч. сельских жителей;
   3. К/Д;
   4. В т.ч. сельских жителей;
3. Выписано:
   1. Всего;
   2. Москва;
   3. Московская область;
   4. Иногородние;
   5. Сельские жители;
   6. Иностранцы;
4. Первично/Вторично;
5. Направляющие организации.

***Приложение 1.3.***

Месячная таблица для всех отделений стационара (кроме отделений “Хирургия новорожденных” и “Патология новорожденных”) должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе. По каждому названию болезни:
5. Возраст:
   1. 29 дней. - 3 мес.;
   2. 4 - 6 мес.;
   3. 7 - 9 мес.;
   4. 10 - 12 мес.;
6. Всего до одного года:
   1. Всего;
   2. В т.ч. сельских жителей;
   3. К/Д;
   4. В т.ч. сельских жителей;
7. Выписано до 14 лет:
   1. 1 год;
   2. 2 года;
   3. 3- 7 лет;
   4. 8- 17 лет;
   5. Всего;
   6. В т.ч. сельские жители;
   7. К/Д;
   8. В т.ч. сельские жители;
8. 15 лет и старше:
   1. Всего;
   2. В т.ч. сельские жители;
   3. К/Д;
   4. В т.ч. сельские жители;
9. Выписано всего:
   1. Всего;
   2. Москва;
   3. Московская область;
   4. Иногородние;
   5. Сельские жители;
   6. Иностранцы;
   7. К/Д;
10. Первично/Вторично;
11. Направляющие организации.

***Приложение 1.4.***

Таблица распределения выбывших по возрасту и району за год должна включать следующие графы:

1. Возраст:
   1. 0 - 6 дней;
   2. 7 - 28 дней;
   3. Всего новорожденных;
   4. 29 дней - 3 мес.;
   5. 4 - 6 мес.;
   6. 7 - 9 мес.;
   7. 10 - 12 мес.;
   8. Всего до одного года;
   9. Один год;
   10. 2 - 3 года;
   11. 4 - 7 лет;
   12. 8 - 14 лет;
   13. 15 лет;
2. Всего выбыло;
3. Число койко-дней выбывших;
4. Число койко-дней умерших;
5. Выбыло москвичей;
6. По каждому возрасту исход:
   1. Выбыло домой;
   2. Переведено в другие учреждения;
   3. Умерло;
7. Для каждого вида исхода распределение по районам:
   1. Москва;
   2. Московская область;
   3. Иногородние;
   4. Сельские жители;
   5. Иностранцы;
8. Общее число больных;
9. Общее число больных из прикрепленных районов;
10. Общее число иногородних больных;
11. Процент от общего числа больных из прикрепленных районов;
12. Процент от общего числа больных иногородних;

***Приложение 1.5.***

Таблица "Состав больных выбывших в возрасте от 0 до 6 суток жизни" должна включать следующие графы:

1. Наименование болезни и шифр по МКБ:
   1. Острые распираторные инфекции 460 – 466;
   2. Пневмония 480-486;
   3. Инфекции кожи и подкожной клетчатки 680 – 686;
   4. Врожденные аномалии 740 – 759;
   5. Внутричерепная родовая травма 767.6;
   6. Другие родовые травмы 767.4-7;
   7. Внутриматочная гипоксия и асфиксия при родах 768;
   8. Синдром дыхательных расстройств и другие распираторные состояния 769, 770.1-9;
   9. Врожденные пневмонии 770.0;
   10. Инфекции, специфичные для перинатального периода 771.0-7;
   11. Сепсис новорожденных 771.8-часть;
   12. Гемолитическая болезнь, обусловленная иммунизацией 773;
   13. Другие болезни перинатальной желтухи 774;
   14. Прочие болезни плода и новорожденного;
2. Количество детей с массой тела при рождении:
   1. До 1000 грамм;
   2. Большей либо равной 1000 грамм.
   3. Для новорожденных с массой тела как меньшей 1000 грамм, так и большей либо равной 1000 грамм.
3. Поступило больных в первые 0 - 6 суток после рождения;
4. Умерло:
   1. Всего;
   2. В том числе в первые 0 - 6 суток после рождения.

***Приложение 1.6.***

Таблица "Больные, переведенные в другие лечебные учреждения" должна включать следующие графы:

1. Наименование классов заболеваний;
2. Шифр по МКБ;
3. Общее количество больных, переведенных в другие лечебные учреждения по каждому классу заболеваний;
4. Количество больных до года, переведенных в другие лечебные учреждения по каждому классу заболеваний;
5. Причины перевода по каждому классу заболеваний.

***Приложение 1.7.***

Таблица "Нозология больных, переведенных из других лечебных учреждений" должна включать следующие графы:

1. Название болезни;
2. Шифр по МКБ; По каждому названию болезни
3. Количество детей:
   1. до 28 дней;
   2. до года;
   3. всего;
4. Причины направления в ДГКБ:
   1. оперативное вмешательство;
   2. на гемодиализ;
   3. в реанимационное отделение;
   4. прочее;
5. Из каких учреждений направлены больные.

***Приложение 1.8.***

Таблица “Нозология умерших” должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе.

По каждому названию болезни.

1. Пол:
   1. Всего мальчиков;
   2. Всего девочек;

***Приложение 1.9.***

Таблица "Нозология умерших больных по возрастам" должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе. По каждому названию болезни:
5. Пол:
   1. Всего мальчиков;
   2. Всего девочек;
6. Возраст:
   1. 0 - 6 дней;
   2. 7 - 28 дней;
   3. Всего новорожденных;
   4. 29 дней - 3 мес.;
   5. 4 - 6 мес.;
   6. 7 - 9 мес.;
   7. 10 - 12 мес.;
   8. Всего до одного года;
   9. Койко-дни больных до одного года;
   10. Один год;
   11. 2 - 3 года;
   12. 4 - 7 лет;
   13. 8 - 14 лет;
   14. 15 лет;
   15. Всего умерло;
   16. Койко-дни всех умерших;
7. Кем направлен:
   1. Род. Дома;
   2. Клиники, больницы;
   3. Поликлиники;
   4. Консультации;
   5. Минздрав России;
   6. Скорая;
   7. Самотек;
   8. ГУЗМ;
   9. Военная часть;
   10. Милиция;
   11. Повторные больные;
   12. ГУЗМО;
   13. РЗГВ;
   14. Прочее;
8. Районы:
   1. Москва;
   2. Московская область;
   3. Иногородние;
   4. Сельские жители;
   5. Иностранцы;
9. Сколько дней пробыл в больнице:
   1. До суток;
   2. Одни сутки;
   3. Двое суток;
   4. Трое суток;
   5. Свыше трех суток;

***Приложение 1.10.***

Таблица “Нозология умерших в возрасте от 0 до 6 дней жизни” должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе. По каждому названию болезни.
5. Пол:
   1. Всего мальчиков;
   2. Всего девочек;
6. Возраст:
   1. 0 - 1 дня;
   2. 2 - 3 дней;
   3. 4 - 6 дней;
   4. Всего умерло;
7. Кем направлен:
   1. Род. Дома;
   2. Клиники, больницы;
   3. Поликлиники;
   4. Консультации;
   5. Минздрав России;
   6. Скорая;
   7. Самотек;
   8. ГУЗМ;
   9. Военная часть;
   10. Милиция;
   11. Повторные больные;
   12. ГУЗМО;
   13. РЗГВ;
   14. Прочее;
8. Районы:
   1. Москва;
   2. Московская область;
   3. Иногородние;
   4. Сельские жители;
9. Пробыло в стационаре:
   1. До суток;
   2. Сутки;
   3. Двое суток;
   4. Трое суток;
   5. Четверо суток;
   6. Пять суток;
   7. Шесть суток.

***Приложение 1.11.***

Таблица "Операции умерших" должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе.

По каждому названию болезни;

1. Количество оперированных больных всего;

На нервной системе:

* 1. На головном мозге;
  2. На периферийной нервной системе;

1. На эндокринной системе:
   1. Из них тиреотомии;
2. Операции на органах зрения:
   1. По поводу глуакомы;
   2. По поводу энуклеации;
   3. По поводу катаракты;
3. Операции на органах уха, горла, носа:
   1. Из них на ухе;
4. Операции на органах дыхания:
   1. На миндалинах и аденоидах;
   2. Пульмонэктомия;
   3. Резекция доли легкого;
   4. Резекция сегмента легкого;
5. Операции на сердце:
   1. На открытом сердце;
   2. Имплантации кардиостимулятора;
6. Операции на сосудах:
   1. Операции на артериях;
   2. Операции на венах;
7. Операции на органах брюшной полости:
   1. На желудке по поводу язвенной болезни;
   2. По поводу острой непроходимости кишечника;
   3. По поводу острого аппендицита;
   4. По поводу прободной язвы желудка;
   5. По поводу желудочно-кишечного кровотечения;
   6. По поводу ущемленной грыжи;
   7. По поводу острого холецистита;
   8. По поводу острого панкреатита;
   9. Аппендиктомии при хроническом аппендиците;
   10. Грыжеиссечение при неущемленной грыже;
   11. Холецистомии при хронических холециститах;
   12. Липафотомии диагностические;
8. Операции на почках и мочеточниках;
9. Операции на предстательной железе;
10. Операции на женских половых органах:
    1. Выскабливание матки (кроме аборта);
    2. По поводу стерилизации женщин;
11. Акушерские операции:
    1. По поводу внематочной беременности;
    2. Наложение щипцов;
    3. Вакуум-экстрация;
    4. Кесарево сечение (кроме малого влагалищного);
    5. Плодоразрушающие;
    6. Аборт;
12. Операции на костно-мышечной системе:
    1. На костях и суставах;
13. Операции на молочной железе;
14. Операции на коже и подкожной клетчатке:
    1. Прочие операции;
15. Для каждой операции число операций;
16. В том числе детям до 14 лет включительно.

***Приложение 1.12.***

Таблица "Нозология умерших до суток" должна включать следующие графы:

1. Шифр классов болезней;
2. Названия классов болезней;
3. Шифр болезни в классе;
4. Название болезни в классе. По каждому названию болезни.
5. Пол:
   1. Всего мальчиков;
   2. Всего девочек;
6. Возраст:
   1. 0 - 6 дней;
   2. 7 - 14 дней;
   3. 15 - 21 дня;
   4. 22 дней - 28 дней;
   5. Всего новорожденных;
   6. 29 дней - 3 мес.;
   7. Всего умерло;
7. Кем направлен:
   1. Род. Дома;
   2. Клиники, больницы;
   3. Поликлиники;
   4. Консультации;
   5. Минздрав России;
   6. Скорая;
   7. Самотек;
   8. ГУЗМ;
   9. Военная часть;
   10. Милиция;
   11. Повторные больные;
   12. ГУЗМО;
   13. РЗГВ;
   14. Прочее;
8. Время пребывания в больнице:
   1. 0-30 мин;
   2. 30 мин;
   3. 1 час;
   4. 1-3 ч;
   5. 10-24 ч;
   6. более 24 ч;
9. Койко-дни умерших.

***Приложение 1.13.***

Таблица "Нозология инфекционных заболеваний" должна включать следующие графы:

1. Шифр:
   1. Общее количество больных выбывших из стационара;
   2. Количество больных в возрасте до 1 года, выбывших из стационара;
   3. Количество койко-дней проведенное выписанными больными;
   4. Количество умерших;
   5. Количество умерших до года;
   6. Количество койко-дней умерших;
   7. Летальность от общего числа умерших.

***Приложение 1.14.***

Таблица "Распределение инфекционных заболеваний по отделениям" должна включать следующие графы:

1. Шифр и название болезни;
2. Отделение, в котором зарегистрировано данное заболевание.

***Приложение 1.15.***

Таблица “Распределения выбывших иногородних больных по каналам госпитализации и отделениям больницы” должна включать следующие графы:

1. Всего выписанных больных из больницы:
   1. из Москвы;
   2. иностранцев;
2. Наименование страны содружества;
3. Шифр страны;

По каждой стране содружества:

1. Направляющее учреждение;
2. Отделения больницы;
3. Койко-дни;
4. Всего выписанных иногородних больных;
5. Процент иногородних больных от числа выписанных больных.

.

# Приложение 2

## Описание применения “Автоматизированной системы учета выбывших из стационара”

.***1.Назначение программы.***

Автоматизированная система учета выбывших из стационара (условное обозначение КАРТА) предназначена для автоматизации процесса заполнения формы №066/у Минздрава России и формирования на ее основе отчетных форм для стационара.

Программа позволяет во много раз сократить время получения справочной информации по выбывшим больным за отчетный период, практически исключить ошибки, возникающие при оформлении документов, упростить процедуру коррекции данных.

Программа обеспечивает:

1. Ввод, хранение, коррекцию данных по выбывшим из стационара больных;
2. Оформление печатных документов по данным о выбывших больных за отчетный период;
3. Настройку на структуру конкретного стационара.

Программа ориентирована на пользователя, не являющегося профессионалом в области вычислительной техники.

***2.Условия применения.***

Программа КАРТА реализована для персональных ЭВМ IBM PC XT/АТ со следующим набором периферийных устройств: принтер, дисплей с платой адаптера EGA/VGA, накопитель на жестком диске объемом не менее 40 МБ. Минимальный объем свободной оперативной памяти 540 КБ.

Программа работает под управлением операционной системы MS DOS версии 3.1 и выше.

Программа реализована на языке программирования Clipper Summer'87.

***3.Описание задачи.***

Программа обеспечивает ввод и хранение следующих данных:

1. Номер истории болезни больного;
2. Фамилия И.О. больного;
3. Пол больного;
4. Дата рождения;
5. Время рождения;
6. Возраст;
7. Вес;
8. Адрес постоянного местожительства;
9. Район проживания;
10. Горожанин / селянин;
11. Государство, гражданином которого является больной;
12. Направляющая организация;
13. Причина направления;
14. Отделение стационара;
15. Профиль койки;
16. Доставлен в стационар по экстренным показаниям / нет;
17. Через сколько часов после заболевания (получения травмы);
18. Дата поступления в стационар;
19. Время поступления в стационар;
20. Исход заболевания;
21. Дата выписки / перевода /смерти;
22. Время выписки / перевода /смерти;
23. Количество, проведенных в стационаре дней;
24. Диагноз направившего учреждения;
25. Госпитализирован впервые / повторно;
26. Диагноз стационара;
27. Хирургические операции;
28. Обследование на реакцию Вассермана;
29. Фамилия И.О. лечащего врача.

Программа обеспечивает формирование отчетных документов:

* месячных по любому отделению стационара;
* годовых (квартальных) как по всему стационару, так и по любому его отделению:

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Таблицы 1-3,16-19 формируются для написания отчетов о деятельности стационара для администрации больницы и в вышестоящие организации, 4-15 - только для администрации больницы.

Различия между месячными и квартальными отчетами наблюдаются только в составе информации, выводимой в таблицы. И те, и другие отчеты могут формироваться за любой период.

Программа обеспечивает использование и настройку следующих видов справочников:

1. Справочник названий отделений стационара;
2. Справочник названий профилей коек;
3. Справочник названий направляющих организаций;
4. Справочник причин смерти;
5. Справочник названий районов;
6. Справочник возрастов;
7. Справочник названий учреждений, откуда и куда переводятся больные;
8. Справочник причин перевода больного;
9. Справочник причин направлений больного;
10. Справочник названий стран содружества и их шифров.

***4. Входная и выходная информация программы.***

***4.1 Входная информация.***

Входной информацией программы является следующая:

* данные, вводимые пользователем;
* данные, хранящиеся в базе данных по пациентам;
* текущая системная дата;
* данные, хранящиеся в справочных базах данных;

***4.2 Выходная информация.***

Выходной информацией программы является следующая:

* данные, введенные пользователем в базу данных по пациентам;
* отчетные документы.

# Приложение 3

## Руководство оператора “Автоматизированной системы учета выбывших из стационара”

.***1.Введение.***

Автоматизированная система учета выбывших из стационара (условное обозначение КАРТА) предназначена для автоматизации процесса заполнения формы №066/у Минздрава России и формирования на ее основе отчетных форм для стационара.

Программа обеспечивает:

* Ввод, хранение, коррекцию данных по выбывшим из стационара больным;
* Оформление печатных документов по данным о выбывших больных за отчетный период;
* Настройку на структуру конкретного стационара.

Программа ориентирована на пользователя, не являющегося профессионалом в области вычислительной техники.

***2.Условия выполнения программы.***

Программа "КАРТА" реализована на персональных ЭВМ IBM PC XT/AT со следующим набором периферийных устройств: принтер, дисплей с платой адаптера EGA/VGA, накопитель на жестком диске объемом не менее 80 Мб. Минимальный объем свободной оперативной памяти 540 Кб.

Программа работает под управлением операционной системы MS DOS версии 3.1 и выше.

***3.Выполнение программы.***

***3.1 Запуск программы.***

Для запуска программы необходимо набрать в командной строке =>karta и нажать клавишу ENTER.

Об успешном запуске программы свидетельствует приглашение ввести дату (см пункт 3.2.1).

***3.2 Выполнение программы.***

Процесс работы с программой разбивается на несколько этапов:

* ввод текущей даты;
* работа в главном меню;
* ввод новых данных;
* коррекция введенных данных;
* формирование и печать отчетов за выбранный период;
* настройка программы на структуру конкретного стационара;
* выход из программы.

***3.2.1 Ввод текущей даты***

При запуске программы на экране появляется заставка с надписью “СЕГОДНЯ: “ и некоторой датой. На машинах, снабженных автономными встроенными часами, будет выведена текущая дата, на других машинах выводится дата, установленная последней системной командой "DATE". Для подтверждения даты нажмите клавишу ENTER, для изменения - наберите нужную дату и нажмите клавишу ENTER.

Внимание! Можно также ввести любую дату, а не только текущую.

***3.2.2 Работа в главном меню.***

Главное меню программы содержит следующие пункты:

* создание;
* удаление;
* редактирование;
* навигация;
* отчеты;
* выход.

Перемещение по пунктам главного меню осуществляется клавишами перемещения курсора: стрелка вправо, стрелка вниз и стрелка влево, стрелка вверх. Выбор и отказ от выбора пункта меню осуществляется клавишами ENTER и ESC соответственно. В нижней части экрана имеется подсказка о тех действиях, которые будут производиться при выборе соответствующего пункта главного меню.

***3.2.3 Ввод новых данных***

Если выбран пункт главного меню "СОЗДАНИЕ", то можно ввести данные в новую карту, выбывшего из стационара (другими словами, заполнить новую форму №066/у).

Перед вами высвечивается приглашение для введения номера истории болезни вновь выбывшего из стационара больного.

Внимание! Новая карта будет создана только в том случае, если информации о больном, с введенным вами номером истории болезни, в базе данных нет.

После этого на экране появляется текст, который содержит основные пункты формы №066/у, подлежащие заполнению (см. пункт 3 "Описания применения"). Движение по этому тексту осуществляется клавишами перемещения курсора, при этом по экрану перемещается "световое пятно". Для заполнения нужного пункта КАРТЫ надо подвести "световое пятно" к данному пункту текста и нажать клавишу ENTER. Если данный пункт формы №066/у может содержать одно из наперед заданных значений, то на экран выводятся все эти значения и пользователь, используя клавиши перемещения курсора и клавишу ENTER, осуществляет выбор (используя клавишу ESC, можно выйти из данного пункта КАРТЫ без выбора какого-либо значения). Все эти, "наперед заданные" значения хранятся в справочниках. Если данный пункт может содержать любое значение, то пользователю предлагается ввести его самому.

Возврат в основное меню происходит по нажатию клавиши F10.

Внимание! Если некоторые пункты КАРТЫ (а именно, те которые нужны для составления отчетных документов) не будут заполнены, то вновь введенная форма не будет сохранена. При попытке выхода, в этом случае, пользователю будет предложено заполнить нужные пункты КАРТЫ.

