ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Глава 1. ОСНОВНАЯ ЧАСТЬ

1.1 Назначение и область применения

1.2 Постановка задачи

1.3 Описание и обоснование выбора состава технических и программных средств

1.4 Информационная модель и ее описание

1.4.1 Диаграмма потоков данных

1.4.2 Инфологическая модель данных

1.4.3 Датологическая модель данных

1.4.4Реализация и обоснование нормализации базы данных

Глава 2. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

2.1 Описание алгоритма программы

Руководство оператора

2.2.1 Назначение программы

2.2.2 Условия выполнения программы

2.2.3 Выполнение программы

2.2.4 Сообщения оператору

Глава 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ

ГЛАВА 4. МЕРОПРИЯТИЯ ПО ОБЕСПЕЧЕНИЮ ТЕХНИКИ БЕЗОПАСНОСТИ И ПОЖАРНОЙ БЕЗОПАСНОСТИ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ

ВВЕДЕНИЕ

# Дипломный проект на тему: Информационная система рабочее место менеджера таксопарка "Семерочка" разрабатывается на основе учебного плана специальности 230103 "Автоматизированные системы обработки информации и управления" Сосенского радиотехнического техникума.

Целью данного дипломного проекта является автоматизация работы менеджера по работе с клиентами таксопарка.

Менеджеру таксопарка необходимо постоянно заполнять данные об обратившихся клиентах, отслеживать поездки водителей. Для этого ему нужно постоянно вести переговоры с водителем, знать время отправления и маршрут. Все это менеджеру необходимо фиксировать. Так как он не может это постоянно запоминать, следовательно, использует бумаги, журналы для записей, в которых информация может быть по ошибке записана не куда нужно, либо совсем потеряна.

В работе менеджера возникает много проблем, для этого необходимо автоматизировать данный процесс.

Автоматизация данного процесса облегчит работу менеджера по работе с клиентами, сократиться время её выполнения, что, в свою очередь, увеличит производительность.

Для достижения поставленной цели нужно решить следующие задачи: необходимо создание программы, которая поможет менеджеру быстро заполнить данные о клиентах, которые будут храниться и при необходимости, возможно, осуществить поиск нужного клиента. Менеджер сможет видеть перед собой все данные о любой поездке: время, маршрут. Так же программа должна позволять отслеживать, кто из водителей занят, а кто свободен. При использовании данной программы значительно уменьшается время обслуживания клиентов таксопарка, упрощается способ хранения и доступа к данным. Поиск аналогов программы результатов не дал.

Глава 1. ОСНОВНАЯ ЧАСТЬ

1.1 Назначение и область применения

Таксопарк "Семерочка" создан для осуществления пассажирских перевозок по заказу клиента. Перевозки осуществляются по всей территории России. На время осуществления заказа в распоряжение клиенту предоставляется машина с водителем. Штат сотрудников таксопарка представлен: директором, работником отдела кадров, работником транспортного отдела, менеджером по работе с клиентами и водителями.

Структурная схема таксопарка представлена в виде схемы: (см. Приложение 1).

В должностные обязанности директора входит:

* организация текущего и перспективного планирования деятельности таксопарка;
* проведение рекламных компаний по ознакомлению населения об условиях предоставления услуг таксопарком;
* разработку необходимой методической документации;
* осуществление контроля качества за процессом оказания услуг и объективностью оценки стоимости поездки;
* обеспечение своевременного составления установленной отчетной документации, и т.д.

Свои профессиональные обязанности директор выполняет, взаимодействуя со следующим подразделениями таксопарка:

Таблица 1. Подразделения таксопарка, с которыми взаимодействует директор

|  |  |
| --- | --- |
| Подразделение таксопарка | Информационные процессы |
| Отдел кадров | * передача информации о сотрудниках; |
| Транспортный отдел | * передача информации о поставленных на учет и снятых с учета автомобилей; |
| Отдел по работе с клиентами | * передача информации о клиентах и поездках; |

Разрабатываемая информационная система предназначена для автоматизации труда менеджера по работе с клиентами таксопарка "Семерочка". При организации работы таксопарка важным аспектом является мобильная, квалифицированная работа сотрудников, способных организовать процесс обслуживания клиентов как можно быстрее и качественней.

В обязанности менеджера по работе с клиентами входят следующие функции:

* контроль поступивших в распоряжение таксопарка машин и водителей;
* подготовка статистической информации по требованиям директора таксопарка для принятия им управленческих решений.
* регистрация клиентов обратившихся в таксопарк

При обращении клиента в таксопарк происходит регистрация его личных данных в базу данных.

В таксопарке "Семерочка" традиционно использовано программное обеспечение Microsoft Office и имеет следующие характеристики:

* вся информация находится в текстовых файлах формата Word, таблицах Excel или бумажных носителях;
* отсутствует система защиты от несанкционированного доступа;
* сбор аналитической информации производится путем анализа информации, хранящейся на бумажных носителях;
* контроль поступления и убытия машин и водителей осуществляется путем регистрации информации на бумажных носителях;

Все выше перечисленные недостатки позволяют сделать вывод о необходимости автоматизации труда менеджера по работе с клиентами таксопарка "Семерочка". Целью использования вычислительной техники является устранение всех вышеперечисленных недостатков, а так же автоматизация обработки и анализа первичной и промежуточной информации и представление итоговой информации в виде отчетов.

Работа менеджера по работе с клиентами таксопарка – это процесс, происходящий во времени, который можно разделить на последовательные этапы:

* этап оформления документов клиента;
* этап формирования списков машин и водителей;
* этап оформления заказа;
* этап корректировки информации
* этап выдачи информации о поездках

В каждом этапе происходят информационные процессы, которые сведены в таблицу:

Таблица 2. Информационные процессы этапов.

|  |  |
| --- | --- |
| Этап | Информационные процессы |
| 1. Этап оформления документов клиента | * сбор анкетных и других данных у клиента; |
| 1. Этап формирования списков машин и водителей | * регистрация новых машин и водителей; * снятие с учета машин и водителей; |
| 1. Этап оформления заказа | * Сохранение информации о маршруте; * Изменение статуса о машине, идущей в рейс * Изменение статуса о водителе, идущем в рейс |
| 1. Этап корректировки информации | * Поиск и выбор необходимого водителя из списка; * Корректировка данных о количестве поездок выбранного водителя; |
| 1. Этап выдачи информации о поездках | * Выдача отчёта по осуществлённым поездкам; * Выдача отчёта по осуществлённым поездкам за определённое число; * Выдача отчёта по количеству осуществлённых поездок водителя; |

Проанализировав этапы и их информационные процессы, автоматизации подлежат следующие функции:

* ввод исходных данных клиента при оформлении поездки;
* поиск необходимых клиентов по базе данных;
* постановка на внутрифирменный учет машин;
* снятие машин с внутрифирменного учета;
* изменение информации о машинах;
* изменение статуса машин и водителей;
* принятие на работу новых водителей;
* увольнение с работы водителей;
* принятие на работу новых операторов;
* увольнение с работы операторов;
* ввод и сохранение данных о маршруте поездки;
* вывод информации по поездкам на печать;
* изменение и обнуление количества поездок каждого водителя;
* вывод информации количества поездок каждого водителя на печать;
* предоставление справки о работе программы;

1.2 Постановка задачи

Автоматизация управленческой деятельности предполагает формирование электронной системы организации. Билл Гейтс, определяет электронную "нервную систему" любой организации как среду, автоматизирующую исполнение заранее намеченных действий и событий, планирование и учет, позволяющую своевременно реагировать на незапланированные события и изменения ситуации и дающую, таким образом, огромные преимущества в конкуренции и возможность эффективно работать.[1]

Основными целями разработки системы могут служить:

* сокращение времени обработки входной информации и получения результатных показателей;
* повышение степени достоверности выходной информации;
* снижение трудоемкости и количества ошибок при решении задачи.

В программе должны быть реализованы следующие функции:

* ввод исходных данных клиента при оформлении поездки;
* поиск необходимых клиентов по базе данных;
* постановка на внутрифирменный учет машин;
* снятие машин с внутрифирменного учета;
* изменение информации о машинах;
* изменение статуса машин и водителей;
* принятие на работу новых водителей;
* увольнение с работы водителей;
* принятие на работу новых операторов;
* увольнение с работы операторов;
* ввод и сохранение данных о маршруте поездки;
* вывод информации по поездкам на печать;
* изменение и обнуление количества поездок каждого водителя;
* вывод информации количества поездок каждого водителя на печать;
* предоставление справки о работе программы;

Хранилищем входной информации является:

* таблица Машины, предназначенная для хранения информации о машинах принадлежащих таксопарку;
* таблица Водители, предназначенная для хранения информации о водителях, работающих в таксопарке;
* таблица Операторы, предназначенная для хранения информации об операторах, работающих в таксопарке;
* таблица Клиенты, предназначенная для хранения информации о клиентах, обратившихся в таксопарк.

Хранилищем выходной информации является:

* таблица Поездки, предназначенная для хранения информации о поездках.

Окна обрабатывающие входную информацию представлены в таблице 3:

Таблица 3. Перечень окон, хранящих входную информацию.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Название окна | Название таблицы | Примечания |
| 1 | Машины "Семёрочка" | Машины | Для регистрации машин таксопарка |
| 2 | Водители "Семёрочка" | Водители | Для регистрации водителей таксопарка |
| 3 | Оформление поездки | Клиенты  Машины  Водители  Поездки | Для регистрации поездок |

Окна обрабатывающие входную информацию представлены в таблице 4:

Таблица 4. Перечень окон, хранящих служебную информацию.

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Название окна | Название таблицы | Примечания |
|  | Статус | Машины  Водители | Для изменения статуса водителей и машин |
|  | Корректировка поездок | Водители | Для корректировки информации о поездках осуществлённых водителем |

Окна обрабатывающие выходную информацию представлены в таблице 5:

Таблица 5. Перечень таблиц, хранящих выходную информацию.