***3.2.4. Редактирование ранее введенных форм.***

Выбрав пункт главного меню "Редактирование", пользователь может произвести коррекцию любой ранее введенной формы, хранящейся в БД. Для этого ему предлагается либо ввести номер истории болезни, либо фамилию интересующего его больного. Коррекция осуществляется по тем же правилам, что и при "Создании" (см. пункт 3.2.3.).

Для поиска КАРТЫ интересующего больного, пользователь может использовать "Навигатор" БД (см. пункт 3.2.5.).

Пользователь имеет возможность получить твердую копию отредактированной им КАРТЫ. Для этого ему надо, находясь в режиме "Редактирование", сделать следующее:

* включить принтер;
* вставить в принтер лист бумаги;
* нажать клавишу F5.

Перед пользователем появится меню, пунктами которого являются:

1. "Печатать текущий раздел документа", т.е. тот пункт КАРТЫ, на котором в данный момент находится курсор ("световое пятно");
2. "Печатать весь документ", т.е. всю данную КАРТУ целиком;
3. "Отказ от печати".

Клавишами перемещения курсора и клавишей ENTER пользователь осуществляет выбор.

***3.2.5. Навигация (просмотр) БД.***

Выбрав пункт "Навигатор", пользователь может просмотреть всю информацию, которая хранится в БД. Для этого ему предлагается меню второго уровня, используя пункты "Текущая карта", "Следующая карта" и "Предыдущая карта" которого, пользователь может последовательно просмотреть всю БД. При этом в нижней части экрана будет представлены номер истории болезни пациента, его фамилия и дата выписки, соответствующие очередной КАРТЕ. Кроме того, пользователь может ввести условия просмотра:

* либо номер истории болезни интересующего его больного (в этом случае, пользователь будет просматривать БД, начиная с КАРТЫ с введенным номером истории болезни (если она есть в БД); если КАРТЫ с введенным номером истории болезни в БД нет, то будет представлена первая КАРТА с номером большим, чем у заданной);
* либо фамилию больного;
* либо дату выписки больного из стационара (в этом случае, пользователь будет просматривать КАРТЫ больных, выписанных во введенную дату и во все последующие дни).

Возврат в основное меню - по F10 или по ESC.

Внимание! После выхода из "Навигатора" текущим номер истории болезни становится тот номер, который был у последней из просмотренных КАРТ. Это можно использовать для поиска и последующего редактирования интересующей пользователя КАРТЫ или для ее удаления (см. пункт 3.2.6).

***3.2.6. Удаление КАРТ.***

Выбрав этот пункт основного меню, пользователь получает возможность, путем последовательного просмотра БД, помечать определенные им КАРТЫ на удаление.

Внимание! Помеченные пользователем на удаление КАРТЫ после выполнения пункта "Удаление" физически удаляются из БД.

***3.2.7. Получение отчетных форм.***

Выбрав пункт главного меню "Отчеты", можно получить отчеты фиксированной формы на основе введенной информации. Предлагается меню со следующими пунктами:

* Месячные отчеты;
* Квартальные отчеты.

Выбрав первый пункт, можно получить месячный отчет по любому отделению стационара. Для этого пользователю предлагается указать названия отделения, по которому будет формироваться отчет. Если отделение не будет указано, то отчет формироваться не будет.

Если выбран второй пункт, то необходимо указать конкретный отчет, который надо сформировать. Выбор осуществляется с помощью следующего меню:

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Можно получить квартальные отчеты, как по всему стационару, так и по любому его стационару. Для этого пользователю предлагается определить отделение, по которому будет формироваться отчет. Если отделение не будет указано (нажата клавиша ESC), то отчет будет формироваться по всему стационару.

После выбора отчета необходимо ввести дату начала и окончания отчетного периода. Для ввода даты наберите чч/мм/гг на клавиатуре и нажмите клавишу ENTER. Для отказа от ввода нажмите клавишу ESC.

После обработки получается необходимая отчетная форма, которая изображается на экране и сохраняется в файле. Можно получить твердую копию отчетной формы на принтере.

Чтобы получить твердую копию на принтере необходимо выполнить следующие действия:

* включить принтер;
* вставить в принтер лист бумаги;
* выбрать в меню альтернативу с заголовком "Выводить отчет на печать".

Всюду для возврата в предыдущее меню используйте клавишу ESC.

Внимание! Некоторые отчетные документы имеют большую ширину, поэтому для их распечатки необходим широкий принтер. Ширину отчета можно определить при его просмотре после того, как он появится на экране.

***3.2.8. Настройка.***

Режим настройки скрыт от конечного пользователя программы и доступен лишь администратору БД, используя его можно настроить программу на структуру конкретного стационара. Подробнее процедура настройки описана в "Руководстве системного программиста".

***3.2.9. Выход из программы.***

Для окончания работы с программой "КАРТА" следует выбрать пункт "Выход" главного меню.

# Приложение 4

## Руководство системного программиста “Автоматизированной системы учета выбывших из стационара”

***1. Общие сведения о программе***

ВНИМАНИЕ!

Перед прочтением данного документа следует ознакомиться с документами “Описание применения” и “Руководство оператора”. Системному программисту необходимо также ознакомиться с документом “Текст и описание программы”.

Автоматизированная система учета выбывших из стационара (условное обозначение КАРТА) предназначена для автоматизации процесса заполнения формы №066/у Минздрава России и получения на ее основе фиксированных отчетных форм.

Программа обеспечивает:

1. Ввод, хранение, коррекцию данных по выписанным из стационара больным;
2. Оформление печатных документов по данным о выписанных больных за отчетный период;
3. Настройку на структуру конкретного стационара.

Программа "КАРТА" реализована на персональных ЭВМ IBM PC XT/AT со следующим набором периферийных устройств: принтер, дисплей с платой адаптера EGA/VGA, накопитель на жестком диске объемом не менее 80 Мб. Минимальный объем свободной оперативной памяти 540 Кб.

Программа работает под управлением операционной системы MS DOS версии 3.1 и выше.

Перед первым запуском программы в файл AUTOEXEC.BAT необходимо включить запись:

SET CLIPPER=F50.

В файле CONFIG.SYS параметр FILES установить равным 50 (FILES=50).

Программа реализована на языке программирования Clipper Summer'87.

***2.Структура базы данных***

***2.1 Общая структура***

База данных (БД) программы "КАРТА" состоит из следующих основных компонентов:

* Таблица для хранения данных по пациентам;
* Таблица для хранения шаблона формы N66/у;
* Таблица для хранения справочных данных и данных для настройки программы структуру конкретного стационара;
* Таблица для формирования и хранения отчетных документов.

***2.2 Файловая структура БД***

Каждая таблица включает в себя файл (ы) данных и файл (ы) индексов.

Таблица для хранения данных по пациентам содержит файлы:

КАRТА66.DBF и КАRТА66.NTX

DIA66.DBF и DIA66.NTX

OP66.DBF и OP66.NTX

Таблица для хранения шаблона формы №066/у содержит файлы:

MENU.DBF и MENU.NTX

Таблица для хранения справочных данных содержит файлы:

CODIF.DBF и CODIF.NTX

Таблица для формирования и хранения отчетных документов содержит файлы:

CLASS.DBF и CLASS.NTX

GRUP1.DBF и GRUP1.NTX

GRUP2.DBF и GRUP2.NTX

BUFF8.DBF и BUFF8.NTX

Кроме того, имеются файлы с информацией для представления отчетов в удобной для пользователя форме:

OTD.FRM;

OTD1.FRM;

OTD2.FRM;

OTD5.FRM, OTD51.FRM;

NEONAT.FRM;

OTCH.FRM;

OTCH1.FRM;

OTCH2.FRM;

OTCH3.FRM;

OTCH4.FRM, OTCH41.FRM;

OTCH5.FRM;

OTCH6.FRM;

OTCH7.FRM;

OTCH8.FRM;

OTCH9.FRM; OTCH91.FRM; OTCH92.FRM;

OTCH10.FRM; OTCH101.FRM; OTCH102.FRM;

OTCH11.FRM; OTCH111.FRM; OTCH112.FRM;

OTCH12.FRM; OTCH121.FRM;

OTCH13.FRM;

OTCH14.FRM;

OTCH15.FRM; OTCH151.FRM;

OTCH16.FRM;

OTCH17.FRM;

OTCH18.FRM;

OTCH19.FRM;

***2.3 Структуры файлов БД***

***2.3.1 Структура файла KARTA66.DBF.***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

NUM\_IB C 7 номер истории болезни

FAM C 30 фамилия больного

F\_S\_NAME C 30 имя, отчество больного

DATE\_B D 8 дата рождения

HOUR\_B N 2 часы рождения

MINS\_B N 2 минуты рождения

OLD N 2 код возраста

POL N 1 код пола

MASSA C 6 масса

PLACE\_LIV C 60 адрес

RAION N 2 код района

CITY\_VIL N 1 код: горожанин/селянин

DIRECT1 N 2 код направляющего учреждения

DIRECT2 N 2 код стационара

STATE N 2 код государства

WHY N 1 код причины направления

DEPARTMENT N 2 код отделения стационара

KOIKA N 2 код профиля койки

PASS N 1 код состояния поступления

TIME N 1 код поступления после болезни

DATE\_IN D 8 дата поступления в стационар

HOUR\_IN N 2 часы поступления

MINS\_IN N 2 минуты поступления

END1 N 1 код исхода

END2 N 1 код причины исхода

END3 N 2 код учреждения для перевода

DATE\_END D 8 дата исхода

HOUR\_END N 2 часы исхода

MINS\_END N 2 минуты исхода

ALL\_DAY N 2 количество дней в стационаре

DIA\_DIRECT C 4 направляющий диагноз

NUM\_COME N 1 код числа поступлений

RW\_DATE D 8 дата анализа на RW

RW\_REZ N 1 код результата RW

FAM\_DOCTOR N 2 код фамилии лечащего врача

Индексное выражение для файла: NUM\_IB.

***2.3.2 Структура файла DIA66.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

NUM\_IB C 7 номер истории болезни

KOD1 C 1 код заболевания

KOD2 C 1 код диагноза

SHIFR C 4 шифр заболевания

COMM M 10 названия заболеваний

Индексное выражение для файла: NUM\_IB+KOD1.

***2.3.3 Структура файла OP66.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

NUM\_I C 7 номер истории болезни

SHIFR C 4 шифр операции

DATA D 8 дата операции

COMM M 10 название операции

Индексное выражение для файла: NUM\_IB.

***2.3.4 Структура файла MENU.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

KEY C 9 ключ элемента 66 формы

TEXT C 60 текст элемента

Индексное выражение для файла: KEY.

***2.3.5 Структура файла CODIF.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

KEY C 9 ключ элемента справочника

TEXT C 60 текст элемента справочника

Индексное выражение для файла: KEY.

***2.3.6 Структура файла CLASS.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

NAME\_CLASS C 100 название класса заболеваний

SHIFR\_LEFT C 4 левая граница класса

SHIFR\_RIGH C 4 правая граница класса

Индексное выражение для файла: SHIFR\_LEFT.

Файлы GRUP1.DBF и GRUP2.DBF имеют аналогичную структуру.

***2.3.7 Структура файла BUFF8.DBF***

ИМЯ ТИП ДЛИНА СОДЕРЖАНИЕ

NUMBER C 2 Содержание каждого поля

SHIFR C 4 зависит от вида отчетного

NAME C 100 документа, который формируется

NAMECL C 100 и затем хранится в

SHIFRL C 4 этой базе данных.

SHIFRR C 4

COUNT1 N 10

COUNT2 N 10

A1 N 5

........

E1 N 5

Индексное выражение для файла: SHIFR.

***3. Структура программы КАРТА***

***3.1 Общие сведения о структуре программы***

Структура программы КАРТА представлена на *рис.6*.



*Рис. 6. Логическая структура программы*

Программа состоит из 5 функциональных модулей функций и библиотеки функций.

Главный модуль выполняет следующие функции:

* инициализация глобальных переменных;
* ввод текущей даты;
* открытие всех рабочих областей с индексными файлами;
* организация главного меню.

Модуль ввода данных выполняет следующие функции:

* создание и ввод данных в новую КАРТУ;
* редактирование данных уже существующих КАРТ.

Модуль навигации организует просмотр БД по определенным пользователем условиям.

Модуль удаления удаляет из БД КАРТЫ, определенные пользователем.

Модуль отчетов формирует следующие отчеты:

* месячные по любому отделению;
* квартальные как по всему стационару, так и по любому его отделению:

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Модуль настройки выполняет следующие функции:

* организация меню настройки;
* коррекция справочников по структуре стационара.

Вызов всех модулей второго уровня осуществляется из главного меню программы. Связь между отдельными модулями второго уровня осуществляется только по данным через глобальные переменные и файлы активной базы данных.

Работа программы во всех режимах кроме режима настройки детально описана в документе "Руководство программиста". Режим настройки является режимом администратора базы данных и описан в настоящем документе в разделе "Настройка программы".

***4. Настройка программы***

Настройка программы на режим конкретного использования заключается в корректировке справочников. Для корректировки справочников необходимо, находясь в главном меню программы, нажать Ctrl-F10.

При этом пользователь попадает в меню со следующими альтернативами:

* выбор и изменение списка отделений стационара;
* выбор и изменение списка профилей коек;
* выбор и изменение списка направляющих организаций;
* выбор и изменение списка причин смерти;
* выбор и изменение списка районов;
* выбор и изменение списка возрастов;
* выбор и изменение списка причин перевода;
* выбор и изменение списка причин направлений;
* выбор и изменение списка стран содружества.

При выборе любого пункта меню пользователь переходит к меню добавления, удаления и переименования выбранного списка.

В меню, для добавления нового элемента необходимо нажать клавишу INS. При этом вам будет предложено ввести новое название. Введите его и нажмите клавишу ENTER. Для удаления элемента нажмите клавишу DEL. Предоставляется другое меню из двух альтернатив "Удалять элемент из списка", "Не удалять". Для отказа от удаления выберите пункт "Не удалять" и нажмите клавишу ENTER либо клавишу ESC. Для удаления стационара выберите пункт меню "Удалять стационар из списка" и нажмите клавишу ENTER.

При нажатии клавиши ENTER осуществляется выбор текущего элемента в списке с целью его переименования. Введите новое название и нажмите клавишу ENTER. ESC означает отказ от переименования.

При нажатии ESC происходит возврат в предыдущее меню.

***5. Сообщения системному программисту***

В программе не предусмотрено специальных сообщений для системного программиста. При работе программы возможна выдача сообщений исполнительной системой Clipper. Типичными сообщениями могут быть:

* "Not enought memory","Memory fault" - по этим сообщениям следует удалить из оперативной памяти все резидентные программы, кроме русификаторов.
* "Unable to open file <имя\_файла>" - следует проверить наличие в директории указанного файла и при его отсутствии скопировать с резервной дискеты.

В более сложных случаях следует записать сообщение и ситуацию, в которой оно возникло и обратиться к разработчикам программы..

# Приложение 5

## Текст и описание программы: “Автоматизированная система учета выбывших из стационара”

.***1. Общие сведения о программе.***

**ВНИМАНИЕ!**

***Перед прочтением данного документа следует ознакомиться с документами "Описание применения" и "Руководство оператора".***

Автоматизированная система учета выбывших из стационара (условное обозначение КАРТА) предназначена для автоматизации процесса заполнения формы №066/у Минздрава и получения фиксированных отчетных форм.

Программа работает под управлением операционной системы MS DOS версии 3.1 и выше.

Перед первым запуском программы в файл *AUTOEXEC.BAT* необходимо включить запись:

*SET CLIPPER=F50.*

В файле *CONFIG.SYS* параметр *FILES* установить равным *50*

*(FILES=50).*

Программа реализована на языке программирования Clipper Summer'87.

***2. Функциональное назначение.***

Программа выполняет следующие функции:

1. Ввод, хранение, коррекцию данных по выписанным из стационара больным;
2. Оформление печатных документов по данным о выписанных больных за отчетный период;
3. Настройку на структуру конкретного стационара.

***3. Описание логической структуры программы КАРТА.***

***3.1 Общие сведения о структуре программы***

Структура программы КАРТА представлена на *рис.7.*



*Рис. 7. Логическая структура программы.*

Программа состоит из 5 функциональных модулей функций и библиотеки функций.

Главный модуль выполняет следующие функции:

* инициализация глобальных переменных;
* ввод текущей даты;
* открытие всех рабочих областей с индексными файлами;
* организация главного меню.

Модуль ввода данных выполняет следующие функции:

* создание и ввод данных в новую КАРТУ;
* редактирование данных уже существующих КАРТ.

Модуль навигации организует просмотр БД по определенным пользователем условиям.

Модуль удаления удаляет из БД КАРТЫ, определенные пользователем.

Модуль отчетов формирует следующие отчеты:

* месячные по любому отделению;
* квартальные как по всему стационару, так и по любому его отделению:

1. Состав больных в стационаре, сроки и исход лечения;
2. Состав больных новорожденных, поступивших в возрасте 0-6 суток жизни и исход их лечения;
3. Хирургическая работа учреждений;
4. Распределение больных по возрасту и району;
5. Состав больных, выбывших в возрасте от 0 до 6 суток жизни;
6. Больные, переведенные в другие лечебные учреждения;
7. Нозология больных, переведенных из других стационаров;
8. Нозологическая таблица умерших;
9. Нозология больных умерших по возрастам;
10. Нозология умерших в возрасте от 0 до 6 дней жизни;
11. Операции умерших;
12. Нозология умерших до суток;
13. Нозология инфекционных заболеваний;
14. Распределение инфекционных заболеваний по отделениям;
15. Распределение выбывших иногородних больных по каналам госпитализации и отделениям больницы;
16. Число больных, переведенных в другие стационары, из них число новорожденных, переведенных в другие стационары, и число лиц, госпитализированных для обследования и оказавшихся здоровыми;
17. Число новорожденных, умерших в возрасте от 0-6 суток, число умерших в первые 24 часа после поступления в стационар: в возрасте 0-24 часа после рождения, до 1 года, в том числе от пневмонии;
18. Число больных инфарктом миокарда, поступивших в первые сутки от начала заболевания, число больных инфарктом миокарда, умерших в первые 24 часа после поступления в стационар;
19. Число умерших беременных, рожениц и родильниц, из них число умерших от заболеваний, осложняющих беременность и роды.

Модуль настройки выполняет следующие функции:

* организация меню настройки;
* коррекция справочников по структуре стационара.

Вызов всех модулей второго уровня осуществляется из главного меню программы. Связь между отдельными модулями второго уровня осуществляется только по данным через глобальные переменные и файлы активной базы данных.

Работа программы во всех режимах кроме режима настройки детально описана в документе "Руководство оператора". Режим настройки является режимом администратора базы данных и описан в документе "Руководство системного программиста и администратора базы данных" в разделе "Настройка программы".

Исходный текст программы (всех модулей) находится в файле KARTA.PRG. Библиотека функций находится в файле LIB.OBJ. Справочники размещены в файле CODIF.DBF. Для получения загрузочного модуля необходимо набрать на клавиатуре следующую команду:

*rtlink fi KARTA, lib lib terminal, clipper, extend, dbfntx*

и нажать клавишу ENTER.

***3.2 Описание алгоритма программы***

Схема алгоритма работы программы представлена на рис. 8.

Для работы с Международным классификатором болезней (МКБ) использовалась функция MKB().

Для работы с каталогом операций использовалась функция CATALOG().



*Рис. 8. Схема алгоритма работы программы.*

Для работы со справочниками используются следующие функции:

* codif() - функция выбора альтернативы из вертикального меню, построенного на основе данных справочника.
* mempro(), codpic(), codtxt() - функции для экранного редактирования МЕМО - полей баз данных.
* extra() - функция восстановления текста выбранной альтернативы по ее номеру в справочнике.
* ins\_pic() - функция добавления информации в справочник.
* del\_pic() - функция удаления инфомации из справочника.

Для представления текста отчетного документа на экране используется функция viewer().

***4. Используемые технические средства***

Программа "КАРТА" предназначена для установки на персональных ЭВМ IBM PC XT/AT cо следующим набором периферийных устройств: принтер, дисплей с платой адаптера EGA\VGA, накопитель на жестком диске объемом не менее 80 Мб. Минимальный объем свободной оперативной памяти 540 Кб.

***5. Вызов и загрузка***

Для вызова программы следует набрать в командной строке:

*=>karta*

или выбрать файл karta.EXE с помощью «оболочки» типа 'NORTON COMMANDER' и нажать клавишу ENTER.

***6. Входные данные***

Входной информацией программы является следующая:

* данные, вводимые пользователем (см. "Руководство оператора");
* данные, хранящиеся в базе данных по пациентам (см. "Руководство системного программиста");
* текущая системная дата;
* данные, хранящиеся в справочных базах данных (см. "Руководство системного программиста").

***7. Выходные данные***

Выходной информацией программы является следующая:

* данные, введенные пользователем в базу данных по пациентам (см. "Руководство системного программиста");
* документы, сформированные по введенным данным(см. "Руководство оператора");

.***Текст программы на языке Clipper Summer'87***

**Модуль: Karta.prg**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Название программы : "KARTA" \*

\* Дата последних изменений : 23.12.92 \*

\* Исходный текст : Clipper Summer'87 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SET CONSOLE OFF

SET ESCAPE ON

SET MESSAGE TO 23 CENTER

SET BELL OF

SET DATE GERMAN

SET SCOREBOARD OFF

SET CONFIRM ON

SET WRAP ON

SET KEY -9 TO GO\_MAIN && ПО F10 - ВОЗВРАТ В МЕHЮ

SET KEY -29 TO recon

init\_lib() && Функция настройки для работы с библиотекой LIB29

t\_qwerty=.T.

CLEAR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* глобальные переменные программы

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PUBLIC edit\_index && .T.- редактировать номер ИБ нельзя

&& .F.- можно

edit\_index=.F.

PUBLIC gotomain && принудительный возврат в главную процедуру

&& .T.- прервать внутренний цикл и вернуться в MAIN

gotomain=.F.

PUBLIC \_today && Текущая дата работы

PUBLIC rec\_num && Номер текущей записи

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

f1 = CHR(218) + CHR(196) + CHR(191) + CHR(179) + ;

CHR(217) + CHR(196) + CHR(192) + CHR(179)

f2 = CHR(201) + CHR(205) + CHR(187) + CHR(186) + ;

CHR(188) + CHR(205) + CHR(200) + CHR(186)

f3 = CHR(218) + CHR(196) + CHR(191) + CHR(179) + ;

CHR(180) + CHR(196) + CHR(195) + CHR(179)

f1\_fon = CHR(218) + CHR(196) + CHR(191) + CHR(179) + ;

CHR(217) + CHR(196) + CHR(192) + CHR(179) + ;

CHR(178)

f2\_fon = CHR(201) + CHR(205) + CHR(187) + CHR(186) + ;

CHR(188) + CHR(205) + CHR(200) + CHR(186) + ;

CHR(178)

dn\_s=CHR(198)+CHR(205)+CHR(181)+CHR(179)+; && стыкуется с рамкой

CHR(217)+CHR(196)+CHR(192)+CHR(179) && по верхней границе

fon1=CHR(177)

fon2=CHR(32)

singl=CHR(218)+CHR(196)+CHR(191)+CHR(179)+;

CHR(217)+CHR(196)+CHR(192)+CHR(179)

doubl=CHR(201)+CHR(205)+CHR(187)+CHR(186)+;

CHR(188)+CHR(205)+CHR(200)+CHR(186)

IF .NOT. ISCOLOR()

color1="W+/N,N/W,W+/N,W/N,W/N" && для меню

color2="W/N,W+/N" && для gets

color3="W+/N,N/W" && для кодификаторов

color4="W/N,N/W" && для рамки каталога

color5="W/N,N/W" && для меню при редактировании

color6="W/N,W+/N" && для memed

color7=color2 && для шаблонов

color8="W/N,W+/N,N/W" && для HYPERTEXT 1-го уровня

color9="W/N,W+/N,N/W" && для HYPERTEXT 2-го уровня

ELSE

color1="W+/B,N/G,BG/N,RB+/B,BG/B"

color2="BG/B,GR+/B,BG/B,RB+/B,BG/B"

color3="N/W,W+/GR"

color4="N/GR,W+/GR"

color5="G+/B,N/W,BG/B,RB+/B,+GR/B"

color6="W+/GR,N+/W"

color7="N/GR,+GR/GR"

color8="W+/B,G+/B,N/W"

color9="B/G,W+/G,W+/N"

ENDIF

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRIVATE \_NUM\_IB && Номер истории болезни больного

PRIVATE \_FAM && Фамилия больного

\_FAM=SPACE(25)

PRIVATE \_F\_S\_NAME && Имя,Отчество больного

PRIVATE \_DATE\_B && Дата рождения больного

PRIVATE time\_B && Время рождения

time\_B="00.00"

PRIVATE \_HOUR\_B && Часы рождения

PRIVATE \_MINS\_B && Минуты рождения

PRIVATE \_POL && Пол

PRIVATE \_OLD && Возраст на момент поступления

PRIVATE \_OLD\_D && Возраст на момент смерти

PRIVATE \_MASSA && Масса

PRIVATE \_PLACE\_LIV && Место жительства

PRIVATE \_RAION && Район проживания

PRIVATE \_CITY\_VILL && Городской/сельский житель

PRIVATE \_DIRECT1 && Кем направлен

PRIVATE \_DIRECT2 && Номер направляющего стационара

PRIVATE \_STATE && Название государства

PRIVATE \_PLACE && Название области РФ

\*PRIVATE \_WHY && Причины направления

PRIVATE \_DEPARTMENT && Отделение

PRIVATE \_KOIKA && Профиль койки

PRIVATE \_PASS && Характер поступления (экстренно,не экстренно)

PRIVATE \_TIME && Через какое время после заболевания

PRIVATE \_DATE\_IN && Дата поступления

\_DATE\_IN=DATE()

PRIVATE time\_IN && Время поступления

time\_IN="00.00"

PRIVATE \_HOUR\_IN && Часы поступления

PRIVATE \_MINS\_IN && Минуты поступления

PRIVATE \_END1 && Исход заболевания

PRIVATE \_END2 && Причина исхода

PRIVATE \_END3 && Если переведен, то куда

PRIVATE \_DATE\_END && Дата выписки

PRIVATE time\_END && Время выписки

time\_END="00.00"

PRIVATE \_HOUR\_END && Часы выписки

PRIVATE \_MINS\_END && Минуты выписки

PRIVATE \_ALL\_DAY && Общее количество дней, проведенных в стационаре

PRIVATE \_DIA\_DIRECT && Диагноз направляющего учреждения

PRIVATE \_NUM\_COME && Номер поступления

PRIVATE \_RW\_DATE && Дата анализа на RW

PRIVATE \_RW\_REZ && Результат анализа

PRIVATE \_FAM\_DOCTOR && Фамилия лечащего врача

PRIVATE \_KOD1 && Клинический диагноз

PRIVATE \_KOD2 && Поталого-анатомический диагноз

PRIVATE \_SHIFR && Шифр заболевания по МКБ

PRIVATE \_SHIFR\_ILL && Шифр операции из каталога операций

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT 0 && БД шифров заболеваний всех больных

USE DIA66 INDEX DIA66 ALIAS DIA66

COPY STRUCTURE TO BUFF.DBF

SELECT 0 && Вспомогательная БД для формирования диагнозов больного

USE BUFF ALIAS BUFF

INDEX ON NUM\_IB+KOD2+KOD1 TO BUFF.NTX

SELECT 0 && БД шифров операций всех больных

USE OP66 INDEX OP66 ALIAS OP66

COPY STRUCTURE TO BUFF2.DBF

SELECT 0 && Вспомогательная БД для формирования шифров операций

USE BUFF2 ALIAS BUFF2

INDEX ON NUM\_IB TO BUFF2.NTX

SELECT 0 && БД кодификаторов

USE CODIF INDEX CODIF ALIAS CODIF

SELECT 0 && БД с основной информацией о пациентах

USE KARTA66 INDEX KARTA66 ALIAS KARTA

SELECT 0 && БД с шаблонами

USE CODPIC INDEX CODPIC ALIAS CODPIC

SELECT 0 && БД с прототипами

USE CODTXT INDEX CODTXT ALIAS CODTXT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ОСHОВHАЯ РАМКА \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SET COLOR TO "W+/N"

flop\_box('c', 0,0,24,79,doubl+fon1)

saycent(0,0,79," ФОРМА N 66 ")

saycent(24,0,79,' перемещение - выбор F10-меню ')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ВВОД СЕГОДHЯШHЕЙ ДАТЫ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SET COLOR TO(color2)

\_today=DATE()

flop\_box('c', 9,25,11,55,singl+fon2)

@ 10,32 SAY "СЕГОДHЯ:" GET \_today

READ

\_NUM\_IB=RIGHT(STR(YEAR(\_today)),2)+"00000"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ОСНОВНОЙ ЦИКЛ ПРОГРАММЫ \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@ 1,1 CLEAR TO 23,78 && очистка экрана для переменных

SET COLOR TO (color1)

@ 2,1,22,78 BOX f1\_fon

choice = 1

PRIVATE screen0

DO WHILE choice # 6

SET COLOR TO (color1)

gotomain=.f.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ВЫВОД ГЛАВНОГО МЕНЮ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@ 1,2 PROMPT "Создание" MESSAGE " ввод новой записи ИБ"

@ 1,12 PROMPT "Удаление" MESSAGE " удаление записи из ИБ"

@ 1,22 PROMPT "Редактирование/Печать" MESSAGE " редактирование записи ИБ "

@ 1,45 PROMPT "Навигатор" MESSAGE "движение по базе данных"

@ 1,56 PROMPT "Отчет" MESSAGE "составление отчетных форм"

@ 1,67 PROMPT " Выход " MESSAGE " выход из программы "

MENU TO choice

SAVE SCREEN TO screen0

DO CASE

CASE choice=1 && Добавления записи

IF( inpindex()=0) && Ввод ключа "НОМЕР ИСТОРИИ БОЛЕЗНИ"

@ 11,18 CLEAR TO 14,62

saycent(12,20,60,"ПОДОЖДИТЕ НЕМНОГО - ИДЕТ ИНИЦИАЛИЗАЦИЯ")

DO edit WITH .T.

ENDIF

CASE choice=2 && Удаление записи

DO del

CASE choice=3 && Изменение записи ИБ

SET COLOR TO(color2)

PRIVATE D1

DO WHILE .T.

D1=det() && Поиск нужной записи

IF D1=1 && Запись найдена

saycent(12,20,60,"ПОДОЖДИТЕ НЕМНОГО - ИДЕТ СЧИТЫВАНИЕ ИЗ БД")

DO edit WITH .T.

EXIT

ELSEIF D1=2 && Запись не найдена

saycent(12,20,60,"ИНФОРМАЦИИ ОБ УКАЗАННОМ БОЛЬНОМ В БД НЕТ ")

INKEY(5)

ELSE

EXIT

ENDIF

ENDDO

RELEASE D1

CASE choice=4 && Движение по БД

DO navy

CASE choice=5 && Составление отчетных документов

rez()

CASE choice=6 && Завершение программы

EXIT

ENDCASE

PRIVATE sel

sel=SELECT()

SELECT BUFF

ZAP

SELECT BUFF2

ZAP

SELECT (sel)

RELEASE sel

RESTORE SCREEN FROM screen0

ENDDO

COMMIT && Сохраняем рабочие области на диске

CLOSE ALL

DELETE FILE BUFF.DBF

DELETE FILE BUFF.DBT

DELETE FILE BUFF.NTX

DELETE FILE BUFF2.DBF

DELETE FILE BUFF2.DBT

DELETE FILE BUFF2.NTX

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* КОHЕЦ ГЛАВHОГО МОДУЛЯ \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* INPINDEX() - функция ввода номера истории болезни \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION inpindex

PRIVATE sel,ret,scr

ret=-1

@ 2,1,4,78 BOX f3+fon2

sel=SELECT()

SELECT KARTA

SET CURSOR ON

DO WHILE !gotomain

SET COLOR TO(color2)

@ 3,28 SAY "Номер ИБ " GET \_NUM\_IB PICTURE "@R 99/99999"

READ

IF LASTKEY()=27 && ESC

ret= (-1)

EXIT

ENDIF

IF LEN(ALLTRIM(\_NUM\_IB))=7

SEEK \_NUM\_IB

IF FOUND()

TONE(100,3)

message('e',"ТАКАЯ ЗАПИСЬ УЖЕ СУЩЕСТВУЕТ,ПРОВЕРЬТЕ HОМЕР ИБ ")

LOOP

ENDIF

ret=0

EXIT

ELSE

TONE(100,3)

message('e','HЕ ЗАПОЛHЕH НОМЕР ИБ,ПРОВЕРЬТЕ ЗАПИСЬ')

ret=-1

ENDIF

ENDDO

SELECT(sel)

RETURN (ret)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DET() - функция поиска необходимой для редактирования записи \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION det

PRIVATE ret1,menu1

PRIVATE sel1,clr1,screen1

ret1=2

sel1=SELECT()

clr1=SETCOLOR()

SELECT karta

SET COLOR TO &color5

@ 10,8 CLEAR TO 14,72

SAVE SCREEN TO screen1

@ 11,15 PROMPT "ВВЕДИТЕ НОМЕР И/Б "

@ 13,15 PROMPT "ВВЕДИТЕ ФАМИЛИЮ БОЛЬНОГО "

MENU TO menu1

IF menu1=0

ret1=0

ELSEIF menu1=1

SET CURSOR ON

@ 11,45 GET \_NUM\_IB PICTURE "@R 99/99999"

READ

SET CURSOR OFF

SEEK \_NUM\_IB

IF FOUND()

ret1=1

ENDIF

ELSEIF menu1=2

SET CURSOR ON

@ 13,45 GET \_FAM PICTURE "@K" VALID RUSSIAN(\_FAM)

READ

SET CURSOR OFF

SET FILTER TO FAM=ALLTRIM(\_FAM)

GO TOP

IF !EOF()

ret1=1

\_NUM\_IB=NUM\_IB

ENDIF

SET FILTER TO

ENDIF

RESTORE SCREEN FROM screen1

SELECT (sel1)

SET COLOR TO (clr1)

RETURN (ret1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ЗАПОЛНЕНИЕ 66 ФОРМЫ \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE edit

PARAMETERS do\_edit

PRIVATE wt,wb,wl,wr,choice,beg\_line,length,string,string1,title

PRIVATE sel,str,i

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ОБЪЯВЛЕНИЕ МЕНЮ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PRIVATE last,numenu

last=SELECT()

numenu=1

select 0

use menu.dbf index menu alias menu

numenu=RECCOUNT()

DECLARE promp[numenu-1],vars[numenu-1],row[numenu-1],col[numenu-1]

&& массив промптеров для основного меню

GO TOP

i=1

SEEK "MAIN"

title=STRTRAN(ALLTRIM(text),'Н','H')

SKIP

DO WHILE !EOF() &&LEFT(KEY,4)="MAIN"

promp[i]=STRTRAN(ALLTRIM(text),'Н','H')

i=i+1

SKIP

ENDDO

use

SELECT (last)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* КОНЕЦ ОБЪЯВЛЕНИЯ \*\*\*\*\*\*\*\*\*\*\*\*\*\*

AFILL(vars,' ')

AFILL(col,1)

wt=3

wb=22

wl=2

wr=77

length=wr-wl+1 && Длина строки текста, выводимого на экран

beg\_line=1

PRIVATE New\_Str && Признак новой строки для Context

New\_Str=.F. && Без выделения промптеров

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

s=IF(KARTA->END1=3,6,3)

DECLARE promp1[s],vars1[s],row1[s],col1[s] && массив промптеров дополн. меню

promp1[1]="Основное заболевание :"

promp1[2]="Осложнения :"

promp1[3]="Сопутствующие заболевания :"

AFILL(vars1,' ')

AFILL(col1,1)

IF s=6

promp1[4]="Основное заболевание :"

promp1[5]="Осложнения :"

promp1[6]="Сопутствующие заболевания :"

ENDIF

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DO initial && Процедура формирования выводимого текста

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cur\_promp=1

@ 3,1 CLEAR TO 22,78

DO WHILE .T.

IF gotomain.AND.do\_edit

IF yesno(12," Сохранить изменения в базе данных ? ")=1

IF all\_r()

DO new\_save

RETURN

ELSE

gotomain=.F.

ENDIF

ELSE

RETURN

ENDIF

ELSEIF gotomain.AND.!do\_edit

RETURN

ENDIF

new\_str=.F.

choice=hypertxt(wt,wl,wb,wr,string,promp,row,col,@beg\_line,@cur\_promp,color8,;

title)

cur\_promp=cur\_promp%len(promp)+1

IF do\_edit

i=choice

DO CASE

CASE i=0

LOOP

CASE i=1

LOOP

CASE i=2

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@\_FAM,;

"","RUSSIAN(\_FAM)")

CASE i=3

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@\_F\_S\_NAME,;

"","RUSSIAN(\_F\_S\_NAME)")

CASE i=4

\_DATE\_IN=d\_input(\_DATE\_IN)

vars[i]=DTOC(\_DATE\_IN)

\_ALL\_DAY=\_DATE\_END-\_DATE\_IN

IF \_ALL\_DAY=0

\_ALL\_DAY=1

ENDIF

DO ch\_day && Изменение количества дней, проведеннх в стационаре

CASE i=5

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@time\_IN,;

"99.99","check\_T(time\_IN)")

\_HOUR\_IN=VAL(SUBSTR(time\_IN,1,2))

\_MINS\_IN=VAL(SUBSTR(time\_IN,4,5))

CASE i=6

vars[i]=codif1("POLS",@\_POL)

CASE i=7

\_DATE\_B=d\_input(\_DATE\_B)

vars[i]=DTOC(\_DATE\_B)

CASE i=8

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@time\_B,;

"99.99","check\_T(time\_B)")

\_HOUR\_B=VAL(SUBSTR(time\_B,1,2))

\_MINS\_B=VAL(SUBSTR(time\_B,4,5))

y\_m\_day(\_DATE\_B,\_HOUR\_B,\_MINS\_B,\_DATE\_IN,\_HOUR\_IN,\_MINS\_IN)

CASE i=9

vars[i]=codif1("OLDS",@\_OLD)

CASE i=10

vars[i]=m\_input() && Ввод веса тела

CASE i=11

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@\_PLACE\_LIV)

CASE i=12

vars[i]=codif1("RIGS",@\_RAION)

CASE i=13

vars[i]=codif1("CITZ",@\_CITY\_VILL)

CASE i=14

vars[i]=codif1("DIRS",@\_DIRECT1)

IF \_DIRECT1=1

vars[i]=codif1("BIRS",@\_DIRECT2)

ELSEIF \_DIRECT1=2

vars[i]=codif1("HOSP",@\_DIRECT2)

ELSE

\_DIRECT2=0

ENDIF

CASE i=15

vars[i]=codifpic("CODIF","STTE",@\_STATE)

IF \_STATE=1

promp[i]="Регион :"

vars[i]=codifpic("CODIF","PLCE",@\_PLACE)

ELSE

promp[i]="Государство :"

ENDIF

\* CASE i=15

\* vars[i]=codif1("RIZS",@\_WHY)

CASE i=16

vars[i]=codif1("DEPS",@\_DEPARTMENT)

CASE i=17

vars[i]=codif1("KOIK",@\_KOIKA)

CASE i=18

vars[i]=codif1("EXTR",@\_PASS)

CASE i=19

vars[i]=codif1("TIMS",@\_TIME)

CASE i=20

vars[i]=codif1("REZS",@\_END1)

CASE i=21

\_DATE\_END=d\_input(\_DATE\_END)

vars[i]=DTOC(\_DATE\_END)

\_ALL\_DAY=\_DATE\_END-\_DATE\_IN

IF \_ALL\_DAY=0

\_ALL\_DAY=1

ENDIF

IF \_ALL\_DAY>=0.AND.EMPTY(\_DATE\_IN)=.F.

vars[i]=vars[i]+SPACE(5)+"Проведено дней в стационаре :"+STR(\_ALL\_DAY)

ENDIF

CASE i=22

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@time\_END,;

"99.99","check\_T(time\_END)")

\_HOUR\_END=VAL(SUBSTR(time\_END,1,2))

\_MINS\_END=VAL(SUBSTR(time\_END,4,5))

CASE i=23

PRIVATE txtd

txtd=SPACE(100)

vars[i]=offset\_get(wt,wl,beg\_line,row[i],col[i],promp[i],@\_DIA\_DIRECT,;

"@R 999.9")

mkb(1,1,@\_DIA\_DIRECT,@txtd)

IF \_DIA\_DIRECT=" "

vars[23]=""

ELSE

vars[23]=SUBSTR(\_DIA\_DIRECT,1,3)+"."+SUBSTR(\_DIA\_DIRECT,4,1)+" "+;

"<"+TRIM(txtd)+">"

new\_str=.T.