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Название окна | Название таблицы | Примечания |
|  | Оформление поездки | Клиенты  Машины  Водители  Поездки | Для регистрации поездок |

Окна формирования отчётов представлены в таблице 6:

Таблица 6. Перечень окон формирования отчетов.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Название окна | Название таблицы | Примечание |
|  | Выбор даты отчёта | Поездки | Для вывода отчёта по всем осуществлённым поездкам |
|  | Корректировка поездок | Водители | Для вывода отчёта о корректировке поездок водителей |

Схема взаимосвязи программных модулей и информационных хранилищ. ( Приложение 2)

# 1.3 Описание и обоснование выбора состава технических и программных средств

Общую структуру информационной системы можно рассматривать как совокупность обеспечивающих подсистем. Техническое обеспечение – это комплекс технических средств, предназначенных для работы информационной системы. К настоящему времени сложились две основные формы организации технического обеспечения: централизованная и частично или полностью децентрализованная.

Перспективным подходом в настоящее время считается частично децентрализованный подход, применяя как персональные компьютеры, так и большие электронно-вычислительные машины с использованием распределенных сетей. Изучив предметную область и техническое оснащение таксопарка "Семерочка", а так же перспективные планы на будущее, был сделан вывод – применение сетевой технологии в данном случае требует существенные финансовые вложения. Существующее техническое оснащение: персональный компьютер класса Pentium IV, принтер марки Canon 100 и отсутствие компьютерных сетей позволяет сделать вывод, что наиболее приемлемой является децентрализованная форма организации технического обеспечения.

Программное обеспечение – совокупность программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств. В состав программного обеспечения входят общесистемные и специальные программные продукты. К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для типовых задач обработки информации. Информационное обеспечение, реально существующее на рабочем месте методиста – операционная система Windows 98 и Microsoft Office. В связи с этим тип файлов базы данных был выбран .mdb (MS Access).

Специального программного обеспечения для функционирования разрабатываемой информационной системы не требуется, кроме его исходного файла Start.exe и файла Baza.mdb. Подключение баз данных проведено с использованием технологии ADO, разработанной фирмой Microsoft и необходимые драйвера встроены в используемую операционную систему.

В реальных условиях проектирование – это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений. В настоящее время широкое распространение получила RAD-методология – методология быстрой разработки приложений. Данная методология охватывает все этапы жизненного цикла современных информационных систем.

RAD-технология позволяет при разработке и создании информационной системы применять спиральную модель жизненного цикла информационной системы, т.е. использовать прототипы. Возможность согласования прототипов системы с непосредственным будущим пользователем системы позволяет не терять общей картины проектируемой системы, контролировать требования заказчика и оперативно вносить необходимые корректировки функциональных возможностей системы. Визуальные инструменты RAD позволяют создавать современные пользовательские интерфейсы с минимальным написанием кодов программ.

Программные средства RAD-разработки делятся на универсальные и специализированные. Специализированные средства разработки ориентированы только на создание приложений баз данных и, как правило, привязаны к вполне определенным системам управления баз данных (Power Builder фирмы Sybase, Visual FoxPro фирмы Microsoft).

Среди универсальных систем визуального программирования наибольшее распространение получили: Borland Delphi, Borland C++, Visual Basic, с помощью которых можно разработать приложения практически любого типа, причем программы, разработанные такими средствами в состоянии взаимодействовать практически с любой системой управления базами данных. Возможности таких систем при разработке пользовательского интерфейса слились с созданием прототипов системы. При активном участии пользователя системы, способного корректировать как результаты расчетов, так и собственные требования к системе, сроки изготовления программы сокращаются. Наследованное универсальными системами событийного программирования предполагает описание различных событий различных объектов, установленных на форме-контейнере элементов управления. С помощью событийного программирования можно как обрабатывать информацию, хранящуюся в базе данных, так и создавать, редактировать, модифицировать, удалять объекты баз данных.

Как и любая технология, методология RAD имеет ряд своих недостатков и ограничений. Но при разработке относительно небольших систем, разрабатываемых под конкретную организацию, она достаточно эффективна.

# 1.4 Информационная модель и ее описание

1.4.1 Диаграмма потоков данных

Проведя анализ предметной области путем изучения вышеперечисленной информации, были выявлены следующие внешние сущности:

* Клиент, для которого осуществляется рейс;
* Менеджер по работе с клиентами;
* Водители;
* Начальник гаража;
* Отдел кадров.

Накопителями данных являются:

* Клиенты;
* Водители;
* Поездки;
* Машины;
* Операторы.

Информационная система разбита на четыре логических подсистемы:

* система регистрации клиентов;
* система формирования поездок;
* система администрирования;
* система формирования отчетов;

Потоки данных представлены в виде схемы и подробного словесного описания: (см. Приложение 3)

1.4.2 Инфологическая модель данных

Каждая сущность имеет неограниченное количество атрибутов, но, проанализировав требования к системе и осуществив детализацию хранилищ данных, будущую модель можно представить в виде связанных между собой отношениями сущностей:

1. Машины:

* Идентификационный номер машины;
* Марка машины;
* Цвет машины;
* Государственный номер машины;

1. Клиенты:

* Идентификационный номер клиента;
* ФИО клиента;
* Номер паспорта клиента;
* Серия паспорта
* Стаж
* Домашний адрес;
* Телефон.

1. Поездки:

* Номер поездки по порядку;
* Номер оператора;
* Идентификационный номер машины;
* Идентификационный номер водителя;
* Идентификационный номер клиента;
* Время отправления;
* Пункт отправления;
* Пункт назначения.

1. Водители:

* Идентификационный номер водителя;
* ФИО водителя;
* Номер паспорта водителя;
* Серия паспорта водителя;
* Телефон водителя.

1. Операторы:

* Идентификационный номер оператора;
* ФИО оператора;
* Домашний адрес;
* Домашний телефон;
* Номер паспорта;
* Пароль.

Полученная модель данных графически представлена инфологической моделью (Приложение 4).

1.4.3 Датологическая модель данных

Датологическая модель данных строится на основе разработанной инфологической модели и наиболее приемлемой для дальнейшей разработки является реляционная модель данных. Схема датологической модели (см. Приложение 5).

1.4.4 Реализация и обоснование нормализации базы данных

На этапе анализа необходимо подробное исследование как будущих функциональных возможностей разрабатываемой системы, так и информации, необходимой для их выполнения. Поэтому особое внимание было уделено как полноте информации, так и поиску противоречивой, дублирующей или неиспользуемой информации. После проведения нормализации базы данных таблицы "Операторы" и "Пароль" были объединены в таблицу "Операторы".

Глава 2. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

2.1 Описание алгоритма программы

Схема технологического процесса сбора, передачи, обработки и выдачи информации представлена в Приложении 6.

Алгоритм работы программы представлен в виде блок-схемы, демонстрирующей кнопки вызова форм. После запуска файла Start.exe на экране выводится форма Проверка прав (блок 2), которая предлагает ввести пароль. На форме находятся несколько кнопок:

* "Выход" (блок 3) осуществляет закрытие приложения;
* "Далее" (блок 4) если пароль введен правильно, то запускается главная кнопочная форма "Семёрочка" (блок 5), содержащая кнопки быстрого доступа:
* "Выход" (блок 6) осуществляет закрытие приложения;
* "Водители" (блок 7) позволяет редактировать, добавлять и удалять информацию о водителях (Форма Водители - блок 8);
* "Машины" (блок 10) запускает форму Машины (блок 11) для редактирования, добавления и удаления информации о машинах;
* "Операторы" (блок 13) активизирует форму Операторы (блок 14) для редактирования, добавления и удаления информации об операторах;
* "Новый заказ" (блок 16) активизирует форму Оформление поездки (блок 17) для заполнения информации поездке;
* "Статус" (блок 19) активизирует форму Статус (блок 20) для изменения статуса машин и водителей;
* "Корректировка поездок" (блок 22) активизирует форму Корректировка поездок (блок 23) для изменения информации о поездках осуществленных водителем;
* "Отчет" (блок 25) активизирует форму Реестр (блок 26) для просмотра информации о поездках осуществленных водителем
* "Отчет по поездкам" (блок 27) активизирует форму Выбор даты отчета (блок 28) для выбора даты отчета;
* "Напечатать отчет" (блок 30) активизирует форму Отчет по поездкам (блок 31) для просмотра информации об осуществленных поездках;

Блок – схема программы представлена в Приложении 8.

# 2.2 Руководство оператора

2.2.1 Назначение программы

Данная программа предназначена для автоматизации рабочего места менеджера по работе с клиентами таксопарка. Людям, работающим в данной сфере, приходится иметь дело с оформлением большого количества документов, необходимостью проверять наличие свободных машин, а как следствие к значительному увеличению времени обслуживания клиентов. При использовании этой программы менеджеру по работе с клиентами не придется постоянно искать нужную информацию в "куче" бумаг, вручную отмечать прибывшие из рейса, и уходящие в рейс машины, и др. что положительно отразиться на качестве и скорости оказания услуг.

2.2.2 Условия выполнения программы

Данное программное обеспечение имеет весьма "скромные" системные требования. Для обеспечения нормальной работоспособности программы необходимо иметь компьютер с конфигурацией не ниже:

* Процессор – Pentium 1;
* Оперативная память - 64 Мб;
* Свободное место на диске – 30 Мб;
* Монитор;
* Клавиатура;
* Мышь;
* Принтер.

Данная программа работает под управлением семейства операционных систем Windows. Также на компьютере должно быть установлено ПО MS Office(не ниже 2000).

Установка данной программы осуществляется путем копирования файлов программы: Baza.mdb и Start.exe в папку созданную на любом из локальных дисков.