ENDIF

RELEASE txtd

CASE i=24

vars[i]=codif1("VIZI",@\_NUM\_COME)

CASE i=27

\_RW\_DATE=d\_input(\_RW\_DATE)

vars[i]=DTOC(\_RW\_DATE)

CASE i=28

vars[i]=codif1("RWRZ",@\_RW\_REZ)

CASE i=29

vars[i]=codifpic("CODIF","FAMS",@\_FAM\_DOCTOR)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CASE i=25

vars[i]=diagn()

new\_str=.T.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CASE i=26

DO op

new\_str=.T.

ENDCASE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string1=""

IF choice#25.AND.choice#26

vars[choice]=TRIM(vars[choice])+"."

ENDIF

context(@string1,promp[choice],vars[choice],length,New\_Str)

IF choice=20

IF \_END1=2 && переведен

context(@string1,"Причина:",codif1("RIZ2",@\_END2)+".",length,.F.)

context(@string1,"Куда:",codif1("HOSP",@\_END3)+".",length,.F.)

ELSEIF \_END1=3 && умер

context(@string1,"Причина:",codif1("RIZ3",@\_END2)+".",length,.F.)

ENDIF

ELSEIF choice=22.AND.\_END1=3

y\_m\_day(\_DATE\_B,\_HOUR\_B,\_MINS\_B,\_DATE\_END,\_HOUR\_END,\_MINS\_END)

context(@string1,"Возраст на момент смерти :",;

extra1(\_OLD\_D,"OLDS")+".",length,.F.)

ELSEIF choice=26

context(@string1,"Обследование на реакцию ВАССЕРМАНА :","",length,.F.)

ENDIF

stuff1(@string,length,string1,choice,row,len(promp))

ENDIF

ENDDO

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ПРОЦЕДУРА ФОРМИРОВАНИЯ СОДЕРЖИМОГО 66 ФОРМЫ \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE initial

PRIVATE sel,i,v

PRIVATE rez

SET CURSOR OFF

sel=SELECT()

v=replicate(chr(176),30)

@ 13,25 SAY v

SELECT karta

vars[1]= SUBSTR(\_NUM\_IB,1,2)+'/'+SUBSTR(\_NUM\_IB,3,7)

vars[2] =FAM

\_FAM=FAM

vars[3] =F\_S\_NAME

\_F\_S\_NAME=F\_S\_NAME

vars[4]=DTOC(DATE\_IN)

\_DATE\_IN=DATE\_IN

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_HOUR\_IN=HOUR\_IN

\_MINS\_IN=MINS\_IN

IF \_HOUR\_IN=0.AND.\_MINS\_IN=0

time\_IN="00.00"

ELSEIF \_HOUR\_IN=0

time\_IN="00."+STR(MINS\_IN)

ELSEIF \_MINS\_IN=0

time\_IN=STR(HOUR\_IN)+".00"

ELSE

time\_IN=STR(HOUR\_IN)+"."+STR(MINS\_IN)

ENDIF

vars[5]=time\_IN

\*----------------------------------

vars[6] =extra1(POL,"POLS")

\_POL=POL

vars[7] =DTOC(DATE\_B)

\_DATE\_B=DATE\_B

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_HOUR\_B=HOUR\_B

\_MINS\_B=MINS\_B

IF \_HOUR\_B=0.AND.\_MINS\_B=0

time\_B="00.00"

ELSEIF \_HOUR\_B=0

time\_B="00."+STR(MINS\_B)

ELSEIF \_MINS\_B=0

time\_B=STR(HOUR\_B)+".00"

ELSE

time\_B=STR(HOUR\_B)+"."+STR(MINS\_B)

ENDIF

vars[8]=time\_B

\*-----------------------------------

vars[9] =extra1(OLD,"OLDS")

\_OLD=OLD

\_OLD\_D=OLD\_D

vars[10] =MASSA

\_MASSA =MASSA

vars[11] =PLACE\_LIV

\_PLACE\_LIV=PLACE\_LIV

vars[12] =extra1(RAION,"RIGS")

\_RAION =RAION

vars[13]=extra1(CITY\_VILL,"CITZ")

\_CITY\_VILL=CITY\_VILL

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_DIRECT1=DIRECT1

\_DIRECT2=DIRECT2

vars[14]=IF(\_DIRECT2=0,extra1(\_DIRECT1,"DIRS"),;

IF(\_DIRECT1=1,extra1(\_DIRECT2,"BIRS"),;

extra1(\_DIRECT2,"HOSP")))

\*------------------------------------

promp[15]=IF(PLACE#0,"Регион :","Государство :")

vars[15]=IF(STATE#0,IF(STATE=1,;

IF(PLACE=0,"Российская Федерация",extra1(PLACE,"PLCE")),;

extra1(STATE,"STTE")),;

"Российская Федерация")

\_STATE=IF(STATE=0,1,STATE)

\_PLACE=PLACE

vars[16]=extra1(DEPARTMENT,"DEPS")

\_DEPARTMENT=DEPARTMENT

vars[17]=extra1(KOIKA,"KOIK")

\_KOIKA=KOIKA

vars[18]=extra1(PASS,"EXTR")

\_PASS=PASS

vars[19]=extra1(TIME,"TIMS")

\_TIME=TIME

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_END1=END1

\_END2=END2

\_END3=END3

vars[20]=extra1(\_END1,"REZS")

\*----------------------------------

vars[21]=DTOC(DATE\_END)

\_DATE\_END=DATE\_END

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_HOUR\_END=HOUR\_END

\_MINS\_END=MINS\_END

IF \_HOUR\_END=0.AND.\_MINS\_END=0

time\_END="00.00"

ELSEIF \_HOUR\_END=0

time\_IN="00."+STR(MINS\_END)

ELSEIF \_MINS\_END=0

time\_IN=STR(HOUR\_END)+".00"

ELSE

time\_END=STR(HOUR\_END)+"."+STR(MINS\_END)

ENDIF

vars[22]=time\_END

\*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_ALL\_DAY=ALL\_DAY

IF !EMPTY(\_DATE\_END)

vars[21]=vars[21]+SPACE(5)+"Проведено дней в стационаре :"+STR(\_ALL\_DAY)

ENDIF

\*----------------------------------

\_DIA\_DIRECT=SHIFR

IF \_DIA\_DIRECT#" "

PRIVATE txtd

txtd=SPACE(100)

mkb(1,1,@\_DIA\_DIRECT,@txtd)

vars[23]=SUBSTR(\_DIA\_DIRECT,1,3)+"."+SUBSTR(\_DIA\_DIRECT,4,1)+" "+;

"<"+TRIM(txtd)+">"

RELEASE txtd

ELSEIF \_DIA\_DIRECT=" "

vars[23]=\_DIA\_DIRECT

ENDIF

\*----------------------------------

vars[24]=extra1(NUM\_COME,"VIZI")

\_NUM\_COME=NUM\_COME

vars[27]=DTOC(RW\_DATE)

\_RW\_DATE=RW\_DATE

vars[28]=extra1(RW\_REZ,"RWRZ")

\_RW\_REZ=RW\_REZ

vars[29]=extra1(FAM\_DOCTOR,"FAMS")

\_FAM\_DOCTOR=FAM\_DOCTOR

v=replicate(chr(178),10)

@ 13,25 SAY v

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

vars[25]=initial1("DIA66")

v=replicate(chr(178),20)

@ 13,25 SAY v

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SELECT op66

SET SOFTSEEK ON

seek \_num\_ib

SET SOFTSEEK OFF

IF !FOUND()

vars[26]="" && Хирургические операции

\_SHIFR\_ILL="0000" &&SHIFR\_ILL

ELSE

PRIVATE txts,string8

txts=SPACE(70)

STORE "" TO string8

DO WHILE NUM\_IB=\_NUM\_IB

\_SHIFR\_ILL=SHIFR

catalog(@\_SHIFR\_ILL,@txts)

txts=TRIM(txts)

context(@string8,"",txts,length,.F.)

context(@string8," Дата проведения : ",DTOC(DATA)+".",length,.F.)

context(@string8," Название операции : ",ALLTRIM(COMM),length,.F.)

vars[26]=string8

SKIP 1

ENDDO

RELEASE txts,string8

SELECT BUFF2

COMMIT

APPEND FROM OP66 FOR NUM\_IB=\_NUM\_IB

ENDIF

v=replicate(chr(178),30)

@ 13,25 SAY v

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ФОРМИРОВАНИЕ ТЕКСТА \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string="" && Начальный текст

SELECT karta

SEEK \_NUM\_IB

rez=FOUND()

New\_Str=.F.

FOR i=1 TO LEN(promp)

IF (i=23.AND.\_DIA\_DIRECT#" ").OR.i=25.OR.i=26

New\_Str=.T.

ENDIF

IF rez.AND.!EMPTY(vars[i])

row[i]=context(@string,promp[i],TRIM(vars[i])+".",length,New\_Str)

ELSE

row[i]=context(@string,promp[i],vars[i],length,New\_Str)

ENDIF

New\_Str=.F.

IF i=20 && Промпт "ИСХОД"

IF \_END1=2 && переведен

context(@string,"Причина:",extra1(\_END2,"RIZ2")+".",length,.F.)

context(@string,"Куда:",extra1(\_END3,"HOSP")+".",length,.F.)

ELSEIF \_END1=3 && умер

context(@string,"Причина:",extra1(\_END2,"RIZ3")+".",length,.F.)

ENDIF

ELSEIF i=22.AND.\_END1=3

context(@string,"Возраст на момент смерти :",;

extra1(\_OLD\_D,"OLDS")+".",length,.F.)

ELSEIF i=26

context(@string,"Обследование на реакцию ВАССЕРМАНА :","",length,.F.)

ENDIF

NEXT

SET CURSOR ON

SELECT (sel)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция инициализации диагнозов \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION initial1

PARAMETERS DBN

PRIVATE sl,rez1

SET CURSOR OFF

sl=SELECT()

SELECT &DBN

SET SOFTSEEK ON

SEEK \_NUM\_IB

SET SOFTSEEK OFF

rez1=FOUND()

IF !rez1

vars1[1]="" && Основной диагноз

vars1[2]="" && Осложнения

vars1[3]="" && Сопутствующие заболевания

IF \_END1=3

vars1[4]="" && Основной диагноз

vars1[5]="" && Осложнения

vars1[6]="" && Сопутствующие заболевания

ENDIF

\_SHIFR=SPACE(4) && SHIFR

\_KOD1=0 && KOD1

\_KOD2=0 && KOD2

ELSE

PRIVATE txts,string2,string3,string4,string5,string6,string7

txts=SPACE(100)

STORE "" TO string2,string3,string4,string5,string6,string7

DO WHILE NUM\_IB=\_NUM\_IB

\_KOD1=KOD1

\_KOD2=KOD2

\_SHIFR=SHIFR

IF \_SHIFR="0000"

txts="Здоров"

ELSE

IF \_KOD1="1".OR.\_KOD1="2".AND.\_KOD2#"2"

mkb(1,1,@\_SHIFR,@txts)

ENDIF

ENDIF

txts=SUBSTR(\_SHIFR,1,3)+"."+SUBSTR(\_SHIFR,4,1)+" "+"<"+TRIM(txts)+">"

IF \_KOD2#"2"

IF \_KOD1="1"

context(@string2,"",txts,length,.F.)

context(@string2,"",ALLTRIM(COMM1),length,.F.)

vars1[1]=string2

ELSEIF \_KOD1="2"

context(@string3,"",txts,length,.F.)

vars1[2]=string3

ELSEIF \_KOD1="3"

context(@string4,"",ALLTRIM(COMM1),length,.F.)

vars1[3]=string4

ENDIF

ELSEIF \_KOD2="2".AND.\_END1=3

IF \_KOD1="1"

context(@string5,"",txts,length,.F.)

context(@string5,"",ALLTRIM(COMM1),length,.F.)

vars1[4]=string5

ELSEIF \_KOD1="2"

context(@string6,"",ALLTRIM(COMM1),length,.F.)

vars1[5]=string6

ELSEIF \_KOD1="3"

context(@string7,"",ALLTRIM(COMM1),length,.F.)

vars1[6]=string7

ENDIF

ENDIF

SKIP 1

ENDDO

RELEASE txts,string2,string3,string4,string5,string6,string7

SELECT BUFF

APPEND FROM DIA66 FOR NUM\_IB=\_NUM\_IB

ENDIF

PRIVATE string11,j

string11=""

New\_Str=.T.

context(@string11,SPACE(10)+"Клинический диагноз"," ",length,.T.)

FOR j=1 TO s

IF rez1.AND.!EMPTY(vars1[j])

row1[j]=context(@string11,promp1[j],TRIM(vars1[j])+".",length,New\_Str)

ELSE

row1[j]=context(@string11,promp1[j],vars1[j],length,New\_Str)

ENDIF

IF j=3.AND.\_END1=3

context(@string11," "," ",length,.T.)

context(@string11,SPACE(10)+"Паталого-анатомический диагноз"," ",length,.T.)

ENDIF

NEXT

SET CURSOR ON

SELECT (sl)

RETURN (string11)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция ввода даты \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION d\_input

PARAMETERS dat

PRIVATE screen

SAVE SCREEN TO screen

SET CURSOR ON

@ 10,25 CLEAR TO 15,55

@ 10,25 TO 15,55

saycent(10,30,50,"ВВЕДИТЕ В ФОРМАТЕ")

@ 12,36 SAY "дд.мм.гг"

@ 14,36 GET dat PICTURE "@D"

READ

SET CURSOR OFF

RESTORE SCREEN FROM screen

RETURN dat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция ввода массы пациента \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION m\_input

PRIVATE screen

SAVE SCREEN TO screen

SET CURSOR ON

@ 10,25 CLEAR TO 15,55

@ 10,25 TO 15,55

saycent(10,30,50,"ВВЕДИТЕ В ФОРМАТЕ")

@ 12,38 SAY "кг/гр."

@ 14,38 GET \_MASSA PICTURE "@P 99/999"

READ

SET CURSOR OFF

RESTORE SCREEN FROM screen

RETURN \_MASSA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция проверки времени \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION check\_T

PARAMETERS timeS

PRIVATE L,hour,mins

L=.F.

hour=SUBSTR(timeS,1,2)

mins=SUBSTR(timeS,4,5)

IF VAL(hour)<24.AND.VAL(mins)<60

L=.T.

ENDIF

RETURN (L)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Определение количества дней, проведеннх в стационаре \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE ch\_day

PRIVATE string2

string2=""

vars[choice]=vars[choice]+"."

context(@string2,promp[choice],vars[choice],length,New\_Str)

stuff1(@string,length,string2,choice,row,len(promp))

choice=21

vars[choice]=DTOC(\_DATE\_END)

IF \_ALL\_DAY>=0.AND.EMPTY(\_DATE\_IN)=.F.

vars[choice]=DTOC(\_DATE\_END)+SPACE(5)+"Проведено дней в стационаре :"+;

STR(\_ALL\_DAY)

ENDIF

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура работы с диагнозами \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION diagn

PRIVATE txtf,sel,w\_do

PRIVATE F1,screen,color

PRIVATE str

PRIVATE s

PRIVATE q

PRIVATE string11

q=0

str=""

txtf=SPACE(100)

\_SHIFR=SPACE(4)

sel=SELECT()

F1=0

string11=vars[25]

s=IF(\_END1=3,6,3)

IF LEN(promp1)#s

@ 11,18 CLEAR TO 13,62

@ 11,18 TO 13,62

saycent(12,20,60,"ФОРМИРУЕТСЯ МЕНЮ ДИАГНОЗОВ")

DECLARE promp1[s],vars1[s],row1[s],col1[s] && массив промптеров дополн. меню

promp1[1]="Основное заболевание :"

promp1[2]="Осложнения :"

promp1[3]="Сопутствующие заболевания :"

IF s=6

promp1[4]="Основное заболевание :"

promp1[5]="Осложнения :"

promp1[6]="Сопутствующие заболевания :"

ENDIF

AFILL(vars1,' ')

AFILL(col1,1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

string11=initial1("BUFF") && Функция формирования выводимого текста

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ENDIF

wt1=3

wb1=IF(s=3,12,20)

wl1=2

wr1=77

length=wr1-wl1+1 && Длина строки текста, выводимого на экран

beg\_line1=1

PRIVATE New\_Str1 && Признак новой строки для Context

New\_Str1=.F. && Без выделения промптеров

cur\_promp1=1

DO WHILE !gotomain

q=hypertxt(wt1,wl1,wb1,wr1,string11,promp1,row1,col1,;

@beg\_line1,@cur\_promp1,color9," ДИАГНОЗ ПАЦИЕНТА ")

cur\_promp1=cur\_promp1%len(promp1)+1

DO CASE

CASE q=0

LOOP

CASE q=1.OR.q=2.OR.q=4

w\_do=1

SAVE SCREEN TO screen

@ 11,25 CLEAR TO 16,55

@ 11,25 TO 16,55 DOUBLE

@ 11,30 PROMPT "ДОБАВИТЬ"

@ 11,44 PROMPT "УДАЛИТЬ"

IF EMPTY(vars1[q]).OR.BUFF->KOD1="2".AND.BUFF->KOD2="2"

vars1[q]=""

KEYBOARD CHR(13)

ENDIF

MENU TO w\_do

str=vars1[q]

IF w\_do=1

@ 13,30 SAY "ВВЕДИТЕ КОД" GET \_SHIFR PICTURE "@R 999.9"

READ

IF LASTKEY()=27

vars1[q]=str

RESTORE SCREEN FROM screen

LOOP

ENDIF

F1=mkb(1,1,@\_SHIFR,@txtf)

IF F1#-1

txtf=SUBSTR(\_SHIFR,1,3)+"."+SUBSTR(\_SHIFR,4,1)+" "+;

"<"+TRIM(txtf)+">"+"."

SELECT BUFF

APPEND BLANK

REPLACE NUM\_IB WITH \_NUM\_IB

REPLACE SHIFR WITH \_SHIFR

REPLACE KOD2 WITH IF(q=4,"2","1")

REPLACE KOD1 WITH IF(q=1.OR.q=4,"1","2")

REPLACE COMM1 WITH MEMPRO(COMM1,10,5,18,75,;

" ВВЕДИТЕ НЕОБХОДИМЫЕ ЗАМЕЧАНИЯ","ILLS",'ILLS')

context(@str,"",txtf+".",length,.F.)

context(@str,"Замечания :",ALLTRIM(COMM1),length,.T.)