2.2.3 Выполнение программы

Для начала работы необходимо запустить файл Start.exe, заранее скопированный вместе с файлом Baza.mdb в отдельную папку. После запуска файла Start.exe на экране появится окно, в котором необходимо выбрать логин, ввести пароль и нажать кнопку "Далее" или Enter, для того, чтобы войти в программу. Если вы хотите выйти из программы, то нужно нажать кнопку "Выход". После того как введен правильный пароль и нажата кнопка "Далее" или Enter, на экране появится главное окно программы. Дальше нужно выбрать желаемое действие:

Для выполнения, каких либо действий над водителями нужно нажать кнопку "Водители", после чего откроется окно, в котором с водителями можно выполнять следующие операции:

Просмотр данных о водителях осуществляется путем нажатия кнопок "Вперед" и "Назад", либо вследствие прокручивания списка; Добавление водителей. Для добавления водителя необходимо нажать кнопку "Добавить", и ввести данные в поля ввода данных, после этого нужно нажать кнопку "Сохранить", для сохранения данных;

* Удаление водителей. Для удаления водителя необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно удалить, и нажать кнопку "Удалить";
* Изменение данных о водителе. Для изменения данных необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно изменить, и нажать кнопку "Изменить", после этого в поле для ввода информации произвести изменения и нажать кнопку "Сохранить" для сохранения изменений.
* Для отмены нежелаемых действий необходимо нажать кнопку "Отмена";
* Для выхода на главную кнопочную форму – кнопка "На главную".

Для выполнения, каких либо действий над операторами нужно нажать кнопку "Операторы", после чего откроется окно, в котором с операторами можно выполнять следующие операции:

* Просмотр данных об операторах осуществляется путем нажатия кнопок "Вперед" и "Назад";
* Добавление оператора. Для добавления оператора необходимо нажать кнопку "Добавить", и ввести данные в поля ввода данных, после этого нужно нажать кнопку "Сохранить", для сохранения данных;
* Удаление операторов. Для удаления оператора необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно удалить, и нажать кнопку "Удалить";
* Изменение данных об операторе. Для изменения данных необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно изменить, и нажать кнопку "Изменить", после этого в поле для ввода информации произвести изменения и нажать кнопку "Сохранить" для сохранения изменений;
* Для отмены нежелаемых действий необходимо нажать кнопку "Отмена";
* Для выхода на главную кнопочную форму – кнопка "На главную".

Для выполнения, каких либо действий над машинами нужно нажать кнопку "Машины", после чего откроется окно, в котором с машинами можно выполнять следующие действия:

* Просмотр данных о машинах осуществляется путем нажатия кнопок "Вперед" и "Назад";
* Добавление машин. Для добавления машины необходимо нажать кнопку "Добавить", и ввести данные в поля ввода данных, после этого нужно нажать кнопку "Сохранить", для сохранения данных;
* Удаление машин. Для удаления машины необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно удалить, и нажать кнопку "Удалить";
* Изменение данных о машинах. Для изменения данных необходимо посредством кнопок "Вперед" и "Назад" выбрать запись, которую нужно изменить, и нажать кнопку "Изменить", после этого в поле для ввода информации произвести изменения и нажать кнопку "Сохранить" для сохранения изменений;
* Для отмены нежелаемых действий необходимо нажать кнопку "Отмена";
* Для выхода на главную кнопочную форму – кнопка "На главную".

Для оформления поездки нужно нажать кнопку "Новый заказ", после чего на экране появится окно оформления поездки. В открывшемся окне необходимо заполнить все поля для ввода информации. Передвижение между полями ввода информации осуществляется при помощи мыши.

* Для отмены нежелательных действий необходимо нажать кнопку "Отмена";
* Для выхода на главную кнопочную форму – кнопка "На главную".

Для корректировки поездок водителя нужно нажать кнопку "Корректировка поездок", после чего на экране появиться окно корректировка поездок. Для выбора водителя его необходимо выбрать из списка, либо воспользоваться поиском, для этого необходимо нажать кнопку "Найти".

* Изменение данных о количестве поездок водителей. Для изменения данных необходимо выбрать запись, которую нужно изменить, и нажать кнопку "Изменить", после этого в поле для ввода информации произвести изменения и нажать кнопку "Сохранить" для сохранения изменений;
* Обнуление количества поездок водителей. Для обнуления данных о количестве поездок необходимо выбрать запись, которую нужно изменить, и нажать кнопку "Обнулить", нажать кнопку "Сохранить" для сохранения изменений;
* Для отмены изменений нажать кнопку "Отмена";
* Для просмотра отчёта по корректировкам поездок нажать кнопку "Отчёт".

Для выхода из программы нужно нажать кнопку "Выход"

Для просмотра информации об осуществленных поездках необходимо нажать кнопку "Отчет по поездкам". Можно вывести отчёт за определённое число, либо вывести на печать информацию обо всех поездках. Для перехода на главную форму нажать кнопку "На главную".

2.2.4 Сообщения оператору

При работе с данной программой оператору могут выводиться различные сообщения:

1. При вводе ошибочного пароля. (Рис 10.)
2. При вводе новой информации, когда не все поля заполнены. (Рис 11.)
3. Запрос на подтверждение нажатия кнопки. Выводится при нажатии кнопки "Выход". (Рис 12.)

Рисунки иллюстрирующие ошибки представлены в Приложении 9.

Глава 3. Экономическая часть

Для организации бизнес - процессов современных предприятий в настоящее время информационные технологии приобретают всё более важное значение.Информационные технологии становятся главным фактором экономического роста информационные технологии - это совокупность методов, производственных процессов и программно-технических средств, объединены в технологическую цепочку, обеспечивающую сбор, обработку, вывод и распространение информации.

Информационные технологии включают технические и программные средства для хранения, обмена и распространения информации.

Современные достижения в развитии информационных технологий привели к формированию глобальной электронной среды для экономической деятельности, что открыло новые возможности в области организации бизнеса. Электронная среда для экономической деятельности приобрела название "Сетевая экономика". Для того, чтобы пользователи узнали о компании, её товарах и услугах рекламодателя, компания размещает на популярных и тематических сайтах или в рассылках своё рекламное обращение.

3.1 Краткое описание программного продукта и области его применения

Программа "Информационная система менеджера по работе с клиентами таксопарка "Семерочка" предназначена для автоматизации работы менеджера таксопарка. Данный программный продукт необходим для упрощения работы менеджера. С помощью данной программы уменьшиться время выполнения работы, повыситься производительность. Программа выполняет следующие функции: ввод исходных данных клиента при оформлении поездки; поиск необходимых клиентов по базе данных; постановка на внутрифирменный учет машин; снятие машин с внутрифирменного учета; изменение информации о машинах; изменение статуса машин и водителей; принятие на работу новых водителей; увольнение с работы водителей; принятие на работу новых операторов; увольнение с работы операторов; ввод и сохранение данных о маршруте поездки; изменение пароля доступа к программе; вывод информации по поездкам на печать; изменение и обнуление количества поездок каждого водителя; вывод информации количества поездок каждого водителя на печать; предоставление справки о работе программы; Программа имеет удобный интерфейс, не сложный при овладении навыками работы с данной программой.

3.2 Расчёт себестоимости программного продукта и выбор рекламного носителя

В таблице, приведённой ниже, представлена калькуляция статей издержек для фирмы, специализирующейся на разработку программных средств.

Табл. №1

|  |  |  |
| --- | --- | --- |
| № статьи | Наименование статьи | Сумма (руб.) |
| 1 | Материалы | 381 |
| 2 | Расходы на оплату труда | 3437,27 |
| 3 | Отчисления в единый социальный налог | 618,708 |
| 4 | Расходы на энергию | 103,60 |
| 5 | Амортизация | 501,73 |
| 6 | Расходы на отопление | 438,66 |
| 7 | Расходы на содержание зданий, инвентаря, АУП и прочие накладные расходы (данные расходы составляют 50-80 % от суммы расходов на оплату труда) | 1718,63 |
| 8 | Всего издержек | 7199,59 |
| 9 | Расходы на рекламу | 85 |
| 10 | Полные издержки (полная себестоимость) | 7424,69 |

При расчёте издержек следует предусмотреть продолжительность разработки программы (допустим в месяц или более)

3.2.1 Расчёт материальных издержек

Табл. №2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование | Количество | Стоимость (руб.) | Сумма (руб.) |
| 1 | Диск | 1 | 21 | 21 |
| 2 | Бумага | 300 | 1 | 300 |
| 3 | Краска для принтера | 2 | 30 | 60 |
|  | Итого: |  |  | 381 |

3.2.2 Расчёт издержек на зарплату

Табл. №3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория персонала | Оклад (руб.) | Зарплата за 83 часов | Премия от з/п (20%) | Итого (руб.) |
| Программист | 4500 | 2029,89 | 405,97 | 2435,86 |
| Лаборант | 1850 | 834,51 | 166,90 | 1001,41 |
| Всего зарплата: |  |  |  | 3437,27 |

3.2.3 Отчисления в единый социальный налог с ФОТ

Отчисления в единый социальный налог составляют 18 % с фонда оплаты труда.

3437,27\*18%/100%=618,708 руб.

3.2.4 Издержки на энергию

а) потребление энергии компьютером - квт/ч\*кол-во часов работы компьютера

Сэл.эн.ком=n\*Pком\*Сквт/ч, где

Сэл.эн.ком: затраты на электроэнергию, потребляемую компьютером

n - время, затраченное на разработку программы, составило 83 часов

Рком - мощность, потребляемая компьютером за 1 час.

При разработке программы составила 0,13 квт

Сквт/ч - стоимость 1 квт/ч, равна 1,71 руб.

Подставив значения в формулу, получим:

Сэл.эн.ком= 83\*0,13\*1,71 =18,45 руб.

б) освещение кабинета: квт/ч\*кол-во часов работы программиста за компьютером.

Сэл.эн.каб.=n\*b\*Pлам\*Сквт/ч, где

Сэл.эн.каб. - затраты на электроэнергию для освещения кабинета (руб.)

n - количество часов, необходимых для разработки одной программы = 83

b - количество ламп в кабинете= 15

Рлам - мощность, потребляемая 1 лампочкой за 1 час.

При разработке программы составила 0,04 квт

Сквт/ч - стоимость 1 квт/ч, равна 1,71 руб.

Подставив значения в формулу, получим:

Сэл.эн.каб.= 83\*15\*0,04 \*1,71 = 85,15 руб.

Всего потреблено энергии на разработку программы:

18,45+85,15=103,60 руб.

3.2.5 Амортизация основных средств

а) амортизация компьютера

месячная норма амортизации: А=годовая норма/12 мес.

Сумма аморт. комп. = А\*стоимость компьютера

Сам=Аком\*Ском/100%/12, где

Сам - амортизация компьютера, Аком - норма амортизации компьютера при условиях обучения (%) - 1/срок службы \*100%

Ском - стоимость компьютера (руб.) 30000 руб.

Сам= 30000\*16,66%/100%/12=416,66 руб.

б) амортизация компьютерного кабинета:

Стоимость основных фондов (стоимость здания)=1629000 руб.

Годовая норма амортизации =2%

месячная норма амортизации: А=годовая норма/12 мес.

Сумма аморт. здания в целом=А\*стоимость здания

Сумма аморт. здания в целом=2/100%/12\*1629000

Сумма аморт. компьютерного класса=(Сумма аморт. здания/площадь здания)\*площадь класса

(2715/1500)\*47=85,07 руб.

Итого амортизационных отчислений:

Сумма амортизации компьютера + Сумма амортизации класса

416,66+85,07=501,73 руб.

3.2.6 Издержки на отопление рассчитываются следующим образом

Рот=Рат.зд.\*Sкл/Sзд.

Где

Рот - издержки на отопление

Рат.зд. - расходы на отопление всего здания (14000)

Sкл - площадь компьютерного кабинета (47) кв.м

Sзд - площадь всего здания (1500) кв.м

Рот=14000\*47/1500=438,66 руб.

3.2.7 Прочие издержки

Прочие издержки принимаются в объёме 50-80% от суммы з/п (Ф)

3437,27\*50%/100%= 1718,63 руб.

3.2.8 Всего издержек по разработке

2.1+2.2+2.3+2.4+2.5+2.6+2.7=

381+3437,27+618,708+103,60+501,73+438,66+1718,63=7199,59 руб.

3.2.9 Расходы на рекламу

Для реализации рекламного обращения производитель может выбрать различные средства. например, можно разместить рекламу на популярных и тематических сайтах: баннер - прямоугольное графическое изображение, текстовый блок (используется как на web - сайтах, так и при рекламе в рассылках), байрики - небольшие всплывающие одновременно с загрузкой страницы окошках, мини - сайтах, коллаж. Можно разместить рекламу на телевидении, радио, интернете, районной и областной печатях и др. Студенту необходимо выбрать средство для размещения рекламы и рассчитать стоимость рекламы. При осуществлении рекламы через Internet следует выбрать, создавать ли свой сайт или вставить свою страницу в уже созданный сайт.

Второй способ значительно дешевле и эффективнее. Расходы на рекламу в Internet состоят из расходов на создание страницы на сайте и абонентской платы за Internet.

Реклама в Internet:

Количество дней рекламы: 5

Абонентская плата за Internet в месяц: 150

Минута работы в Internet: 1

Время создания страницы/баннера: 60

Срек1=Абонентская плата/30\*Количество дней рекламы+(минута работы в Internet\*время создания страницы)

Срек1=150/30\*5+(1\*60)=85 руб.

Общие затраты на рекламу составили:

Срек=Срек1

Срек=85

В области примерно 10 предприятий, которые могут воспользоваться этой программой. Рассчитаем затраты на тиражирование.

Она складывается из затрат на электроэнергию и затрат на оплату труда рабочего времени лаборанта.

Цена одного CD-R составляет 13 рублей. Затраты на диски составят: 130 рублей.

На тиражирование одного экземпляра программы необходимо 5 минут, следовательно, затраты на электроэнергию составят:

Сэл.эн.комп1=0,13\*(5/60\*1,71)=0,01 руб.

Сэл.эн.ком1=15\*0,04\*(5/60\*1,71)=0,08 руб.

Собщ=0,01+0,08= 0,10 руб.

Затраты на оплату труда лаборанта составили:

Сз/п=(1850/168)\*5/60=0,91 руб.

Затраты на тиражирование

Стир=(0,91+0,10)\*10=10,1 руб.

Переменные затраты составят:

Рпер=Стир+Сдис\*Количество дисков=10,1+13\*10=140,1 руб.

Итого, внепроизводственные затраты составят:

Рпер+Срек=140,1+85=225,1 руб.

3.2.10 Определяются полные издержки

Сумма статей 2.8 и 2.9 и заполняется таблица №1

Сумма=7199,59+225,1=7424,69 руб.

3.3 Ожидаемые результаты от внедрения (реализации) программного продукта

Предположим, что в результате размещения рекламы, наш программный продукт готовы приобрести 10 учебных заведений. Следует определить договорную цену и рассчитать прибыль от реализации 10 программ (допустим, программы не требуют сопровождения).

Цену можно определить по формуле:

Ц=И+Н+НДС, где

И - издержки производства и продажи (полная себестоимость),

Н - норма прибыли ( 13 % от издержек производства),

НДС - налог на добавленную стоимость ( 18 %),

Отпускная цена без НДС:

Ц=7424,69+(7424,69\*13/100%)=8389,89 руб.

Отпускная цена с учётом НДС:

Ц=8389,89+(8389,89\*18)/100%=9900,07 руб.

Таким образом, сумма от реализации 10 программ составит: 83898,9 руб., в том числе чистая прибыль:( 8389,89\*10)-225,1=83673,8 руб.

ГЛАВА 4. МЕРОПРИЯТИЯ ПО ОБЕСПЕЧЕНИЮ ТЕХНИКИ БЕЗОПАСНОСТИ И ПОЖАРНОЙ БЕЗОПАСНОСТИ

Опасные и вредные производственные факторы

Решение проблем, связанных с обеспечением здоровых и безопасных условий, в которых протекает труд человека - одна из наиболее важных задач в разработке новых технологий и систем производства. Работа операторов, программистов и просто пользователей непосредственно связана с компьютерами, а соответственно с дополнительными вредными воздействиями целой группы факторов, что существенно снижает производительность их труда и отрицательно влияет на здоровье.

На оператора видеодисплейных терминалов (ВДТ) могут воздействовать опасные и вредные производственные факторы:

* повышенный уровень напряжения в электрических цепях питания и управления ВДТ, который может привести к электротравме оператора;
* повышенный уровень напряжённость электрического и магнитных полей в широком диапазоне частот (в том числе от токов промышленной частоты 50 Гц от ВДТ, вспомогательных приборов, силовых кабелей, осветительных установок и т.д.);
* несоответствующие санаторным нормам визуальные параметры дисплеев (нестабильность изображения, повешенный уровень прямой и отраженной блесткости, повышенная яркость освещенного изображения, повышенный уровень пульсации светового потока и д.р.);
* повышенный или пониженный уровень освещенности;
* повышенный уровень статического электричества;
* повышенный уровень загазованности (по углекислому газу и аммиаку, которые образуются при выдыхании);
* повышенный уровень загазованности воздуха рабочей зоны от внешних источников;
* нарушение норм по аэроионному составу воздуха;
* повышенный уровень шума от работающих вентилятора охлаждения ВДТ и принтера;
* повышенные зрительные нагрузки и адинамия глазных мышц, т.е. их малая подвижность при высоком статическом зрительном напряжении;
* монотонность труда;
* повышенное умственное напряжение из-за большого объема перерабатываемой и усваиваемой информации;
* физическое перенапряжение из-за нерациональной организации рабочего места (неудобные кресла и т.д.);
* несоответствующие нормам параметры микроклимата;

Для обеспечения высокого уровня безопасности и благоприятного условия труда используются технические, организационные, правовые, экономические методы и средства.

Способы и методы защиты от опасных и вредных производственных факторов

К работам на ВДТ допускаются лица не моложе 18 лет, прошедшие обязательные при приеме на работу и ежегодный медицинский осмотр и признанные годными по состоянию здоровья к работе.

К работам на ВДТ не допускаются женщины во время беременности и в период кормления ребёнка грудью.

До начала работы каждый оператор должен пройти специальную подготовку по охране труда в которую входят:

* вводный инструктаж;
* инструктаж по охране труда на конкретном рабочем месте;
* обучение работе на ВДТ с использованием конкретного программного обеспечения с обучением при этом безопасным приёмам и методам труда;
* проверка знаний, в том числе по электробезопасности, с присвоением I квалификационной группы;
* обучение приёмам и методам оказания первой помощи пострадавшим от несчастных случаев.

Повторное обучение и периодическую проверку знаний норм охраны труда оператор должен проходить не реже 1 раза в 3 года.

Для защиты от опасности поражения электрическим током металлические корпуса электропотребляющих установок должны быть заземлены (занулены). Категорически запрещается использовать в качестве контура заземления паропроводные, водопроводные, газовые, отопительные и другие трубы, радиаторы.

Для защиты от электромагнитного излучения (ЭМИ), которое идет от задней и боковых стенок монитора, рекомендуется размещать рабочие места относительно друг друга в соответствии с рис 1.