ENDIF

ELSEIF w\_do=2

PRIVATE i,j,k,EN,ET,NALL,MALL,NDEL

NALL=INT(LEN(str)/length)

MALL=NALL

FOR i=1 TO NALL

ET=ALLTRIM(SUBSTR(str,length\*(i-1)+1,length))

EN=ASC(ET)

IF EN>57

MALL=MALL-1

ENDIF

NEXT

DECLARE \_0B[MALL],\_0S[MALL]

k=1

FOR j=1 TO NALL

ET=ALLTRIM(SUBSTR(str,length\*(j-1)+1,length))

EN=ASC(ET)

IF EN<58

\_0B[k]=SUBSTR(str,length\*(j-1)+1,length)

\_0S[k]=LEFT(ALLTRIM(\_0B[k]),5)

k=k+1

ELSE

\_0B[k-1]=\_0B[k-1]+SUBSTR(str,length\*(j-1)+1,length)

ENDIF

NEXT

NDEL=ACHOICE(13,35,15,45,\_0S)

SELECT BUFF

IF q=1.OR.q=4

SEEK \_NUM\_IB+IF(q=1,"1","2")+"1"

ELSEIF q=2

SEEK \_NUM\_IB+"1"+"2"

ENDIF

SKIP NDEL-1

DELETE

PACK

str=""

FOR j=1 TO MALL

IF j#NDEL

str=str+\_0B[j]

ENDIF

NEXT

RELEASE j,NALL,NDEL

RELEASE \_0B,\_0S

ENDIF

vars1[q]=str

RESTORE SCREEN FROM screen

CASE q=3.OR.q=5.OR.q=6

PRIVATE str356

STORE "" TO str356

SELECT BUFF

private s

s=\_NUM\_IB+IF(q=3,"1","2")+IF(q=5,"2","3")

SEEK s && \_NUM\_IB+IF(q=3,"1","2")+IF(q=5,"2","3")

IF !FOUND()

APPEND BLANK

REPLACE NUM\_IB WITH \_NUM\_IB

REPLACE KOD1 WITH IF(q=5,"2","3")

REPLACE KOD2 WITH IF(q=3,"1","2")

ENDIF

SET CURSOR ON

REPLACE COMM1 WITH ;

MEMPRO(COMM1,10,5,15,75,;

IF(q=5," ВВЕДИТЕ НАЗВАНИЯ ОСЛОЖНЕНИЙ ",;

" ВВЕДИТЕ НАЗВАНИЯ СОПУТСТВУЮЩИХ ЗАБОЛЕВАНИЙ "),;

"ILLS",'ILLS')

context(@str356,"",ALLTRIM(COMM1),length,.F.)

vars1[q]=str356

RELEASE str356

ENDCASE

new\_str1=.T.

string111=""

context(@string111,promp1[q],vars1[q],length,New\_Str1)

IF q=3.AND.\_END1=3

context(@string111," "," ",length,.T.)

context(@string111,SPACE(10)+"Паталого-анатомический диагноз"," ",length,.T.)

ENDIF

stuff1(@string11,length,string111,q,row1,len(promp1))

ENDDO

REINDEX

gotomain=.F.

SELECT (sel)

RETURN (string11)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура работы с операциями \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE op

PRIVATE txto,sel,w\_do

PRIVATE F2,screen,color

PRIVATE stro

STORE "" TO stro

txto=SPACE(80)

\_SHIFR\_ILL="0000"

sel=SELECT()

SAVE SCREEN TO screen

@ 11,25 CLEAR TO 16,55

@ 11,25 TO 16,55 DOUBLE

@ 11,30 PROMPT "ДОБАВИТЬ"

@ 11,44 PROMPT "УДАЛИТЬ"

IF EMPTY(vars[choice])

KEYBOARD CHR(13)

ENDIF

MENU TO w\_do

stro=vars[choice]

IF w\_do=1

@ 13,30 SAY "ВВЕДИТЕ КОД" GET \_SHIFR\_ILL PICTURE "@R 99.99"

READ

RESTORE SCREEN FROM screen

IF LASTKEY()=27

RETURN

ENDIF

F2=catalog(@\_SHIFR\_ILL,@txto)

IF F2#-1

SELECT BUFF2

APPEND BLANK

REPLACE NUM\_IB WITH \_NUM\_IB

REPLACE SHIFR WITH \_SHIFR\_ILL

REPLACE DATA WITH d\_input(DATA)

SET CURSOR ON

REPLACE COMM WITH ;

MEMPRO(COMM,10,5,15,75," ВВЕДИТЕ НАЗВАНИЕ ОПЕРАЦИИ ","OPER",'OPER')

context(@stro,"",ALLTRIM(txto)+".",length,.F.)

context(@stro," Дата проведения : ",DTOC(DATA)+".",length,.F.)

context(@stro," Название операции : ",ALLTRIM(COMM)+".",length,.F.)

ENDIF

ELSEIF w\_do=2

PRIVATE i,j,k,EN,ET,NALL,MALL,NDEL

NALL=INT(LEN(stro)/length)

MALL=NALL

FOR i=1 TO NALL

ET=ALLTRIM(SUBSTR(stro,length\*(i-1)+1,length))

EN=ASC(ET)

IF EN<>60

MALL=MALL-1

ENDIF

NEXT

DECLARE \_0B[MALL],\_0S[MALL]

k=1

FOR j=1 TO NALL

ET=ALLTRIM(SUBSTR(stro,length\*(j-1)+1,length))

EN=ASC(ET)

IF EN=60

\_0B[k]=SUBSTR(stro,length\*(j-1)+1,length)

\_0S[k]=LEFT(ALLTRIM(\_0B[k]),5)

k=k+1

ELSE

\_0B[k-1]=\_0B[k-1]+SUBSTR(stro,length\*(j-1)+1,length)

ENDIF

NEXT

NDEL=ACHOICE(13,35,15,45,\_0S)

IF LASTKEY()=27

RETURN

ENDIF

SELECT BUFF2

GO NDEL

DELETE

PACK

stro=""

FOR j=1 TO MALL

IF j#NDEL

stro=stro+\_0B[j]

ENDIF

NEXT

RELEASE j,NALL,NDEL

RELEASE \_0B,\_0S

ENDIF

vars[choice]=stro

SELECT (sel)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ПРОЦЕДУРА ЗАПОЛНЕНИЯ БД karta.dbf \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE new\_save

PRIVATE sel,v

sel=SELECT()

SET CURSOR OFF

SELECT karta

@ 11,18 CLEAR TO 13,62

@ 10,17 TO 14,63

saycent(12,20,60,"ПОДОЖДИТЕ НЕМНОГО - ИДЕТ ЗАПИСЬ В БД")

SET COLOR TO W/N

v=replicate(chr(32),30)

SET COLOR TO

@ 13,25 SAY v

SEEK \_NUM\_IB

IF FOUND()=.F.

APPEND BLANK

REPLACE NUM\_IB WITH \_NUM\_IB

rec\_num = RECNO()

ENDIF

REPLACE FAM WITH ALLTRIM(\_FAM)

REPLACE F\_S\_NAME WITH ALLTRIM(\_F\_S\_NAME)

REPLACE DATE\_B WITH \_DATE\_B

REPLACE HOUR\_B WITH \_HOUR\_B

REPLACE MINS\_B WITH \_MINS\_B

REPLACE POL WITH \_POL

REPLACE OLD WITH \_OLD

REPLACE OLD\_D WITH \_OLD\_D

REPLACE MASSA WITH \_MASSA

REPLACE PLACE\_LIV WITH \_PLACE\_LIV

REPLACE RAION WITH \_RAION

REPLACE CITY\_VILL WITH \_CITY\_VILL

REPLACE DIRECT1 WITH \_DIRECT1

REPLACE DIRECT2 WITH \_DIRECT2

REPLACE STATE WITH \_STATE

REPLACE PLACE WITH \_PLACE

\*REPLACE WHY WITH \_WHY

REPLACE DEPARTMENT WITH \_DEPARTMENT

REPLACE KOIKA WITH \_KOIKA

REPLACE PASS WITH \_PASS

REPLACE TIME WITH \_TIME

REPLACE DATE\_IN WITH \_DATE\_IN

REPLACE HOUR\_IN WITH \_HOUR\_IN

REPLACE MINS\_IN WITH \_MINS\_IN

REPLACE END1 WITH \_END1

REPLACE END2 WITH \_END2

REPLACE END3 WITH \_END3

REPLACE DATE\_END WITH \_DATE\_END

REPLACE HOUR\_END WITH \_HOUR\_END

REPLACE MINS\_END WITH \_MINS\_END

REPLACE ALL\_DAY WITH \_ALL\_DAY

REPLACE SHIFR WITH \_DIA\_DIRECT

REPLACE NUM\_COME WITH \_NUM\_COME

REPLACE RW\_DATE WITH \_RW\_DATE

REPLACE RW\_REZ WITH \_RW\_REZ

REPLACE FAM\_DOCTOR WITH \_FAM\_DOCTOR

\*REINDEX

COMMIT

v=replicate(chr(177),10)

@ 13,25 SAY v

SELECT DIA66

DELETE FOR NUM\_IB=\_NUM\_IB

PACK

\*COMMIT

IF \_END1=3

APPEND FROM BUFF FOR NUM\_IB=\_NUM\_IB

ELSE

APPEND FROM BUFF FOR NUM\_IB=\_NUM\_IB.AND.KOD2#"2"

ENDIF

\*REINDEX

COMMIT

SELECT BUFF

ZAP

\*COMMIT

\*REINDEX

COMMIT

v=replicate(chr(177),20)

@ 13,25 SAY v

SELECT OP66

DELETE FOR NUM\_IB=\_NUM\_IB

PACK

\*COMMIT

APPEND FROM BUFF2 FOR NUM\_IB=\_NUM\_IB

v=replicate(chr(177),30)

\*REINDEX

COMMIT

@ 13,25 SAY v

SELECT BUFF2

ZAP

\*COMMIT

\*REINDEX

COMMIT

SELECT (sel)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура удаления записей \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE del

PRIVATE flag\_del && число записей,помеченных для удаления

PRIVATE nr,tr,del\_str,temp,\_01,\_02,sel

@ 5,1,22,78 BOX dn\_s+fon1

sel=SELECT()

flag\_del=0

c\_d=2

SELECT KARTA

\*RECALL ALL

\*GO TOP

nr=RECCOUNT()

DECLARE stor\_ib[nr]

DO WHILE !gotomain

DO first

@ 7,5,16,74 BOX singl+fon2

SET COLOR TO "r+\*/b"

saycent(5,0,79,if(DELETED(),"Запись помечена на удаление",SPACE(27)))

SET COLOR TO (color1)

@ 10,10 PROMPT IF(!BOF(),"Вернуться к предыдущей записи","\*\*\*\*\*\*")

@ 12,10 PROMPT IF(DELETED(),"Отменить удаление текущей записи",;

"Пометить текущую запись на удаление")

@ 14,10 PROMPT IF(!EOF(),"Перейти к следующей записи","\*\*\*\*\*\*")

@ 16,35 PROMPT "Выполнить" MESSAGE "Удалить помеченные записи и "+;

"вернуться в главное меню"

MENU TO c\_d

DO CASE

CASE c\_d=0

LOOP

CASE c\_d=1

IF(!BOF())

SKIP -1

ENDIF

CASE c\_d=2

IF(!EOF())

IF !DELETED()

DELETE

flag\_del=flag\_del+1

stor\_ib[flag\_del]=NUM\_IB

ELSE

RECALL

tr=ASCAN(stor\_ib,NUM\_IB)

ADEL(stor\_ib,tr)

flag\_del=flag\_del-1

ENDIF

ENDIF

CASE c\_d=3

IF(!EOF())

SKIP

ENDIF

CASE c\_d=4

EXIT

ENDCASE

ENDDO

IF flag\_del>0

y=yesno(10,"Удалить помеченные "+alltrim(str(flag\_del))+" записей ?")

IF y=1

temp="NUM\_IB='"

del\_str=temp+stor\_ib[1]+"'"

temp=".OR."+temp

FOR tr=2 TO flag\_del

del\_str=del\_str+temp+stor\_ib[tr]+"'"

NEXT

DELETER(del\_str,"DIA66") && Удаление из DIA66.DBF

DELETER(del\_str,"OP66") && Удаление из OP66.DBF

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pack && Удаление из KARTA66.DBF

ELSE

RECALL ALL

GOTO TOP

ENDIF

ENDIF

SELECT (sel)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура формирования отчетных документов \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION rez

PRIVATE \_OTCH,\_OTCH\_N,scr1

\_OTCH=00

\_OTCH\_N=""

SAVE SCREEN TO scr1

PRIVATE sel

sel=SELECT()

PRIVATE \_DATE\_FROM

\_DATE\_FROM=\_today

PRIVATE \_DATE\_TILL

\_DATE\_TILL=\_today

PRIVATE dep,dep\_name

PRIVATE numb1

PRIVATE txt

PRIVATE pole

PRIVATE count

count=1

PRIVATE \_c

\_c=1

PRIVATE \_p

\_p=1

PRIVATE OT1,OT2

PRIVATE coun,c1,v1,v2

PRIVATE f

f=1

DO WHILE .T.

SELECT 0

USE BUFF8.DBF INDEX BUFF8 ALIAS BUFF8

ZAP

numb1=0

txt=SPACE(100)

pole=1

STORE "" TO OT1,OT2

dep=0

dep\_name=""

codif1("PERD",@\_p)

IF \_p=0

SELECT BUFF8

USE

EXIT

ELSEIF \_p=2

\_OTCH\_N=codif1("OTCH",@\_OTCH)

IF \_OTCH=0

SELECT BUFF8

USE

EXIT

ENDIF

ENDIF

dep\_name=codif1("DEPS",@dep)

IF \_p=1.AND.dep=0

SELECT BUFF8

USE

LOOP

ENDIF

dep\_name=IF(dep=0,"Весь стационар",dep\_name)

IF period()=0 && Ввод пользователем периода отчета

SET CURSOR OFF

IF \_p=1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* МЕСЯЧНЫЕ ОТЧЕТЫ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_OTCH\_N="Месячный отчет"

SELECT DIA66

SET RELATION TO SHIFR INTO BUFF8

SELECT karta

SET RELATION TO NUM\_IB INTO DIA66

GO TOP

PRIVATE OT1D1,OT2D1,OT1D2,OT2D2

IF dep=2.OR.dep=11

OT1="OTD5.FRM"

OT1D1="OTD2.FRM"

OT2D1="OTD51.TXT"

ELSE

OT1="OTD.FRM"

OT1D1="OTD1.FRM"

OT2D1="OTD\_1.TXT"

OT1D2="OTD2.FRM"

OT2D2="OTD\_2.TXT"

ENDIF

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF dep=KARTA->DEPARTMENT.AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

KARTA->END1#3.AND.DIA66->KOD1="1"

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

ENDIF

REPLACE COUNT1 WITH COUNT1+KARTA->ALL\_DAY && ПРОВЕДЕНО ДНЕЙ

REPLACE COUNT2 WITH COUNT2+1 && ВСЕГО БОЛЬНЫХ

pole=FIELD(8+KARTA->RAION)

REPLACE &pole WITH &pole+1 && из Москвы/Моск.обл./Иногородн./Село

pole=FIELD(14+KARTA->NUM\_COME)

REPLACE &pole WITH &pole+1 && Первично/Повторно

pole=FIELD(16+KARTA->DIRECT1)

REPLACE &pole WITH &pole+1 && Направляющие организации

\*--------------------------------------------------------------------

IF dep=2.OR.dep=11

IF KARTA->OLD<7

REPLACE C3 WITH C3+1 && Всего до 1 года

REPLACE C4 WITH C4+KARTA->ALL\_DAY && К/Д

IF KARTA->CITY\_VILL=2

REPLACE C5 WITH C5+1 && В том числе из села

REPLACE C6 WITH C6+KARTA->ALL\_DAY && К/Д

ENDIF

ELSE

IF KARTA->CITY\_VILL=2

REPLACE C9 WITH C9+1 && Из села старше 1 года

ENDIF

ENDIF

IF KARTA->OLD=1

pole=FIELD(43)

ELSEIF KARTA->OLD=2

ad=piece(KARTA->HOUR\_B,KARTA->MINS\_B,KARTA->HOUR\_END,KARTA->MINS\_END)

ad=KARTA->DATE\_END-KARTA->DATE\_B+IF(ad=1,1,IF(ad>=0,0,-1))

pole=FIELD(42+IF(ad<=14,2,IF(ad>14.AND.ad<=21,3,4)))

ELSE

pole=FIELD(44+KARTA->OLD)

ENDIF

\*--------------------------------------------------------------------

ELSE

IF KARTA->OLD<7

REPLACE C3 WITH C3+1 && Всего до 1 года

REPLACE C4 WITH C4+KARTA->ALL\_DAY && К/Д

IF KARTA->CITY\_VILL=2

REPLACE C5 WITH C5+1 && В том числе из села

REPLACE C6 WITH C6+KARTA->ALL\_DAY && К/Д

ENDIF

ELSEIF KARTA->OLD<11

REPLACE C7 WITH C7+1 && Всего до 14 лет

REPLACE C8 WITH C8+KARTA->ALL\_DAY && К/Д

IF KARTA->CITY\_VILL=2

REPLACE C9 WITH C9+1 && В том числе из села

REPLACE C0 WITH C0+KARTA->ALL\_DAY && К/Д

ENDIF

ELSE

REPLACE D1 WITH D1+1 && Всего 15 лет и старше

REPLACE D2 WITH D2+KARTA->ALL\_DAY && К/Д

IF KARTA->CITY\_VILL=2

REPLACE D3 WITH D3+1 && В том числе из села

REPLACE D4 WITH D4+KARTA->ALL\_DAY && К/Д

ENDIF

ENDIF

IF KARTA->OLD<=3

pole=FIELD(43)

ELSE

pole=FIELD(40+KARTA->OLD)

ENDIF

ENDIF

\*--------------------------------------------------------------------

REPLACE &pole WITH &pole+1 && Возраст

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

SELECT DIA66

SET RELATION TO

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

SELECT BUFF8

OT2="OTD"+ALLTRIM(STR(dep))+".TXT"

@ 13,25 SAY " СОЗДАЕТСЯ ОТЧЕТ : "+OT2+" "

IF dep#2.AND.dep#11

REPORT FORM &OT1D2 TO FILE &OT2D2 PLAIN

ENDIF

REPORT FORM &OT1D1 TO FILE &OT2D1 PLAIN

REPORT FORM &OT1 TO FILE &OT2 PLAIN

REPORT FORM OTCH.FRM TO FILE OTCH.TXT PLAIN

USE

corr\_ttl("OTCH.TXT",dep\_name,DTOC(\_DATE\_FROM),DTOC(\_DATE\_TILL))

link2("OTCH.TXT",OT2)

RENAME OTCH.TXT TO &OT2

link2(OT2,OT2D1)

IF dep#2.AND.dep#11

link2(OT2,OT2D2)

ENDIF

ELSEIF \_p=2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* КВАРТАЛЬНЫЕ ОТЧЕТЫ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OT1="OTCH"+ALLTRIM(STR(\_OTCH))+".FRM"

OT2="OTCH"+ALLTRIM(STR(\_OTCH))+".TXT"

IF f\_FRM()

DO CASE

\*-------------------------------------------------

CASE \_OTCH=1

\*-------------------------------------------------

SELECT DIA66

SET RELATION TO SHIFR INTO BUFF8

SELECT karta

SET RELATION TO NUM\_IB INTO DIA66

GO TOP

DO show\_st

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

DIA66->KOD1="1"

state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

ENDIF

IF KARTA->OLD>10 && СТАРШЕ 14 лет

IF KARTA->END1=1.OR.KARTA->END1=2

REPLACE COUNT1 WITH COUNT1+1 && ВЫПИСАНО

REPLACE A1 WITH A1+KARTA->ALL\_DAY && ПРОВЕДЕНО ИМИ ДНЕЙ

ELSE && KARTA->END1=3

REPLACE A2 WITH A2+1 && УМЕРЛО

ENDIF

ELSE && KARTA->OLD<=10 && ДО 14 лет

IF KARTA->END1=1.OR.KARTA->END1=2

REPLACE COUNT2 WITH COUNT2+1 && ВЫПИСАНО

REPLACE A3 WITH A3+KARTA->ALL\_DAY && ПРОВЕДЕНО ИМИ ДНЕЙ

IF KARTA->OLD<7

REPLACE A4 WITH A4+1 && ДЕТИ ДО 1 года

ENDIF

ELSE && KARTA->END1=3

REPLACE A5 WITH A5+1 && УМЕРЛО

IF KARTA->OLD<7

REPLACE A6 WITH A6+1 && ДЕТИ ДО 1 года

ENDIF

ENDIF

ENDIF

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

IF INKEY(0.01)=27

IF yesno(14," Прервать обработку ? ")=1

SELECT (sel)

RETURN 0

ENDIF

ENDIF

ENDDO

SET RELATION TO

SELECT DIA66

SET RELATION TO

SELECT BUFF8

PRIVATE \_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6 TO ;

\_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6

APPEND BLANK

REPLACE NUMBER WITH "|",NAMECL WITH "Всего :",;

SHIFRL WITH "0000",SHIFRR WITH "999 ",;

COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,A1 WITH \_A1,;

A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,A5 WITH \_A5,A6 WITH \_A6

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

grad1() && РАЗБИЕНИЕ КЛАССОВ НА ГРУППЫ

\*-------------------------------------------------

CASE \_OTCH=2.OR.\_OTCH=5

\*-------------------------------------------------

SELECT DIA66

SET RELATION TO SHIFR INTO BUFF8

SELECT KARTA

SET RELATION TO NUM\_IB INTO DIA66

GO TOP

DO show\_st

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

(KARTA->DATE\_IN-KARTA->DATE\_B+piece(KARTA->HOUR\_B,KARTA->MINS\_B,;

KARTA->HOUR\_IN,KARTA->MINS\_IN))<7.AND.DIA66->KOD1="1"

state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

ENDIF

IF LEFT(KARTA->MASSA,2)="00".OR.LEFT(KARTA->MASSA,2)=" ".AND.;

VAL(RIGHT(KARTA->MASSA,3))>500

REPLACE A1 WITH A1+1

IF KARTA->END1=3

REPLACE A2 WITH A2+1

IF (KARTA->DATE\_END-KARTA->DATE\_B+;

piece(KARTA->HOUR\_B,KARTA->MINS\_B,;

KARTA->HOUR\_END,KARTA->MINS\_END))<7

REPLACE A3 WITH A3+1

ENDIF

ENDIF

ELSE

REPLACE A4 WITH A4+1

IF KARTA->END1=3

REPLACE A5 WITH A5+1

IF (KARTA->DATE\_END-KARTA->DATE\_B+;

piece(KARTA->HOUR\_B,KARTA->MINS\_B,;

KARTA->HOUR\_END,KARTA->MINS\_END))<7

REPLACE A6 WITH A6+1

ENDIF

ENDIF

ENDIF

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

SELECT DIA66

SET RELATION TO

SELECT BUFF8

PRIVATE \_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6 TO ;

\_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6

APPEND BLANK

REPLACE NUMBER WITH "|",NAMECL WITH "Всего детей :",;

COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,A1 WITH \_A1,;

A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,A5 WITH \_A5,A6 WITH \_A6

grad1() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА ГРУППЫ

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6 TO ;

\_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6 FOR NUMBER="-"

GO TOP

\_COUNT1=COUNT1-\_COUNT1

\_COUNT2=COUNT2-\_COUNT2

\_A1=A1-\_A1

\_A2=A2-\_A2

\_A3=A3-\_A3

\_A4=A4-\_A4

\_A5=A5-\_A5

\_A6=A6-\_A6

APPEND BLANK

REPLACE NUMBER WITH "-",SHIFR WITH "774 ";

NAMECL WITH "Прочие болезни плода и новорожденного",;

COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,A1 WITH \_A1,;

A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,A5 WITH \_A5,A6 WITH \_A6

\*-------------------------------------------------

CASE \_OTCH=3

\*-------------------------------------------------

SELECT OP66

SET RELATION TO NUM\_IB INTO KARTA, TO SHIFR INTO BUFF8

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL

\_SHIFR\_ILL=OP66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR\_ILL

catalog(@\_SHIFR\_ILL,@txt)

REPLACE NAME WITH ALLTRIM(txt)

ENDIF

REPLACE COUNT1 WITH COUNT1+1

IF KARTA->OLD<=10

REPLACE COUNT2 WITH COUNT2+1

ENDIF

IF KARTA->END1=3

REPLACE A1 WITH A1+1

ENDIF

SELECT OP66

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

summ() && Суммирование по классам операций

\*-------------------------------------------------

CASE \_OTCH=4

\*-------------------------------------------------

SELECT BUFF8

APPEND BLANK

REPLACE NUMBER WITH "1"

REPLACE NAME WITH "ВЫПИСАНО"

APPEND BLANK

REPLACE NUMBER WITH "2"

REPLACE NAME WITH "ПЕРЕВЕДЕНО"

APPEND BLANK

REPLACE NUMBER WITH "3"

REPLACE NAME WITH "УМЕРЛО"

SELECT KARTA

GO TOP

PRIVATE OT1D1,OT2D1

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL

SELECT BUFF8

GOTO KARTA->END1

pole=FIELD(8+KARTA->OLD)

REPLACE &pole WITH &pole+1 && ВОЗРАСТ БОЛЬНЫХ

pole=FIELD(19+KARTA->RAION)

REPLACE &pole WITH &pole+1 && РАЙОН ПРОЖИВАНИЯ

REPLACE COUNT1 WITH COUNT1+KARTA->ALL\_DAY && ПРОВЕДЕНО ДНЕЙ

REPLACE COUNT2 WITH COUNT2+1 && ВСЕГО БОЛЬНЫХ

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

OT1D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".FRM" && OTCH\*1.FRM

OT2D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".TXT" && OTCH\*1.TXT

SELECT BUFF8

REPORT FORM &OT1D1 TO FILE &OT2D1 PLAIN

\*-------------------------------------------------

CASE \_OTCH=6.OR.\_OTCH=8

\*-------------------------------------------------

SELECT DIA66

SET RELATION to NUM\_IB into KARTA, TO SHIFR INTO BUFF8

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

DIA66->KOD1="1"

count=state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF \_OTCH=6.AND.KARTA->END1=2

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

ENDIF

REPLACE COUNT1 WITH COUNT1+1

ELSEIF \_OTCH=8.AND.KARTA->END1=3

pole=FIELD(8+KARTA->POL)

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

ENDIF

REPLACE &pole WITH &pole+1

ENDIF

SELECT DIA66

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

\*------------------------------------------------

CASE \_OTCH=7

\*------------------------------------------------

SELECT KARTA

SET RELATION TO SHIFR INTO BUFF8

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL

\_SHIFR=KARTA->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH ALLTRIM(txt)

ENDIF

REPLACE A3 WITH A3+1 && Всего

IF KARTA->OLD<=7 && До 1 года

REPLACE A2 WITH A2+1

IF KARTA->OLD<=2 && До 28 дней

REPLACE A1 WITH A1+1

ENDIF

ENDIF

pole=FIELD(11+KARTA->WHY)

REPLACE &pole WITH &pole+1 && Причины направления

pole=FIELD(15+KARTA->DIRECT1)

REPLACE &pole WITH &pole+1 && Направляющие организации

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

numb\_STR() && НУМЕРАЦИЯ СТРОК

\*------------------------------------------------

CASE (\_OTCH=9.AND.dep#14).OR.\_OTCH=10.OR.\_OTCH=12

\*------------------------------------------------

SELECT DIA66

SET RELATION to NUM\_IB into KARTA, TO SHIFR INTO BUFF8

GO TOP

PRIVATE OT1D1,OT2D1,OT1D2,OT2D2

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

KARTA->END1=3.AND.DIA66->KOD1="1"

IF (\_OTCH=9.OR.;

\_OTCH=10.AND.;

(KARTA->DATE\_END-KARTA->DATE\_B+;

piece(KARTA->HOUR\_B,KARTA->MINS\_B,KARTA->HOUR\_END,KARTA->MINS\_END)<7).OR.;

\_OTCH=12.AND.;

(KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,KARTA->HOUR\_END,KARTA->MINS\_END)<1))

count=state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

ENDIF

pole=FIELD(6+KARTA->POL)

REPLACE &pole WITH &pole+1 && ПОЛ УМЕРШИХ

IF \_OTCH=9.OR.\_OTCH=12

REPLACE B2 WITH B2+KARTA->ALL\_DAY && КОЛ-ВО ДНЕЙ, ПРОВЕДЕННОЕ ИМИ

IF \_OTCH=9.AND.KARTA->OLD\_D<7

REPLACE B3 WITH B3+KARTA->ALL\_DAY && ---"--- БОЛЬНЫМИ ДО 1 года

ENDIF

pole=FIELD(8+KARTA->OLD\_D)

ELSEIF \_OTCH=10

PRIVATE ad

ad=piece(KARTA->HOUR\_B,KARTA->MINS\_B,KARTA->HOUR\_END,KARTA->MINS\_END)

pole=FIELD(9+(KARTA->DATE\_END-KARTA->DATE\_B+;

IF(ad=1,1,IF(ad>=0,0,-1))))

ENDIF

REPLACE &pole WITH &pole+1 && ВОЗРАСТ УМЕРШИХ

pole=FIELD(21+KARTA->DIRECT1)

REPLACE &pole WITH &pole+1 && НАПРАВЛЯЮЩЕЕ УЧРЕЖДЕНИЕ

pole=FIELD(35+KARTA->RAION)

REPLACE &pole WITH &pole+1 && РАЙОН

IF \_OTCH=9

IF KARTA->ALL\_DAY=1

pole=FIELD(44+IF(KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,;

KARTA->HOUR\_END,KARTA->MINS\_END)<1,0,1))

ELSE

pole=FIELD(44+IF(KARTA->ALL\_DAY<4,KARTA->ALL\_DAY,4))

ENDIF

ELSEIF \_OTCH=10

IF KARTA->ALL\_DAY=1

pole=FIELD(44+IF(KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,;

KARTA->HOUR\_END,KARTA->MINS\_END)<1,0,1))

ELSE

pole=FIELD(44+KARTA->ALL\_DAY)

ENDIF

ELSE &&\_OTCH=12

PRIVATE t,d

STORE 0 TO t,d

t=KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,KARTA->HOUR\_END,KARTA->MINS\_END)

d=IF(t<(0.5/24),0,IF(t<(1/24),1,IF(t<3/24,2,IF(t<10/24,3,4))))

pole=FIELD(44+d)

RELEASE t,d

ENDIF

REPLACE &pole WITH &pole+1 && ПРОВЕДЕНО ДНЕЙ

SELECT DIA66

ENDIF

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

OT1D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".FRM" && OTCH\*1.FRM

OT2D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".TXT" && OTCH\*1.TXT

SELECT BUFF8

REPORT FORM &OT1D1 TO FILE &OT2D1 PLAIN

IF \_OTCH=9.OR.\_OTCH=10

OT1D2="OTCH"+ALLTRIM(STR(\_OTCH))+"2"+".FRM" && OTCH\*2.FRM

OT2D2="OTCH"+ALLTRIM(STR(\_OTCH))+"2"+".TXT" && OTCH\*2.TXT

REPORT FORM &OT1D2 TO FILE &OT2D2 PLAIN

link2(OT2D1,OT2D2) && СЛИЯНИЕ ДВУХ ФАЙЛОВ

ENDIF

\*------------------------------------------------------

CASE \_OTCH=9.AND.dep=14 && Дла Неонатального центра

\*------------------------------------------------------

SELECT DIA66

SET RELATION TO SHIFR INTO BUFF8

SELECT karta

SET RELATION TO NUM\_IB INTO DIA66

GO TOP

OT1="NEONAT.FRM"

OT2="NEONAT.TXT"

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF dep=KARTA->DEPARTMENT.AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

KARTA->END1=3.AND.DIA66->KOD1="1"

count=state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

ENDIF

pole=FIELD(6+KARTA->POL)

REPLACE &pole WITH &pole+1 && Пол

pole=FIELD(16+KARTA->DIRECT1)

REPLACE &pole WITH &pole+1 && Направляющие организации

REPLACE C3 WITH C3+1 && Всего умерло

REPLACE C4 WITH C4+KARTA->ALL\_DAY && К/Д

IF KARTA->OLD=1

pole=FIELD(43)

ELSEIF KARTA->OLD=2

ad=piece(KARTA->HOUR\_B,KARTA->MINS\_B,KARTA->HOUR\_END,KARTA->MINS\_END)

ad=KARTA->DATE\_END-KARTA->DATE\_B+IF(ad=1,1,IF(ad>=0,0,-1))

pole=FIELD(42+IF(ad<=14,2,IF(ad>14.AND.ad<=21,3,4)))

ELSE

pole=FIELD(44+KARTA->OLD)

ENDIF

REPLACE &pole WITH &pole+1 && Возраст

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

SELECT DIA66

SET RELATION TO

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

\*------------------------------------------------

CASE \_OTCH=11

\*------------------------------------------------

codif1("FULL",@f)

IF f=0

LOOP

ENDIF

SELECT DIA66

SET CURSOR OFF

SET RELATION to SHIFR into BUFF8

SELECT OP66

SET RELATION to NUM\_IB into KARTA, TO NUM\_IB INTO DIA66

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

KARTA->END1=3.AND.DIA66->KOD1="1"

state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

\_NUM\_IB=OP66->NUM\_IB

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

ENDIF

REPLACE COUNT1 WITH COUNT1+1 && ВСЕГО ОПЕРИРОВАННЫХ БОЛЬНЫХ

SELECT 0

USE CATO.DBF INDEX CATO ALIAS CATO

DO WHILE .T.

SEEK OP66->SHIFR

SELECT BUFF8

pole=FIELD(8+CATO->NUMBER)

REPLACE &pole WITH &pole+1

REPLACE COUNT2 WITH COUNT2+1 && ВСЕГО ОПЕРАЦИЙ

SKIP 1 ALIAS OP66

SELECT CATO

IF OP66->NUM\_IB#\_NUM\_IB

SKIP -1 ALIAS OP66

EXIT

ENDIF

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

USE

ENDIF

SELECT OP66

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

SELECT DIA66

SET RELATION TO

grad() && РАЗБИЕНИЕ БОЛЕЗНЕЙ НА КЛАССЫ

IF f=1

OT1="OTCH"+ALLTRIM(STR(\_OTCH))+"L"+".FRM"

ELSE

OT1D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".FRM" && OTCH\*1.FRM

OT2D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".TXT" && OTCH\*1.TXT

SELECT BUFF8

REPORT FORM &OT1D1 TO FILE &OT2D1 PLAIN

OT1D2="OTCH"+ALLTRIM(STR(\_OTCH))+"2"+".FRM" && OTCH\*2.FRM

OT2D2="OTCH"+ALLTRIM(STR(\_OTCH))+"2"+".TXT" && OTCH\*2.TXT

REPORT FORM &OT1D2 TO FILE &OT2D2 PLAIN

link2(OT2D1,OT2D2) && СЛИЯНИЕ ДВУХ ФАЙЛОВ

ENDIF

\*------------------------------------------------

CASE \_OTCH=13

\*------------------------------------------------

SELECT DIA66

SET RELATION to NUM\_IB into KARTA, TO SHIFR INTO BUFF8

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

DIA66->SHIFR>"0000".AND.DIA66->SHIFR<="1399".AND.DIA66->KOD1="1"

count=state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR

ENDIF

IF KARTA->END1=1.OR.KARTA->END1=2

REPLACE COUNT1 WITH COUNT1+1 && ОБЩЕЕ КОЛИЧЕСТВО ВЫБЫВШИХ

REPLACE A1 WITH A1+KARTA->ALL\_DAY && ПРОВЕДЕНО ИМИ ДНЕЙ

IF KARTA->OLD<7

REPLACE COUNT2 WITH COUNT2+1 && КОЛИЧЕСТВО ВЫБЫВШИХ ДО 1

ENDIF

ELSEIF KARTA->END1=3

REPLACE A2 WITH A2+1 && ОБЩЕЕ КОЛИЧЕСТВО УМЕРШИХ

REPLACE A3 WITH A3+KARTA->ALL\_DAY && ПРОВЕДЕНО ИМИ ДНЕЙ

IF KARTA->OLD<7

REPLACE A4 WITH A4+1 && КОЛИЧЕСТВО УМЕРШИХ ДО 1

ENDIF

ENDIF

SELECT DIA66

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

numb\_STR() && НУМЕРАЦИЯ СТРОК

\*------------------------------------------------

CASE \_OTCH=14

\*------------------------------------------------

SELECT DIA66

SET RELATION to NUM\_IB into KARTA, TO SHIFR INTO BUFF8

GO TOP

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL.AND.;

DIA66->SHIFR>"0000".AND.DIA66->SHIFR<="1399".AND.DIA66->KOD1="1"

count=state() && Поиск паталого-анатомического диагноза (если он есть)

\_SHIFR=DIA66->SHIFR

SELECT BUFF8

IF EOF()

APPEND BLANK

mkb(1,1,@\_SHIFR,@txt)

REPLACE NAME WITH txt

REPLACE SHIFR WITH \_SHIFR

ENDIF

pole=FIELD(8+KARTA->DEPARTMENT)

REPLACE &pole WITH &pole+1

SELECT DIA66

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

numb\_STR() && НУМЕРАЦИЯ СТРОК

\*-----------------------------------------------

CASE \_OTCH=15

\*-----------------------------------------------

SELECT KARTA

GO TOP

PRIVATE \_NAME,\_NUMBER

PRIVATE OT1D1,OT2D1

DO show\_st && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL

IF KARTA->RAION>1

SELECT BUFF8

IF KARTA->STATE=1

\_NUMBER=""

IF KARTA->PLACE=0

\_SHIFR="99 "

\_NAME="Прочие области и районы РФ"

ELSE

\_SHIFR=RIGHT(ALLTRIM(extra1(KARTA->PLACE,"PLCE")),4)

\_NAME=extra1(KARTA->PLACE,"PLCE")

ENDIF

IF KARTA->RAION=2

\_NUMBER="\*"

\_SHIFR="1000"

\_NAME="Московская область"

ENDIF

ELSE

\_NUMBER="\*"

\_SHIFR=SPACE(2)+STR(KARTA->STATE,2)

\_NAME=extra1(KARTA->STATE,"STTE")

ENDIF

SEEK \_SHIFR

IF !FOUND()

APPEND BLANK

REPLACE NUMBER WITH \_NUMBER,SHIFR WITH \_SHIFR,NAME WITH \_NAME

ENDIF

pole=FIELD(8+KARTA->DIRECT1)

REPLACE &pole WITH &pole+1 && НАПРАВЛЯЮЩЕЕ УЧРЕЖДЕНИЕ

pole=FIELD(23+KARTA->DEPARTMENT)

REPLACE &pole WITH &pole+1 && ОТДЕЛЕНИЯ БОЛЬНИЦЫ

pole=FIELD(38+KARTA->PASS)

REPLACE &pole WITH &pole+1 && Планово/экстренно

REPLACE COUNT1 WITH COUNT1+KARTA->ALL\_DAY && Проведено дней

REPLACE COUNT2 WITH COUNT2+1 && ВСЕГО ВЫПИСАНО

SELECT KARTA

ENDIF

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SELECT BUFF8

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6,A7,A8,A9,A0,;

B1,B2,B3,B4,B5,B6,B7,B8,B9,B0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C0 TO;

\_1,\_2,\_3,\_4,\_5,\_6,\_7,\_8,\_9,\_10,\_11,\_12,\_13,\_14,\_15,\_16,\_17,\_18,\_19,;

\_20,\_21,\_22,\_23,\_24,\_25,\_26,\_27,\_28,\_29,\_30,\_31,\_32

&& Суммирование по всем столбцам

APPEND BLANK

REPLACE SHIFR WITH " ",NAME WITH "Всего",COUNT1 WITH \_1,;

COUNT2 WITH \_2,A1 WITH \_3,A2 WITH \_4,A3 WITH \_5,A4 WITH \_6,;

A5 WITH \_7,A6 WITH \_8,A7 WITH \_9,A8 WITH \_10,A9 WITH \_11,A0 WITH \_12,;

B1 WITH \_13,B2 WITH \_14,B3 WITH \_15,B4 WITH \_16,B5 WITH \_17,;