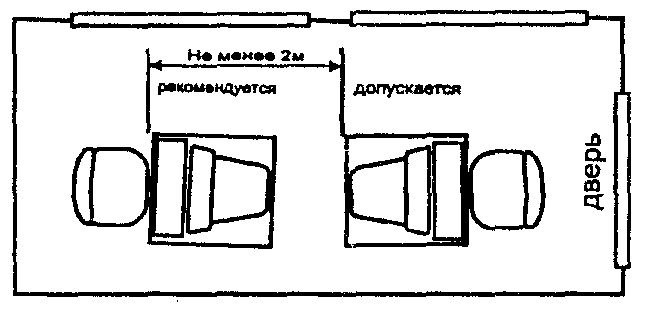


Рисунок 1. Рекомендации по размещению рабочих мест относительно друг друга, окон, стен помещения и приборов

Для обеспечения надежного считывания информации при соответствующей степени комфортности ее восприятия должны быть определены оптимальные и допустимые диапазоны визуальных параметров, которые внесены в техническую документацию на ВДТ.

Помещение, в котором расположены ВДТ, должно быть оборудовано системами естественного и искусственного освещения. Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север или северо-восток и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1,2 %. Оконные проемы должны быть оборудованы регулируемыми устройствами типа жалюзи, занавесей, внешних козырьков и др. Расположение рабочих мест по отношения к световым проемам приведены на рис 2, 3.

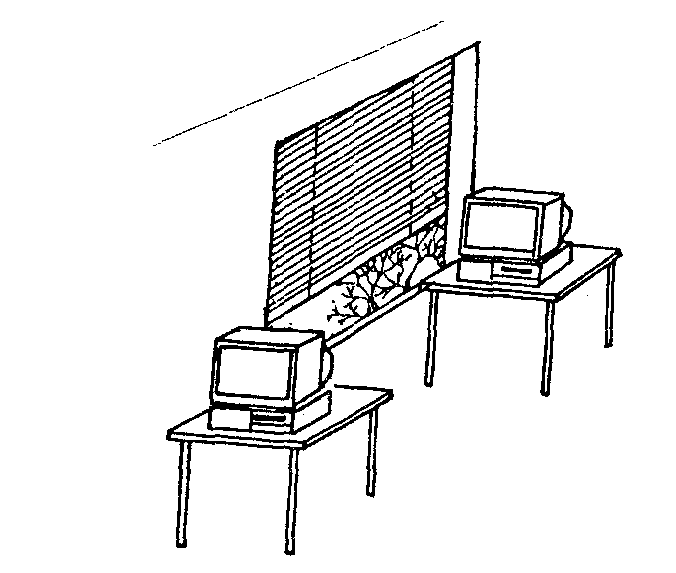


Рисунок 2. Расположение рабочих мест

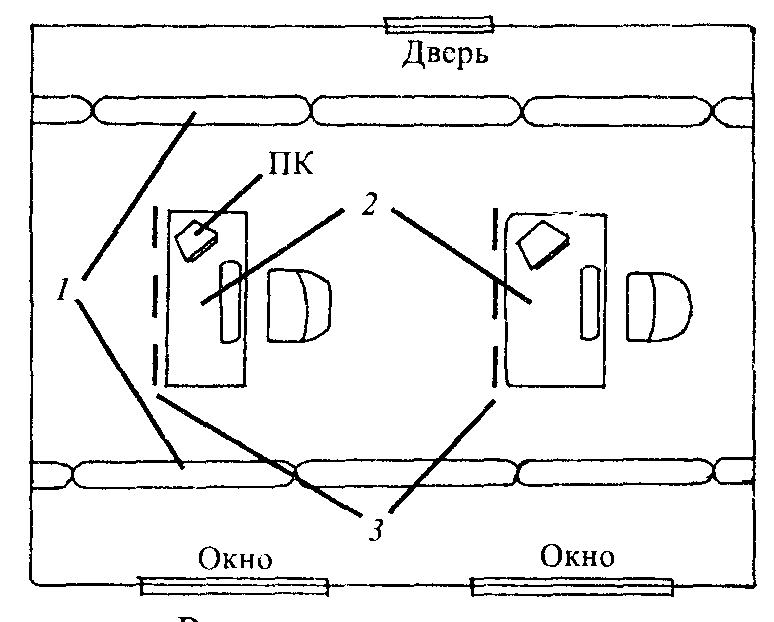


Рисунок 3. Рекомендуемое размещение светильников относительно рабочих мест (1 - светильники 2 - стол с ПК: 3 -защитная перегородка (экран) при повышенном излучении от задней части дисплея)

Искусственное освещение должно осуществляться системой общего освещения. Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов.

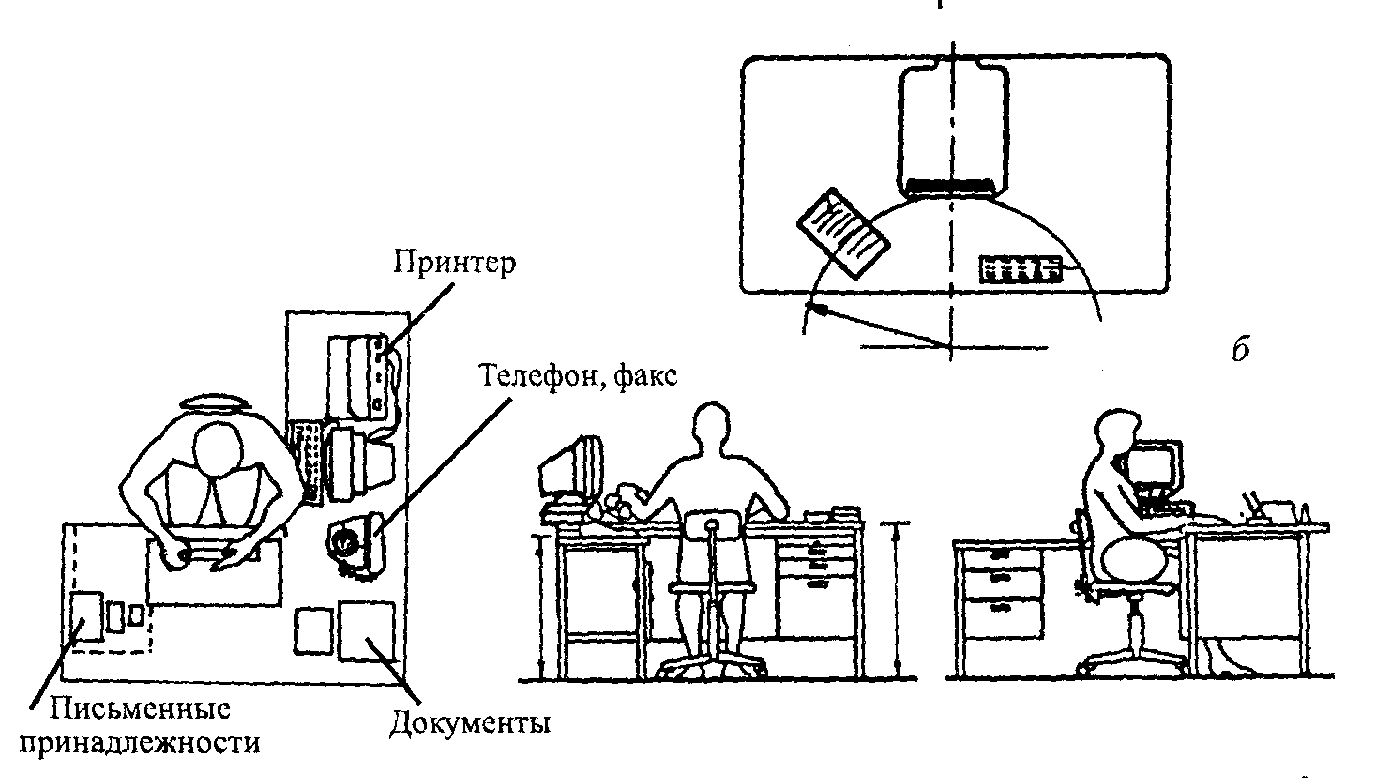
Для уменьшения электростатических зарядов, образующихся на кистях рук при трении о клавиатуру и "мышь" рекомендуется регулярно смачивать ладони (3-5 раз в смену). Поверхность пола должна обладать антистатическими свойствами.

Для снижения уровня шума в помещениях с ВДТ используют звукопоглощающие материалы с максимальным коэффициентом звукопоглощения в области частот 63-8000 Гц для отделки помещений. Дополнительным звукопоглощением служат однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии 15-20 см. от ограждения. Ширина занавеси должна быть в 2 раза больше ширины окна.

Воздух помещения, где находится много людей и ВДТ насыщается тяжелыми и лёгкими аэроионами пыли, что приводит к ухудшению здоровья. Для поддержки аэроионного состава воздуха применяются различные аппарата аэроионопрофилактики и обеспыливания: аппараты серии "Элион", "СА-1 Москва", "Супер-плюс" и др. Для снижения нервного напряжения, утомления зрительной и опорно-двигательной системы оператора рекомендуется следующий режим работы. Не рекомендуется работать на ВДТ более 2 часов подряд без перерыва. Во время перерывов в работе для восстановления работоспособности рекомендуется делать различные упражнения. Для уменьшения отрицательного влияния монотонности рекомендуется менять вид работы, например, чередовать ввод данных и редактирование, считывание информации и ее осмысление.

Организации рабочего места

Производительность и безопасность труда операторов ВДТ зависит от правильной организации рабочего места. Рабочее место организуют так, чтобы использовать рациональные приемы работы и эксплуатации машины при наименьшем числе движений оператора и удобном обращении с обрабатываемым материалом.



Рекомендации по рациональной организации рабочего места приведены на рис 4.

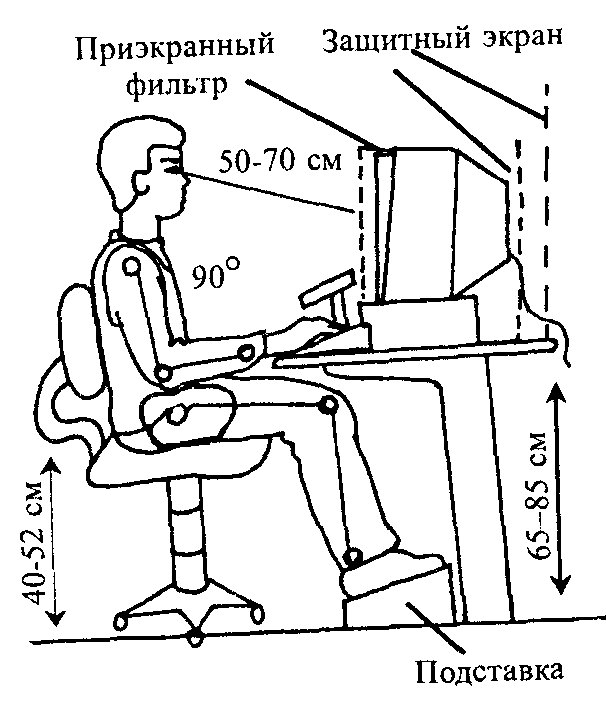


Рисунок 4. Рекомендации по рациональной организации рабочего места, расположению оргтехники и приспособлений на рабочем месте оператора

Конструкция рабочего‚ стола должна обеспечивать оптимальное размещение, оборудования и оргтехники с учетом их конструктивных особенностей (размер ПК, клавиатуры, пюпитра др.) и характера выполняемой работы. Дисплей в зависимости от размеров алфавитно-цифровых знаков должен находиться от глаз пользователя на оптимальном расстоянии 60—70 см, но не ближе 50 см.

Рекомендуемое расположение оргтехники и приспособлений на рабочем месте оператора:

а — при периодической работе на ПК;

б — при постоянной работе на ПК

(свыше 4 часов за рабочий день)

Площадь на одно рабочее место с ВДТ должна составлять не менее 6 м2 а объем – не менее 20,0м3 .

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования, характера выполняемой работы.

Рабочий стул (кресло) должно быть подъемно-поворотным и регулируемым по высоте и углами наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья. Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, не электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Рабочее место должно быть оснащено легко перемещаемым пюпитром для документов.

Климатические условия на рабочем месте оператора должны соответствовать следующим санитарно-гигиеническим нормам:

* температура воздуха – то 220 до 240 С (в холодный период года) от 230 до 250 С (в теплый период года);
* относительная влажность –40-60% но не более 75%;
* скорость движения воздуха – не более 0,1 м/с.

Для поддержания заданных климатических условий помещения, в котором расположены ВДТ, должны быть оборудованы системой вентиляции, кондиционирования и отопления.

Пожарная безопасность

При эксплуатации ВДТ возможны аварийные ситуации: короткое замыкание, перегрузки, повышение переходных сопротивлений в электрических контактах, перенапряжение и др. При возникновении аварийных ситуаций происходит разное выделение тепловой энергии, которая может явиться причиной возникновения пожара.

Помещения, в которых располагаются ВДТ, относятся к категории пожароопасности "В" (характеризуется наличием воспламеняющихся пластмасс, лакокрасочных покрытий). Исходя из норм пожарной безопасности для помещений площадью до100м2 , в которых эксплуатируется ВДТ, в качестве первичного средства пожаротушения требуется один углекислотный огнетушитель типа 04-5 или порошковый огнетушитель типа ОП-5. Рекомендуется оснащать помещения дымовыми противопожарными извещателями.

Для безопасности эвакуации персонала рядом с дверными проемами, выключателями, рубильниками следует размещать фотолюминесцентные эвакуационные знаки.

Расчет потребного воздухообмена при общеобменной вентиляции

1. Общие сведения

Вентиляция - организованный и регулируемый воздухообмен обеспечивающий удаление из помещении воздуха загрязненного вредными примесями (газами, парами, пылью), и подачу в него свежего воздуха.

По способу подачи в помещение свежего воздуха и удалению загрязненного системы вентиляции подразделяют на естественную, механическую и смешанную. По назначению вентиляция может быть общеобменной и местной.

2. Методика расчета

При общеобменной вентиляции потребный воздухообмен определяют из условия удаления избыточной теплоты и разбавления вредных выделений свежим воздухом до допустимых концентраций. Предельно допустимые концентрации вредных веществ в воздухе рабочей зоны устанавливают по ГОСТ 12.1.005—88.

2.1. Расход приточного воздуха, М3/Ч, необходимый для отвода избыточной теплоты,



где Qизб - избыточное количество теплоты, кДж/ч; с - теплоёмкость воздуха, Дж/(кг•К); с = 1,2 кДж/(кг•К); р - плотность воздуха, кг/м3; tуд - температура воздуха, удаляемого из помещения принимается равной температуре воздуха в рабочей зоне, °С; tпр - температура приточного воздуха, °С.

Расчетное значение температуры приточного воздуха зависит от географического расположения предприятия; для Москвы ее Принимают равной 22,3 °С.

Температуру воздуха в рабочей зоне принимают на 3.. .5 °С выше расчетной температуры наружного воздуха, Плотность воздуха, кг/м3, поступающего в помещение,



Избыточное количество теплоты, подлежащей удалению из производственного помещения, определяют по тепловому балансу:



где - теплота, поступающая в помещение от различных источников кДж/ч; - теплота, расходуемая (теряемая) стенами здания и уходящая с нагретыми материалами, кДж/ч.



К основным источникам тепловыделений в производственных помещениях относятся:

оборудование с приводом от электродвигателей; солнечная радиация;

персонал, работающий в помещении;

Поскольку перепад температур воздуха внутри и снаружи здания в теплый период года незначительный (3... 5°С), то при расчете воздухообмена по избытку тепловыделений потери теплоты через конструкции зданий можно не учитывать. При этом некоторое увеличение воздухообмена благоприятно влияет на условия труда работающих в наиболее жаркие дни теплого периода года.

С учетом изложенного формула принимает следующий вид:



В настоящем расчетном задании избыточное количество теплоты определяется только с учетом тепловыделений электрооборудования и работающего персонала:



где Qэ.о - теплота, выделяемая при работе электродвигателей оборудования, кДж/ч; Qр - теплота, выделяемая работающим персоналом, кДж/ч.

Теплота, выделяемая электродвигателями оборудования,

Qэ.о = 3528βN

Qэ.о = 3528\*0.27\*0.13=123.833

где β - коэффициент, учитывающий загрузку оборудования, одновременность его работы, режим работы; β = 0,25...0,35; N - общая установочная мощность электродвигателей, кВт.

Теплота, выделяемая работающим персоналом,

Qp = nKp

Qp = 5\*400=2000

где n — число работающих, чел.; Кp — теплота, выделяемая одним человеком, кДж/ч (принимается равной при легкой работе 300 кДж/ч; при работе средней тяжести 400 кДж/ч; при тяжелой работе 500 кДж/ч).

Кратность воздухообмена, 1/ч,



где L-потребный воздухообмен, м3/ч; Vc-внутренний свободный объём помещения, м3

Кратность воздухообмена помещений обычно составляет от 1 до 10.

ЗАКЛЮЧЕНИЕ

В результате проведенной работы, были изучены основополагающие документы, позволяющие понять специфику выбранной предметной области (см. список использованной литературы), разработана схема потоков данных и проведено их описание. На основании подготовленной схемы потоков данных разработана концептуальная модель базы данных сущностей и датологическая модель базы данных. Проведена нормализация таблиц базы данных. С использованием возможностей по созданию проверочной схемы связей таблиц СУБД Access создана нормализованная схема данных.

Произведен выбор системы программирования и разработаны окна пользовательского интерфейса и отчетов, основные не визуальные компоненты представлены в виде спецификаций. Наиболее загруженные формы имеют пояснительные выноски. Выполнена блок-схема загрузки форм программы с описанием основных функциональных кнопок.

В результате проведенной работы была создана информационная система менеджера по работе с клиентами таксопарка "Семерочка", имеющая основные функциональные возможности реальной системы. Данная программа может быть использована для дальнейшего усовершенствования и доработки.

Требуемое техническое задание выполнено в полном объёме.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. В.В.Годин, И.К.Корнеев "Информационное обеспечение управленческой деятельности". Мск. Изд. "Высшая школа", 2001г;
2. Форма № 44 (накладная);
3. А.Я. Архангельский "Программирование в Delphi 7" – М, ООО "Бином-Пресс", 2005г. – 1152л.
4. Электронный учебник "Delphi v7.0 Book";
5. Лекции по предмету "Разработка и эксплуатация автоматизированных информационных систем";
6. Лекции по предмету "Базы данных";
7. ГОСТы 19.404-79, 19.402-78, 10.505-79;
8. Гофман В.Э., Хомоменко А.Д. "Delphi 6" – СПб., БХВ-Петербург, 2002. – 1152л.
9. Электронный учебник "Delphi v6.0 Book";

# ПРИЛОЖЕНИЯ

Приложение 1



Рисунок 1. Структурная схема организации

Приложение 2



Рисунок 2. Схема взаимосвязи программных модулей и информационных хранилищ.

Приложение 3



Запрос на предоставление машины

1. Запрос от клиента менеджеру по работе с клиентами на предоставление машины;
2. Запрос менеджера по работе с клиентами на получение личных данных клиента;
3. Личные данные клиента;
4. Запрос менеджером по работе с клиентами системе регистрации клиентов на регистрацию клиента;
5. Запрос на регистрацию личных данных клиента системой регистрации клиентов в таблицу "Клиенты";
6. Информация о регистрации личных данных клиента в таблице "Клиенты" системой регистрации клиентов;
7. Информация о регистрации клиента на экране монитора.