B6 WITH \_18,B7 WITH \_19,B8 WITH \_20,B9 WITH \_21,B0 WITH \_22,;

C1 WITH \_23,C2 WITH \_24,C3 WITH \_25,C4 WITH \_26,C5 WITH \_27,;

C6 WITH \_28,C7 WITH \_29,C8 WITH \_30,C9 WITH \_31,C0 WITH \_32

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6,A7,A8,A9,A0,;

B1,B2,B3,B4,B5,B6,B7,B8,B9,B0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C0 TO;

\_1,\_2,\_3,\_4,\_5,\_6,\_7,\_8,\_9,\_10,\_11,\_12,\_13,\_14,\_15,\_16,\_17,\_18,\_19,;

\_20,\_21,\_22,\_23,\_24,\_25,\_26,\_27,\_28,\_29,\_30,\_31,\_32;

FOR SHIFR>" ".AND.SHIFR<"1000"

&& Суммирование столбцов по всем иностранцам

APPEND BLANK

REPLACE SHIFR WITH " 100",NAME WITH "Всего иностранцев",COUNT1 WITH \_1,;

COUNT2 WITH \_2,A1 WITH \_3,A2 WITH \_4,A3 WITH \_5,A4 WITH \_6,;

A5 WITH \_7,A6 WITH \_8,A7 WITH \_9,A8 WITH \_10,A9 WITH \_11,A0 WITH \_12,;

B1 WITH \_13,B2 WITH \_14,B3 WITH \_15,B4 WITH \_16,B5 WITH \_17,;

B6 WITH \_18,B7 WITH \_19,B8 WITH \_20,B9 WITH \_21,B0 WITH \_22,;

C1 WITH \_23,C2 WITH \_24,C3 WITH \_25,C4 WITH \_26,C5 WITH \_27,;

C6 WITH \_28,C7 WITH \_29,C8 WITH \_30,C9 WITH \_31,C0 WITH \_32

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6,A7,A8,A9,A0,;

B1,B2,B3,B4,B5,B6,B7,B8,B9,B0,C1,C2,C3,C4,C5,C6,C7,C8,C9,C0 TO;

\_1,\_2,\_3,\_4,\_5,\_6,\_7,\_8,\_9,\_10,\_11,\_12,\_13,\_14,\_15,\_16,\_17,\_18,\_19,;

\_20,\_21,\_22,\_23,\_24,\_25,\_26,\_27,\_28,\_29,\_30,\_31,\_32 FOR SHIFR>"1000"

&& Суммирование столбцов по всем областям РФ

APPEND BLANK

REPLACE SHIFR WITH "9990",NAME WITH "Всего по РФ",COUNT1 WITH \_1,;

COUNT2 WITH \_2,A1 WITH \_3,A2 WITH \_4,A3 WITH \_5,A4 WITH \_6,;

A5 WITH \_7,A6 WITH \_8,A7 WITH \_9,A8 WITH \_10,A9 WITH \_11,A0 WITH \_12,;

B1 WITH \_13,B2 WITH \_14,B3 WITH \_15,B4 WITH \_16,B5 WITH \_17,;

B6 WITH \_18,B7 WITH \_19,B8 WITH \_20,B9 WITH \_21,B0 WITH \_22,;

C1 WITH \_23,C2 WITH \_24,C3 WITH \_25,C4 WITH \_26,C5 WITH \_27,;

C6 WITH \_28,C7 WITH \_29,C8 WITH \_30,C9 WITH \_31,C0 WITH \_32

OT1D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".FRM" && OTCH\*1.FRM

OT2D1="OTCH"+ALLTRIM(STR(\_OTCH))+"1"+".TXT" && OTCH\*1.TXT

REPORT FORM &OT1D1 TO FILE &OT2D1 PLAIN

\*------------------------------------------------

CASE \_OTCH=16.OR.\_OTCH=17.OR.\_OTCH=18.OR.\_OTCH=19

\*------------------------------------------------

SELECT BUFF8

APPEND BLANK

SELECT KARTA

SET RELATION TO NUM\_IB INTO DIA66

GO TOP

DO show\_st

DO WHILE !EOF()

IF IF(dep=0,.T.,IF(dep=KARTA->DEPARTMENT,.T.,.F.)).AND.;

KARTA->DATE\_END>=\_DATE\_FROM.AND.KARTA->DATE\_END<=\_DATE\_TILL

SELECT BUFF8

IF \_OTCH=16

IF KARTA->END1=2

REPLACE A1 WITH A1+1 && ВСЕГО

IF KARTA->OLD<3 && НОВОРОЖДЕННЫЕ

REPLACE A2 WITH A2+1

ENDIF

ENDIF

IF DIA66->SHIFR="0000" && ОКАЗАВШИЕСЯ ЗДОРОВЫМИ

REPLACE A3 WITH A3+1

ENDIF

ELSEIF \_OTCH=17.AND.KARTA->END1=3

IF KARTA->OLD=1

REPLACE A1 WITH A1+1 && УМЕРЛО В ВОЗРАСТЕ 0-6 СУТОК

ENDIF

IF (KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,KARTA->HOUR\_END,KARTA->MINS\_END)<1)

IF (KARTA->DATE\_END-KARTA->DATE\_B+;

piece(KARTA->HOUR\_B,KARTA->MINS\_B,KARTA->HOUR\_END,KARTA->MINS\_END)<=1)

&& В ВОЗРАСТЕ ДО СУТОК

REPLACE A2 WITH A2+1

ENDIF

IF KARTA->OLD<7 && В ВОЗРАСТЕ ДО ГОДА

REPLACE A3 WITH A3+1

SELECT DIA66

state() && Поиск паталого-анатомического диагноза (если он есть)

DO WHILE DIA66->NUM\_IB=KARTA->NUM\_IB

IF DIA66->KOD1="1".AND.;

(DIA66->SHIFR>="4800".AND.DIA66->SHIFR<="4869")

SELECT BUFF8

REPLACE A4 WITH A4+1 && В ТОМ ЧИСЛЕ УМЕРЛО ОТ ПНЕВМОНИИ

EXIT

ENDIF

SKIP 1

ENDDO

ENDIF

ENDIF

ELSEIF \_OTCH=18.AND.(KARTA->SHIFR="410 ".OR.KARTA->SHIFR="412 ")

IF KARTA->TIME<3

REPLACE A1 WITH A1+1 && ВСЕГО ПОСТУПИЛО БОЛЬНЫХ ИНФАРКТОМ

ENDIF

IF KARTA->END1=3.AND.(KARTA->DATE\_END-KARTA->DATE\_IN+;

piece(KARTA->HOUR\_IN,KARTA->MINS\_IN,KARTA->HOUR\_END,KARTA->MINS\_END)<1)

REPLACE A2 WITH A2+1 && УМЕРЛО В ПЕРВЫЕ 24 ЧАСА

ENDIF

ELSEIF \_OTCH=19.AND.(KARTA->SHIFR>="6300".AND.KARTA->SHIFR<="6769")

IF KARTA->END1=3

REPLACE A1 WITH A1+1 && ВСЕГО УМЕРЛО БЕРЕМЕННЫХ,РОЖЕНИЦ И РОДИЛЬНИЦ

SELECT DIA66

state() && Поиск паталого-анатомического диагноза (если он есть)

DO WHILE DIA66->NUM\_IB=KARTA->NUM\_IB

IF DIA66->KOD1="1".AND.;

(DIA66->SHIFR>="6300".AND.DIA66->SHIFR<="6769")

SELECT BUFF8

REPLACE A2 WITH A2+1 && УМЕРЛО ОТ ОСЛОЖНЯЮЩИХ ЗАБОЛЕВАНИЙ

EXIT

ENDIF

SKIP 1

ENDDO

ENDIF

ENDIF

SELECT KARTA

ENDIF

SKIP 1

show\_din(count) && ПРОЦЕДУРА ПРЕДСТАВЛЕНИЯ ОБРАБОТКИ ЗАПИСЕЙ

ENDDO

SET RELATION TO

numb\_STR() && НУМЕРАЦИЯ СТРОК

\*------------------------------------------------

ENDCASE

\*------------------------------------------------

SELECT BUFF8

IF \_OTCH=6

DELETE FOR EMPTY(COUNT2)=.T.

PACK

ENDIF

@ 13,25 SAY " СОЗДАЕТСЯ ОТЧЕТ : "+OT2+" "

REPORT FORM &OT1 FOR IF(\_OTCH=1.OR.\_OTCH=2.OR.\_OTCH=5,;

!EMPTY(NUMBER),.T.) TO FILE &OT2 PLAIN

IF \_OTCH=9.OR.\_OTCH=10.OR.\_OTCH=11.OR.\_OTCH=12

REPORT FORM OTCH.FRM TO FILE OTCH.TXT PLAIN

USE

corr\_ttl("OTCH.TXT",dep\_name,DTOC(\_DATE\_FROM),DTOC(\_DATE\_TILL))

link2("OTCH.TXT",OT2)

RENAME OTCH.TXT TO &OT2

ELSE

USE

corr\_ttl(OT2,dep\_name,DTOC(\_DATE\_FROM),DTOC(\_DATE\_TILL))

ENDIF

IF \_OTCH=4.OR.\_OTCH=9.AND.dep#14.OR.;

\_OTCH=10.OR.\_OTCH=11.AND.f=2.OR.\_OTCH=12.OR.\_OTCH=15

link2(OT2,OT2D1)

ENDIF

ELSE

SELECT BUFF8

USE

LOOP

ENDIF

ENDIF

SET CURSOR ON

fileview(OT2,3,2,21,77,"N/BG",350)

do\_PRN()

RESTORE SCREEN FROM scr1

SET CURSOR OFF

ELSE

SELECT BUFF8

USE

ENDIF

ENDDO

RELEASE coun,c1,v1,v2,txt,seek,numb1,\_COUNTALL,rec

SELECT (sel)

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция нумерации строк в отчетном документе \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION numb\_STR

SELECT BUFF8

GO TOP

PRIVATE numb1

numb1=0

DO WHILE !EOF()

numb1=numb1+1

REPLACE NUMBER WITH STR(numb1,5)

SKIP 1

ENDDO

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция разбиения болезней на классы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION grad

lsl=SELECT()

SELECT 0

USE CLASS.DBF INDEX CLASS ALIAS CLASS

PRIVATE coun1,K,seek,\_COUNTALL,rec

coun1=RECCOUNT()

seek=" "

\_COUNTALL=0

rec=0

GO TOP

SELECT BUFF8

SET SOFTSEEK ON

FOR K=1 TO coun1

seek=CLASS->SHIFR\_LEFT

SEEK seek

IF !EOF()

IF BUFF8->SHIFR <= CLASS->SHIFR\_RIGH

numb1=numb1+1

rec=RECNO()

IF \_OTCH=1

\_SHIFR=SHIFR

\_COUNT1=COUNT1

\_COUNT2=COUNT2

\_A1=A1

\_A2=A2

\_A3=A3

\_A4=A4

\_A5=A5

\_A6=A6

APPEND BLANK

REPLACE SHIFR WITH \_SHIFR,COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,;

A1 WITH \_A1,A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,;

A5 WITH \_A5,A6 WITH \_A6

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6 TO ;

\_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6 ;

WHILE BUFF8->SHIFR <= CLASS->SHIFR\_RIGH

GOTO rec

REPLACE COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,A1 WITH \_A1,;

A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,A5 WITH \_A5,A6 WITH \_A6

ENDIF

REPLACE BUFF8->NUMBER WITH STR(numb1,2)

REPLACE BUFF8->NAMECL WITH CLASS->NAME\_CLASS

REPLACE BUFF8->SHIFRL WITH CLASS->SHIFR\_LEFT

REPLACE BUFF8->SHIFRR WITH CLASS->SHIFR\_RIGH

IF \_OTCH=6

SUM COUNT1 TO \_COUNTALL WHILE BUFF8->SHIFR <= CLASS->SHIFR\_RIGH

GO rec

REPLACE BUFF8->COUNT2 WITH \_COUNTALL

ENDIF

ENDIF

SKIP 1 ALIAS CLASS

ELSE

EXIT

ENDIF

NEXT

SET SOFTSEEK OFF

SELECT CLASS

USE

SELECT (lsl)

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция разбиения на группы ( для отчета N1,(N2 и N5) ) \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION grad1

lsl=SELECT()

SELECT 0

IF \_OTCH=1

USE GRUP1.DBF INDEX GRUP1 ALIAS GRUP

ELSE && для \_OTCH=2 и \_OTCH=5

USE GRUP2.DBF INDEX GRUP2 ALIAS GRUP

ENDIF

PRIVATE coun1,K,seek

coun1=RECCOUNT()

seek=" "

GO TOP

SELECT BUFF8

SET SOFTSEEK ON

FOR K=1 TO coun1

seek=GRUP->SHIFR\_LEFT

SEEK seek

IF !EOF()

IF BUFF8->SHIFR <= GRUP->SHIFR\_RIGH

IF !EMPTY(BUFF8->NUMBER)

SKIP 1 ALIAS BUFF8

ENDIF

rec=RECNO()

SUM COUNT1,COUNT2,A1,A2,A3,A4,A5,A6 TO ;

\_COUNT1,\_COUNT2,\_A1,\_A2,\_A3,\_A4,\_A5,\_A6 ;

WHILE BUFF8->SHIFR <= GRUP->SHIFR\_RIGH

GOTO rec

REPLACE COUNT1 WITH \_COUNT1,COUNT2 WITH \_COUNT2,A1 WITH \_A1,;

A2 WITH \_A2,A3 WITH \_A3,A4 WITH \_A4,A5 WITH \_A5,A6 WITH \_A6

REPLACE BUFF8->NUMBER WITH "-"

REPLACE BUFF8->NAMECL WITH GRUP->NAME\_GRUP

REPLACE BUFF8->SHIFRL WITH GRUP->SHIFR\_LEFT

REPLACE BUFF8->SHIFRR WITH GRUP->SHIFR\_RIGH

ENDIF

SKIP 1 ALIAS GRUP

ELSE

EXIT

ENDIF

NEXT

SET SOFTSEEK OFF

SELECT GRUP

USE

SELECT (lsl)

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция слияния двух текстовых файлов \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION link2

PARAMETERS F1,F2

RUN ("COPY &F1+&F2 &F1>NUL")

DELETE FILE &F2

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Представление на экране обработки записей БД ( начало ) \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE SHOW\_ST

@ 4,7 CLEAR TO 15,72

saycent(5,5,75," \*\*\* "+\_OTCH\_N+" \*\*\* ")

saycent(6,5,75,"по "+IF(dep=0,"всему стационару ","отделению "+dep\_name))

saycent(7,5,75,"за период с "+DTOC(\_DATE\_FROM)+" по "+DTOC(\_DATE\_TILL))

STORE 0 TO c1,v1,v2

coun=RECCOUNT()

v1=replicate(chr(178),60)

PRIVATE clr11

clr11=SETCOLOR()

SET COLOR TO (color1)

@ 8,8 CLEAR TO 15,71

@ 8,8 TO 15,71 DOUBLE

saycent(15,5,75," ESC - прервать обработку ")

@ 12,9 TO 14,70

@ 13,10 say v1

@ 9,10 TO 11,37

@ 10,11 SAY "ОБРАБОТАНО:"

@ 10,24 SAY 0

@ 9,41 TO 11,70

@ 10,42 SAY "ВСЕГО ЗАПИСЕЙ:"

@ 10,61 SAY coun

SET COLOR TO (clr11)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Представление на экране обработки записей БД ( динамика ) \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE SHOW\_DIN

PARAMETERS counts

c1=c1+counts

v2=replicate(chr(219),int(60\*(c1/coun)))

@ 13,10 SAY v2

@ 10,24 SAY c1

count=1

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Суммирование колонок по классам операций для отчета N3 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION summ

PRIVATE k,s,s1,n,A,B,C

SELECT BUFF8

SET SOFTSEEK ON

GO TOP

FOR k=2 TO 16

s=IF(k<10,"0"+STR(k,1),STR(k,2))+"00"

SEEK s

IF !FOUND()

APPEND BLANK

REPLACE SHIFR WITH s

catalog(@s,@txt)

REPLACE NAME WITH ALLTRIM(txt)

ENDIF

n=RECNO()

SKIP 1

s1=IF(k+1<10,"0"+STR(k+1,1),STR(k+1,2))+"00"

SUM COUNT1,COUNT2,A1 TO A,B,C WHILE SHIFR<s1

GO n

REPLACE COUNT1 WITH COUNT1+A,COUNT2 WITH COUNT2+B,A1 WITH A1+C

NEXT

SUM COUNT1,COUNT2,A1 TO A,B,C FOR RIGHT(SHIFR,2)="00"

APPEND BLANK

REPLACE SHIFR WITH "9999", NAME WITH "\*\*\* Всего \*\*\*",;

COUNT1 WITH COUNT1+A,COUNT2 WITH COUNT2+B,A1 WITH A1+C

SET SOFTSEEK OFF

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура навигации ( просмотра ) БД \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE navy

PRIVATE sel1,clr1,screen1

sel1=SELECT()

clr1=SETCOLOR()

menu1=1

D2=.F.

SELECT karta

SET SOFTSEEK ON

SET COLOR TO &color5

DO WHILE menu1#0

@ 7,8 CLEAR TO 14,72

SAVE SCREEN TO screen1

@ 8,15 PROMPT "ВВЕДИТЕ НОМЕР И/Б "

@ 9,15 PROMPT "ВВЕДИТЕ ФАМИЛИЮ БОЛЬНОГО "

@ 10,15 PROMPT "ВВЕДИТЕ ДАТУ ПОСТУПЛЕНИЯ "

@ 11,15 PROMPT "ТЕКУЩАЯ КАРТА "

@ 12,15 PROMPT "СЛЕДУЮЩАЯ КАРТА "

@ 13,15 PROMPT "ПРЕДЫДУЩАЯ КАРТА "

MENU TO menu1

IF menu1=1

SET CURSOR ON

@ 8,45 GET \_NUM\_IB PICTURE "@R 99/99999"

READ

SET CURSOR OFF

SEEK \_NUM\_IB

D2=EOF()

menu1=5

ELSEIF menu1=2

SET CURSOR ON

@ 9,45 GET \_FAM PICTURE "@K" VALID RUSSIAN(\_FAM)

READ

SET CURSOR OFF

SET FILTER TO FAM>=ALLTRIM(\_FAM)

GO TOP

D2=EOF()

menu1=5

SET FILTER TO

ELSEIF menu1=3

SET CURSOR ON

@ 10,45 GET \_DATE\_IN PICTURE "@D"

READ

SET CURSOR OFF

SET FILTER TO DATE\_IN=\_DATE\_IN

GO TOP

D2=EOF()

IF D2=.F.

menu1=1

@ 16,8 CLEAR TO 20,72

DO WHILE menu1#0.AND.!D2

\_NUM\_IB=NUM\_IB

\_FAM=FAM

\_DATE\_IN=DATE\_IN

DO first

@ 11,14 TO 14,40 DOUBLE

@ 12,15 PROMPT "СЛЕДУЮЩАЯ КАРТА "

@ 13,15 PROMPT "ПРЕДЫДУЩАЯ КАРТА "

MENU TO menu1

IF menu1=1

SKIP

D2=EOF()

ELSEIF menu1=2

SKIP -1

D2=BOF()

ENDIF

ENDDO

menu1=1

ENDIF

SET FILTER TO

ELSEIF menu1=5

SKIP

D2=EOF()

ELSEIF menu1=6

SKIP -1

D2=BOF()

ENDIF

@ 16,8 CLEAR TO 20,72

IF D2=.F.

\_NUM\_IB=NUM\_IB

\_FAM=FAM

\_DATE\_IN=DATE\_IN

DO first

ELSEIF D2=.T.

@ 17,25 TO 19,55 DOUBLE

@ 18,31 SAY "БОЛЬШЕ ЗАПИСЕЙ НЕТ!"