Запрос на осуществление перевозки

1. Запрос менеджера по работе с клиентами клиенту на получение информации о маршруте;
2. Информация о маршруте;
3. Запрос системе формирования поездок на регистрацию информации о маршруте поездки;
4. Запрос системой формирования поездок в таблицу "Клиенты" на получение данных о клиенте;
5. Запрос системой формирования поездок в таблицу "Водители" на получение данных о водителе;
6. Запрос системой формирования поездок в таблицу "Машины" на получение данных о машине;
7. Информация о получении данных о клиенте системой формирования поездок;
8. Информация о получении данных о водителе системой формирования поездок;
9. Информация о получении данных о машине системой формирования поездок;
10. Запрос системой формирования поездок на сохранение данных о поездке в таблице "Поездки"
11. Информация о регистрации данных о поездке системой формирования поездок;
12. Информация менеджеру по работе с клиентами о сохранении данных о поездке.

Запрос на получение информации о осуществленных поездках

1. Запрос менеджеру по работе с клиентами от водителя на получение информации о осуществленных поездках;
2. Запрос менеджера по работе с клиентами в систему формирования отчетов на получение информации о осуществленных поездках;
3. Запрос системой формирования отчетов на получение информации о осуществленных поездках в таблицу "Поездки";
4. Информация о осуществленных поездках из таблицы "Поездки" в систему формирования отчетов;
5. Информация менеджеру по работе с клиентами об осуществленных поездках из системы формирования отчетов;
6. Информация об осуществленных поездках водителю в виде отчета.

Запрос на добавление(списание) машины

1. Запрос начальником гаража менеджеру по работе с клиентами на добавление (списание) машины;
2. Запрос системе администрирования на добавление (списание) машины
3. Запрос системой администрирования в таблицу "Машины" на добавление (списание) машины;
4. Информация о добавлении (списание) машины в таблице "Машины" системой администрирования;
5. Информация менеджеру по работе с клиентами из системы администрирования о добавлении (списание) машины;
6. Информация начальнику гаража о добавлении (списание) машины.

Принятие(увольнение) водителей

1. Запрос работником отдела кадров на принятие (увольнение) водителя;
2. Запрос системе администрирования на принятие (увольнение) водителя;
3. Запрос системой администрирования в таблицу "Водители" на сохранение (удаление) данных о водителе;
4. Информация о сохранении (удалении) данных о водителе системой администрирования;
5. Информация менеджеру по работе с клиентами из системы администрирования о принятии (увольнении) водителя;
6. Информация работнику кадров о принятии (увольнении) водителя;

Принятие(увольнение) операторов

1. Запрос работником отдела кадров на принятие (увольнение) оператора;
2. Запрос системе администрирования на принятие (увольнение) оператора;
3. Запрос системой администрирования в таблицу "Операторы" на сохранение (удаление) данных об операторе;
4. Информация о сохранении (удалении) данных об операторе системой администрирования;
5. Информация менеджеру по работе с клиентами из системы администрирования о принятии (увольнении) оператора;
6. Информация работнику кадров о принятии (увольнении) оператора;

Корректировка поездок

1. Запрос менеджера по работе с клиентами в систему администрирования на корректировку данных о поездках осуществлённых водителем;
2. Запрос системой администрирования в таблицу "Водители" на корректировку данных о поездках осуществлённых водителем;
3. Информация из таблицы "Водители" в систему администрирования о корректировке данных о поездках, осуществлённых водителем;
4. Информация менеджеру по работе с клиентами из системы администрирования о корректировке данных о поездках, осуществлённых водителем;

Запрос на получение информации о поездках осуществлённых водителем

1. Запрос менеджеру по работе с клиентами от водителя на получение информации о осуществленных им поездках;
2. Запрос менеджера по работе с клиентами в систему формирования отчетов на получение информации о поездках осуществлённых водителем;
3. Запрос системой формирования отчетов на получение информации о поездках осуществлённых водителем в таблицу "Водители";
4. Информация о поездках осуществлённых водителем из таблицы "Водители" в систему формирования отчетов;
5. Информация менеджеру по работе с клиентами об поездках осуществлённых водителем из системы формирования отчетов;
6. Информация об поездках осуществлённых водителем от менеджера по работе с клиентами водителю в виде отчета.

Приложение 4

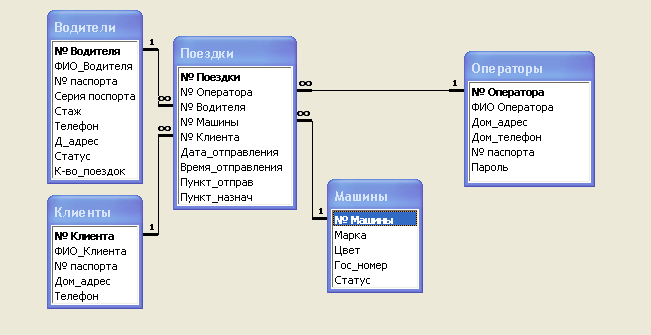


Рисунок 3. Инфологическая модель

Приложение 5

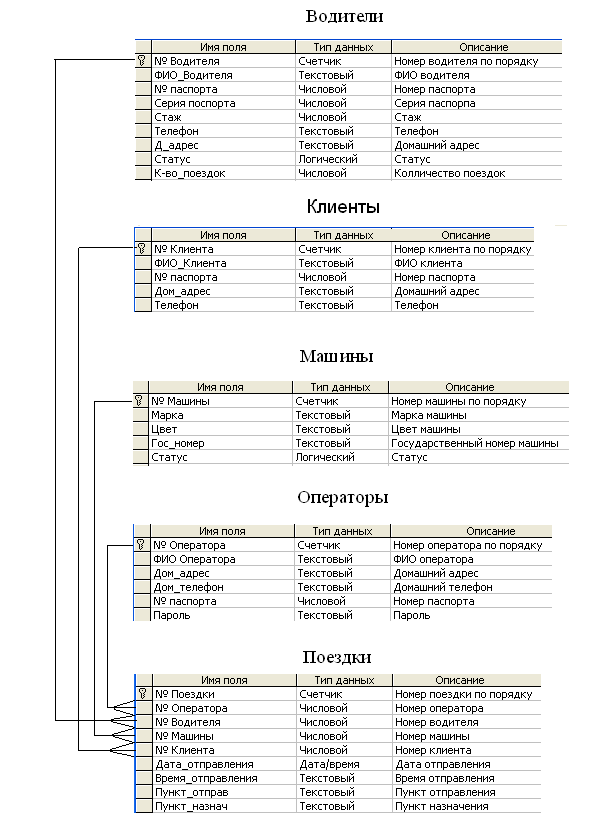


Рисунок 4. Схема датологической модели

Приложение 6



Рисунок 5. Схема загрузки формы проверки прав доступа



Рисунок 6. Выбор пункта подменю "Далее"



Рисунок 7. Выбор пункта подменю "Водители"



Рисунок 8. Выбор пункта подменю "На главную"

Приложение 8



Рисунок 9. Блок-схема.

Приложение 9

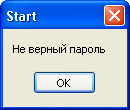


Рисунок 10. Сообщение оператору "О неверном пароле"

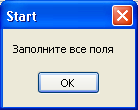


Рисунок 11. Сообщение оператору "О незаполненных полях"

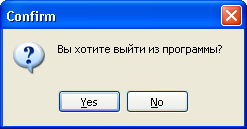


Рисунок 12. Сообщение оператору "О выходе из программы"

Приложение 10

unit Glavnaya

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Buttons, jpeg, ExtCtrls, StdCtrls, DB, ADODB, Grids, DBGrids,

XPMan,shellapi, ComCtrls;

type

TGlavnaya\_f = class(TForm)

SpeedButton1: TSpeedButton;

SpeedButton4: TSpeedButton;

GroupBox1: TGroupBox;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton7: TSpeedButton;

SpeedButton8: TSpeedButton;

SpeedButton9: TSpeedButton;

SpeedButton10: TSpeedButton;

procedure SpeedButton1Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton6Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure SpeedButton8Click(Sender: TObject);

procedure SpeedButton9Click(Sender: TObject);

procedure SpeedButton10Click(Sender: TObject);

procedure SpeedButton7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Glavnaya\_f: TGlavnaya\_f;

implementation

uses Voditeli, Parol\_red, Mashini , Operator, Oformlenie\_poezdki,

Otchet, Status, Vybor\_otcheta, Korrektirovka;

{$R \*.dfm}

procedure TGlavnaya\_f.SpeedButton1Click(Sender: TObject);

begin

// вывод сообщения

if MessageDlg('Вы хотите выйти из программы?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then

Application.Terminate; //закрытие программы

end;

procedure TGlavnaya\_f.SpeedButton5Click(Sender: TObject);

begin

Voditeli\_f.Show;//открытие формы водители

Glavnaya\_f.Hide;//скрытие формы главная

end;

procedure TGlavnaya\_f.SpeedButton6Click(Sender: TObject);

begin

Mashini\_f.Show; //открытие формы машины

Glavnaya\_f.Hide;//скрытие формы главная

end;

procedure TGlavnaya\_f.SpeedButton4Click(Sender: TObject);

begin

Glavnaya\_f.Hide;

Oformlenie\_poezdki\_f.show;

end;

procedure TGlavnaya\_f.SpeedButton8Click(Sender: TObject);

begin

Vybor\_otcheta\_.Show;

Glavnaya\_f.Hide;

end;

procedure TGlavnaya\_f.SpeedButton9Click(Sender: TObject);

begin

Glavnaya\_f.Hide;

Status\_f.show;

end;

procedure TGlavnaya\_f.SpeedButton10Click(Sender: TObject);

begin

Korrektirovka\_.Show;