ENDIF

ENDDO

SET SOFTSEEK OFF

SELECT (sel1)

SET COLOR TO (clr1)

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ПРОВЕРКА ПРАВИЛЬНОСТИ ЗАПОЛНЕНИЯ КАРТЫ \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION all\_r

PRIVATE \_qui

\_qui=.F.

IF EMPTY(\_FAM)=.T.

message('e',"НЕ ВВЕДЕНА ФАМИЛИЯ ПАЦИЕНТА")

beg\_line=1

cur\_promp=2

ELSEIF EMPTY(\_DATE\_B)=.T.

message('e',"НЕ ВВЕДЕНА ДАТА РОЖДЕНИЯ")

beg\_line=1

cur\_promp=5

ELSEIF EMPTY(\_OLD)=.T.

message('e',"НЕ ВВЕДЕН ВОЗРАСТ")

beg\_line=1

cur\_promp=6

ELSEIF EMPTY(\_RAION)=.T.

message('e',"НЕ ВВЕДЕН РАЙОН ПРОЖИВАНИЯ")

beg\_line=1

cur\_promp=9

ELSEIF EMPTY(\_CITY\_VILL)=.T.

message('e',"НЕ ВВЕДЕН ПУНКТ <ЖИТЕЛЬ>")

beg\_line=1

cur\_promp=10

ELSEIF EMPTY(\_STATE)=.T.

message('e',"НЕ ВВЕДЕНО НАЗВАНИЕ ГОСУДАРСТВА <по умолчанию - РФ> ")

beg\_line=1

cur\_promp=12

ELSEIF EMPTY(\_DEPARTMENT)=.T.

message('e',"НЕ ВВЕДЕНO НАЗВАНИЕ ОТДЕЛЕНИЕ")

beg\_line=1

cur\_promp=13

ELSEIF EMPTY(\_KOIKA)=.T.

message('e',"НЕ ВВЕДЕН ПРОФИЛЬ КОЙКИ")

beg\_line=1

cur\_promp=14

ELSEIF EMPTY(\_DATE\_IN)=.T.

message('e',"НЕ ВВЕДЕНА ДАТА ПОСТУПЛЕНИЯ")

beg\_line=1

cur\_promp=17

ELSEIF EMPTY(\_DATE\_END)=.T.

message('e',"НЕ ВВЕДЕНА ДАТА ВЫПИСКИ")

beg\_line=20

cur\_promp=20

ELSEIF \_ALL\_DAY<0.AND.EMPTY(\_DATE\_END)=.F.

beg\_line=1

cur\_promp=17

message('e',"НЕСООТВЕТСТВИЕ МЕЖДУ ДАТАМИ ПОСТУПЛЕНИЯ И ВЫПИСКИ")

ELSEIF \_END1=3.AND.EMPTY(\_OLD\_D)=.T.

message('e',"НЕ ВВЕДЕН ВОЗРАСТ НА МОМЕНТ СМЕРТИ")

beg\_line=1

cur\_promp=18

ELSEIF EMPTY(\_END1)=.T.

message('e',"НЕ ВВЕДЕН ПУНКТ <ИСХОД>")

beg\_line=1

cur\_promp=19

ELSEIF EMPTY(\_NUM\_COME)=.T.

message('e',"НЕ ВВЕДЕНО КОЛИЧЕСТВО ГОСПИТАЛИЗАЦИЙ")

beg\_line=20

cur\_promp=22

\* ELSEIF EMPTY(\_DIA\_DIRECT)=.T.

\* message('e',"НЕ ВВЕДЕН НАПРАВЛЯЮЩИЙ ДИАГНОЗ")

\* beg\_line=20

\* cur\_promp=21

ELSEIF LEN(vars1[1])=0

message('e',"НЕ ВВЕДЕН ОСНОВНОЙ ДИАГНОЗ")

beg\_line=20

cur\_promp=23

ELSEIF AT("000.0",vars1[1])#0.AND.LEN(vars[1])>80

message('e',"ОШИБОЧНЫЙ ДИАГНОЗ")

beg\_line=20

cur\_promp=25

ELSEIF AT("000.0",vars1[1])#0.AND.LEN(vars1[2])#0

message('e',"ОШИБОЧНЫЙ ДИАГНОЗ")

beg\_line=20

cur\_promp=25

ELSE

\_qui=.T.

ENDIF

RETURN (\_qui)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Представление на экране основной информации из 66 формы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE first

IF !BOF().AND.!EOF()

@ 16,8 CLEAR TO 20,72

@ 17,15 SAY "НОМЕР И/Б :"+NUM\_IB

@ 18,15 SAY "ФАМИЛИЯ БОЛЬНОГО :"+ALLTRIM(FAM)

@ 19,15 SAY "ДАТА ПОСТУПЛЕНИЯ :"

@ 19,34 SAY DATE\_IN

ENDIF

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Каталог операций \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION catalog

PARAMETERS s,t

PRIVATE sel3,screen3,N3

sel3=SELECT()

SAVE SCREEN TO screen3

select 0

use cato.dbf index cato alias cato

SET SOFTSEEK ON

SEEK s

SET SOFTSEEK OFF

IF FOUND()

t=NAME\_ILL

ELSE

private NUILL,K

go top

nuill=RECCOUNT()

declare OPERATION[NUILL]

for K=1 to NUILL

operation[k]=NAME\_ILL

skip 1

next

release NUILL,K

@ 4,1 CLEAR TO 21,78

@ 4,1 TO 21,78

saycent(4,1,78," КАТАЛОГ ОПЕРАЦИЙ ")

N3=ACHOICE(5,2,20,77,operation,.T.,"",NUMBER-1)

IF LASTKEY()=27

RESTORE SCREEN FROM screen3

use

SELECT (sel3)

RETURN (-1)

ENDIF

GO N3

s=SHIFR

t=NAME\_ILL

ENDIF

RESTORE SCREEN FROM screen3

use

SELECT (sel3)

RETURN (0)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Процедура настройки каталогов \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROCEDURE recon

PRIVATE N4,N5,cod\_name

STORE 0 TO N4,N5

DO WHILE gotomain=.F.

cod\_name=SPACE(4)

codif1("CORR",@N4)

IF LASTKEY()=27

SET CURSOR OFF

RETURN

ELSEIF N4=1

cod\_name="RIGS"

ELSEIF N4=2

cod\_name="DIRS"

ELSEIF N4=3

cod\_name="STTE"

ELSEIF N4=4

cod\_name="HOSP"

ELSEIF N4=5

cod\_name="BIRS"

ELSEIF N4=6

cod\_name="RIZS"

ELSEIF N4=7

cod\_name="DEPS"

ELSEIF N4=8

cod\_name="KOIK"

ELSEIF N4=9

cod\_name="RIZ1"

ELSEIF N4=10

cod\_name="RIZ2"

ELSEIF N4=11

cod\_name="RIZ3"

ELSEIF N4=12

cod\_name="OLDS"

ELSEIF N4=13

cod\_name="PLCE"

ENDIF

codifM("CODIF",cod\_name,@N5)

ENDDO

RELEASE N4,N5,cod\_name

RETURN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Продедура работы с каталогами \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION codifM

PARAMETERS codfile,code\_name,code\_var

PRIVATE screen,sel,ret,i,k,svtx,maxlen,color,count,first,x1,x2,y1,y2

PRIVATE prom,prom1

IF !t\_qwerty

RETURN 0

ENDIF

SAVE SCREEN TO screen

SET CURSOR OFF

color=SETCOLOR()

sel=SELECT()

SET COLOR TO (color3)

SET EXACT OFF

SELECT &CODFILE

CLEAR TYPEAHEAD

prom= "ESC- отказ,ENTER-переименовать"

prom1="INS-добавить,DEL-удалить"

first=1

DO WHILE .T.

SEEK (code\_name)

IF !FOUND()

RETURN ""

ENDIF

svtx=ALLTRIM(TEXT)

maxlen=MAX(LEN(svtx),MAX(LEN(prom),LEN(prom1)))

COUNT WHILE SUBSTR(KEY,1,4)=SUBSTR(code\_name+' ',1,4) TO COUNT

count=count-1 && не учитываем заголовок

DECLARE A[count],B[count]

\* A[]-массив для текстов шаблонов

\* B[]-массив для номеров шаблонов

IF count=0

DECLARE A[1]

a[1]=" Кодификатор пуст,воспользуйтесь клавишей INS"

maxlen=MAX(maxlen,40)

ENDIF

SEEK(code\_name)

FOR k=1 TO COUNT

SKIP

A[K]=ALLTRIM(TEXT)

B[K]=SUBSTR(KEY,5)

maxlen=MAX(maxlen,LEN(A[K]))

NEXT

y1=12-ROUND(MIN(count,13)/2 +0.49,0)

x1=37-ROUND(MIN(maxlen,72)/2 +0.49,0)

\* рисование рамки и заголовка \*

SET COLOR TO (color3)

y2=MIN(y1+count+2,20)

x2=MIN(x1+maxlen+3,77)

RESTORE SCREEN FROM SCREEN

@ y1,x1,y2,x2 BOX singl+fon2

@ y2,x1,y2+3,x2 BOX "+-+¦--L¦"+fon2

saycent(y2+1,x1,x2,prom)

saycent(y2+2,x1,x2,prom1)

saycent(y1,x1,x2,svtx)

I=ACHOICE(y1+1,x1+1,y2-1,x2-1,a,.t.,"u\_key1",first)

IF i=0

ret=""

CLEAR TYPEAHEAD

EXIT

ELSE

DO CASE

CASE LASTKEY()=13.AND.COUNT>0 &&<ENTER>

SEEK(code\_name)

SKIP I

PRIVATE scr,col1,pict

pict=SPACE(LEN(TEXT))

scr=SAVESCREEN(10,9,12,70)

col1=SETCOLOR()

SET COLOR TO (color7)

@10,9,12,70 box singl+fon2

saycent(10,9,70,"ВВОДИТЕ НОВОЕ ИМЯ")

SET CURSOR ON

@ 11,10 GET pict

READ

PICT=STRTRAN(pict,'Н','H')

SET CURSOR OFF

SETCOLOR(col1)

RESTSCREEN(10,9,12,70,scr)

IF LASTKEY()#27.AND.!EMPTY(PICT) && ESC

REPLACE TEXT WITH pict

ENDIF

RELEASE scr,col1,pict

CASE LASTKEY()=22 &&<INS>

IF count>0

ins\_pic(code\_name,b[count])

ELSE

ins\_pic(code\_name,' ')

ENDIF

first=count+1

CASE LASTKEY()=7 &&<DEL>

IF count>0

del\_pic(code\_name,i)

ENDIF

first=i-1

ENDCASE

ENDIF

ENDDO

\*CLEAR TYPEAHEAD

REINDEX

RESTORE SCREEN FROM screen

SET COLOR TO (color)

SELECT(sel)

SET CURSOR OFF

RETURN ret

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Проверка наличия в текущей директории файла отчета \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION f\_FRM

PRIVATE log,screen

log=.T.

IF !FILE(OT1)

log=.F.

SAVE SCREEN TO screen

@ 8,8 CLEAR TO 15,71

@ 8,8 TO 15,71 DOUBLE

saycent(8,20,60,"ВНИМАНИЕ")

@ 11,15 SAY "ДЛЯ СОЗДАНИЯ ОТЧЕТА НЕОБХОДИМ ФАЙЛ :"+OT1

@ 12,15 SAY "УКАЗАННОГО ФАЙЛА НЕТ В РАБОЧЕЙ ДИРЕКТОРИИ"

INKEY(10)

RESTORE SCREEN FROM screen

ENDIF

RETURN (log)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция ввода отчетного периода \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION period

PRIVATE screen,M1,R1

R1=0

M1=1

SAVE SCREEN TO screen

SET CURSOR ON

@ 8,8 CLEAR TO 15,71

@ 8,8 TO 15,71 DOUBLE

DO WHILE .T.

saycent(8,20,60,"ВВЕДИТЕ ОТЧЕТНЫЙ ПЕРИОД")

@ 9,17 TO 11,34

@ 10,20 SAY "c " GET \_DATE\_FROM PICTURE "@D"

@ 9,47 TO 11,64

@ 10,50 SAY "по " GET \_DATE\_TILL PICTURE "@D"

@ 12,17 TO 14,64

@ 13,21 PROMPT " Ok "

@ 13,38 PROMPT " ПОВТОР "

@ 13,53 PROMPT " ОТКАЗ "

READ

MENU TO M1

IF M1=1

EXIT

ELSEIF M1=2

M1=1

ELSEIF M1=0.OR.M1=3

R1=1

EXIT

ENDIF

ENDDO

SET CURSOR OFF

RESTORE SCREEN FROM screen

RETURN (R1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Вывод отчетного документа на печать \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION do\_PRN

PRIVATE YN

YN=1

codif1("PRNT",@YN)

IF YN=2

SET CURSOR OFF

TYPE &OT2 TO PRINT

ENDIF

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция определения возраста пациента \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION y\_m\_day

PARAMETERS day\_bir,hour\_bir,mins\_bir,day\_bas,hour\_bas,mins\_bas

PRIVATE years,mons,days,screen,txt

SAVE SCREEN TO screen

txt=""

years="00"

@ 1,20 CLEAR TO 3,60

@ 1,20 TO 3,60

@ 2,22 SAY IF(choice=8," Возраст пациента :","Возраст на момент смерти:")

years=oldM(day\_bir,day\_bas)

IF VAL(years)>0

txt=years

IF VAL(years)=1

txt=txt+" год"

ELSEIF VAL(years)<5

txt=txt+" года"

ELSE

txt=txt+" лет"

ENDIF

ELSE

mons=INT((day\_bas-day\_bir)/30)

IF mons>0

txt=ALLTRIM(STR(mons))

IF mons=1

txt=txt+" месяц"

ELSEIF mons<5

txt=txt+" месяца"

ELSE

txt=txt+" месяцев"

ENDIF

ELSE

PRIVATE \_add

\_add=piece(hour\_bir,mins\_bir,hour\_bas,mins\_bas)

days=day\_bas-day\_bir+\_add

txt=ALLTRIM(STR(days))

IF days=1

txt=txt+" день"

ELSEIF days<5

txt=txt+" дня"

ELSE

txt=txt+" дней"

ENDIF

ENDIF

ENDIF

@ 2,50 SAY txt

vars[choice]=vars[choice]+"."

PRIVATE string2

string2=""

IF choice=8

context(@string2,promp[choice],vars[choice],length,New\_Str)

stuff1(@string,length,string2,choice,row,len(promp))

choice=9

vars[choice]=codif1("OLDS",@\_OLD)

ELSEIF choice=22

codif1("OLDS",@\_OLD\_D)

ENDIF

RESTORE SCREEN FROM screen

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция определения полных лет пациента \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION oldM

PARAMETERS b\_dat,today

PRIVATE old1

PRIVATE year1

SET CENTURY OFF

year1=year(today)-year(b\_dat)

if month(today)>month(b\_dat)

old1=alltrim(str(year1))

else

if month(today)<month(b\_dat)

old1=alltrim(str(year1-1))

else

if day(today)<day(b\_dat)

old1=alltrim(str(year1-1))

else

old1=alltrim(str(year1))

endif

endif

endif

RETURN old1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция перевода минут в сутки \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION piece

PARAMETERS H1,M1,H2,M2

PRIVATE P

P=0.00

P=((60\*H2+M2)-(60\*H1+M1))/1440

RETURN (P)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Коррекция заголовка отчетного документа \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION corr\_ttl

PARAMETERS \_file,\_str1,\_str2,\_str3

PRIVATE h,l,v

h=FCREATE("\_0000F",0)

FSEEK(h,0,0)

FWRITE(h,"Отделение: "+\_str1+CHR(13)+CHR(10),11+LEN(\_str1)+2)

FWRITE(h,"Отчетный период: "+\_str2+" - "+\_str3+CHR(13)+CHR(10),;

17+LEN(\_str2)+3+LEN(\_str3)+2)

FWRITE(h,"Дата формирования отчета : "+DTOC(\_today)+CHR(13)+CHR(10),;

27+LEN(DTOC(\_today))+2)

FCLOSE(h)

RUN ("COPY \_0000F+&\_file \_0000F>NUL")

DELETE FILE &\_file

RENAME \_0000F TO &\_file

RETURN 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Модуль: VIEWER.PRG**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Функция просмотра текстового файла в заданном окне - fileview. \*

\* Для перемещения текста в окне используются \*

\* только: \*

\* Параметры: \*

\* filename - имя файла, \*

\* wt,wl,wb,wr - окно просмотра, \*

\* color - цвет [необязательный параметр], \*

\* linewide - длина строки(гориз. скроллинг) [необязательный параметр]. \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function fileview

parameters filename,wt,wl,wb,wr,color,linewide

private col\_sv

col\_sv=setcolor()

if pcount()<6

color="W+/B,N/G,BG/N,RB+/B,BG/B"

endif

if pcount()<7

linewide=wr-wl+1

endif

set key 24 to cr

set key 18 to bl

set key 3 to bl

set key 29 to bl

set key 30 to bl

set key 31 to bl

if empty(color)

color="W+/B,N/G,BG/N,RB+/B,BG/B"

endif

setcolor(color)

private f\_mov

private fh,file\_len,file\_down,file\_up

private blok,pos\_str,pos\_cur

private lines,old\_line,count,cnt\_pos

private buf,p,wt,wl,wb,wr

private str\_vid,p\_vid

private buf1,buf2

buf="buf1"

blok=2000

pos\_str=wb-wt+1

pos\_cur=wb-wt+1

lines=0

count=0

cnt\_pos=0

old\_line=0

last=chr(13)+chr(10)

f\_mov=0

fh=fopen(filename,0)

if ferror()#0

@ 1,2 say "Ошибка при открытии файла "+filename

return(0)

endif

file\_len=fseek(fh,0,2)

fseek(fh,0,0)

buf1=freadstr(fh,blok)

file\_down=blok

file\_up=-1

str\_vid=buf1

p\_vid= rat(last,str\_vid)

str\_vid=left(str\_vid,p\_vid-1)

do while .T.

clear typeahead

memoedit(STRTRAN(str\_vid,"Н","H"),wt,wl,wb,wr,.F.,"mod",linewide,'',pos\_str,0,pos\_cur,0)

if lastkey()=27

exit

endif

do case

case f\_mov=1

str\_vid=&buf

buf=if(buf="buf1","buf2","buf1")

fseek(fh,file\_down,0)

file\_down=file\_down+blok

file\_up=file\_down-3\*blok

&buf=freadstr(fh,blok)

str\_vid=str\_vid+&buf

pos\_str=lines-old\_line+1

pos\_cur=wb-wt+1

old\_line=pos\_str-1

p\_vid= rat(last,str\_vid)

str\_vid=left(str\_vid,p\_vid-1)

count=count+1

if count>cnt\_pos

cnt\_pos=cnt\_pos+1

p="pos"+alltrim(str(cnt\_pos))

private &p

&p=pos\_str

endif

case f\_mov=-1

fseek(fh,file\_up,0)

file\_down=file\_down-blok

file\_up=file\_down-3\*blok

&buf=freadstr(fh,blok)

str\_vid=&buf

buf=if(buf="buf1","buf2","buf1")

str\_vid=str\_vid+&buf

count=count-1

p="pos"+alltrim(str(count))

pos\_str=&p+wb-wt+1

pos\_cur=wb-wt+1

p\_vid= rat(last,str\_vid)

str\_vid=left(str\_vid,p\_vid-1)

otherwise

endcase

enddo

fclose(fh)

set key 24

set key 18

set key 3

set key 29

set key 30

set key 31

setcolor(col\_sv)

RETURN(0)

function mod

parameters mode,line,col

private key

key=lastkey()

do case

case key=13 .and. line=lines .and. file\_down<file\_len

f\_mov=1

keyboard chr(23)

return(0)

case key=5 .and. line<=wb-wt+2 .and. file\_up>-1

f\_mov=-1

keyboard chr(23)

return(0)

otherwise

lines=line

endcase

return(0)

procedure cr

keyboard chr(13)

return

procedure bl

keyboard chr(32)

return