Glavnaya\_f.Hide;

end;

procedure TGlavnaya\_f.SpeedButton7Click(Sender: TObject);

begin

Operatory\_f.Show;//открытие формы водители

Glavnaya\_f.Hide;//скрытие формы главная

end;

end.

unit Korrektirovka

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, StdCtrls, Mask, DBCtrls;

type

TKorrektirovka\_ = class(TForm)

DBGrid1: TDBGrid;

Label1: TLabel;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

Label2: TLabel;

Label3: TLabel;

GroupBox1: TGroupBox;

Button6: TButton;

Edit1: TEdit;

Label10: TLabel;

CheckBox2: TCheckBox;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button7: TButton;

procedure Button6Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Korrektirovka\_: TKorrektirovka\_;

implementation

uses DM, Glavnaya, Otchet\_poezdki, Otchet\_vse\_voditeli;

{$R \*.dfm}

procedure TKorrektirovka\_.Button6Click(Sender: TObject);

begin

if

CheckBox2.Checked then

DM\_f.ADOTable2.Filtered:=false

else

begin

DM\_f.ADOTable2.Filtered:=false;

DM\_f.ADOTable2.Filter:='ФИО\_Водителя='''

+ (Edit1.Text)+'''';

DM\_f.ADOTable2.Filtered:=true;

end;

end;

procedure TKorrektirovka\_.Button5Click(Sender: TObject);

begin

Glavnaya\_f.Show;

Korrektirovka\_.Hide;

end;

procedure TKorrektirovka\_.Button1Click(Sender: TObject);

begin

DBEdit2.ReadOnly:=false;

Button1.Enabled:=false;

Button2.Enabled:=false;

Button5.Enabled:=false;

Button7.Enabled:=false;

Button3.Enabled:=true;

Button4.Enabled:=true;

DM\_f.ADOTable2.Edit;

DBEdit2.SetFocus;

end;

procedure TKorrektirovka\_.Button2Click(Sender: TObject);

var

a:integer;

begin

a:=0;

DBEdit2.ReadOnly:=false;

DM\_f.ADOTable2.Edit;

DBEdit2.Text:=IntToStr(a);

DM\_f.ADOTable2.Post;

end;

procedure TKorrektirovka\_.Button3Click(Sender: TObject);

begin

if DBEdit2.Text='' then

begin

ShowMessage('Введите колличество поездок');

DBEdit2.SetFocus;

end

else

begin

DBEdit2.ReadOnly:=true;

Button1.Enabled:=true;

Button2.Enabled:=true;

Button5.Enabled:=true;

Button7.Enabled:=true;

Button3.Enabled:=false;

Button4.Enabled:=false;

DM\_f.ADOTable2.Post;

ShowMessage('Данные сохранены');

end;

end;

procedure TKorrektirovka\_.Button4Click(Sender: TObject);

begin

DBEdit2.ReadOnly:=true;

Button1.Enabled:=true;

Button2.Enabled:=true;

Button5.Enabled:=true;

Button7.Enabled:=true;

Button3.Enabled:=false;

Button4.Enabled:=false;

DM\_f.ADOTable2.Cancel;

end;

procedure TKorrektirovka\_.Button7Click(Sender: TObject);

begin

if CheckBox2.Checked then

Otchet\_vse\_voditeli\_.QuickRep1.Preview

else

Otchet\_poezdki\_.QuickRep1.Preview;

end;

end.

unit Mashini

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls, DB, ADODB, Grids,

DBGrids;

type

TMashini\_f = class(TForm)

DBGrid1: TDBGrid;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

SpeedButton7: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton3: TSpeedButton;

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton1: TSpeedButton;

SpeedButton8: TSpeedButton;

procedure SpeedButton7Click(Sender: TObject);

procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit2KeyPress(Sender: TObject; var Key: Char);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

procedure SpeedButton3Click(Sender: TObject);

procedure SpeedButton8Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton6Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Mashini\_f: TMashini\_f;

implementation

uses Glavnaya, DM;

{$R \*.dfm}

procedure TMashini\_f.SpeedButton7Click(Sender: TObject);

begin

Mashini\_f.Hide;//скрытие формы машины

Glavnaya\_f.Show;//открытие формы главная

end;

procedure TMashini\_f.DBEdit1KeyPress(Sender: TObject; var Key: Char);

begin

if key=#13 then //нажатие Enter

DBEdit2.SetFocus;//Переход на следующую строку при нажатии Enter

end;

procedure TMashini\_f.DBEdit2KeyPress(Sender: TObject; var Key: Char);

begin

if key=#13 then //нажатие Enter

DBEdit3.SetFocus;//Переход на следующую строку при нажатии Enter

end;

procedure TMashini\_f.SpeedButton2Click(Sender: TObject);

begin

DM\_f.ADOTable3.Prior;

end;

procedure TMashini\_f.SpeedButton1Click(Sender: TObject);

begin

DM\_f.ADOTable3.Next;

end;

procedure TMashini\_f.SpeedButton3Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DBEdit1.SetFocus;

DM\_f.ADOTable3.Insert;

end;

procedure TMashini\_f.SpeedButton8Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DM\_f.ADOTable3.Edit;

DBEdit1.SetFocus;

end;

procedure TMashini\_f.SpeedButton4Click(Sender: TObject);

begin

DM\_f.ADOTable3.Delete;

end;

procedure TMashini\_f.SpeedButton5Click(Sender: TObject);

begin

if (DBEdit1.Text='') or (DBEdit2.Text='') or

(DBEdit3.Text='')

then

ShowMessage('Заполните все поля')

else begin

DM\_f.ADOTable3.Post;

SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

end;

end;

procedure TMashini\_f.SpeedButton6Click(Sender: TObject);

begin

SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

DM\_f.ADOTable3.Cancel;

end;

procedure TMashini\_f.FormShow(Sender: TObject);

begin

DM\_f.ADOTable3.Filtered:=false;

end;

end.

unit Oformlenie\_poezdki

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids, DBGrids, Mask, DBCtrls, ExtCtrls, Buttons,

ComCtrls;

type

TOformlenie\_poezdki\_f = class(TForm)

GroupBox1: TGroupBox;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

DBEdit2: TDBEdit;

DBEdit1: TDBEdit;

DBEdit4: TDBEdit;

DBEdit3: TDBEdit;

DBGrid1: TDBGrid;

GroupBox2: TGroupBox;

GroupBox3: TGroupBox;

DBEdit16: TDBEdit;

DBEdit15: TDBEdit;

SpeedButton11: TSpeedButton;

SpeedButton3: TSpeedButton;

GroupBox5: TGroupBox;

DBEdit5: TDBEdit;

Label7: TLabel;

DBEdit7: TDBEdit;

Label6: TLabel;

DBEdit6: TDBEdit;

Label1: TLabel;

SpeedButton7: TSpeedButton;

SpeedButton1: TSpeedButton;

DBEdit9: TDBEdit;

Button1: TButton;

Button2: TButton;

Button3: TButton;

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

GroupBox4: TGroupBox;

SpeedButton10: TSpeedButton;

SpeedButton2: TSpeedButton;

DBEdit10: TDBEdit;

DBEdit8: TDBEdit;

Label8: TLabel;

Label9: TLabel;

Button4: TButton;

Button5: TButton;

GroupBox6: TGroupBox;

Button6: TButton;

Edit1: TEdit;

CheckBox2: TCheckBox;

Label10: TLabel;

DBEdit11: TDBEdit;

DateTimePicker1: TDateTimePicker;

Button7: TButton;

procedure SpeedButton7Click(Sender: TObject);

procedure SpeedButton10Click(Sender: TObject);

procedure SpeedButton11Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton3Click(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure GroupBox5MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure Button7Click(Sender: TObject);

procedure GroupBox1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure GroupBox2MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure GroupBox6MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure Button6MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure Label10MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Oformlenie\_poezdki\_f: TOformlenie\_poezdki\_f;

implementation

uses DM, Glavnaya;

{$R \*.dfm}

procedure TOformlenie\_poezdki\_f.SpeedButton7Click(Sender: TObject);

begin

if

(DBEdit5.Text='') or (DBEdit6.Text='') or (DBEdit7.Text='')

then ShowMessage('Заполните все поля')

else

begin

DM\_f.ADOTable6.FieldByName('№ Оператора').AsInteger:=1;

DM\_f.ADOTable6.FieldByName('№ Водителя').AsString:=DBEdit10.Text;

DM\_f.ADOTable6.FieldByName('№ Машины').AsString:=DBEdit16.Text;

DM\_f.ADOTable6.FieldByName('№ Клиента').AsString:=DM\_f.ADOTable5.FieldByName('№ Клиента').AsString;

DM\_f.ADOTable6.FieldByName('Дата\_отправления').AsString:=DBEdit9.Text;

DM\_f.ADOTable6.Post;

ShowMessage('Данные сохранены!');//вывод сообщения

DM\_f.ADOTable2.Edit;

DM\_f.ADOTable2.FieldByName('К-во\_поездок').AsInteger:=StrToInt(DBEdit11.Text)+1;

DM\_f.ADOTable2.Post;

DM\_f.ADOTable3.Edit; //Изменение статуса выбранной машины

DM\_f.ADOTable3.FieldByName('Статус').AsString:='False'; // Изменение статуса машины

DM\_f.ADOTable3.Post; // Сохранение данных в таблице машины

DM\_f.ADOTable2.Edit; //Изменение статуса выбранного водителя

DM\_f.ADOTable2.FieldByName('Статус').AsString:='False'; // Изменение статуса водителя

DM\_f.ADOTable2.Post; // Сохранение данных в таблице водители

SpeedButton1.Enabled:=true;

Button4.Enabled:=true;

Button5.Enabled:=false;

SpeedButton7.Enabled:=false;

end;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton10Click(Sender: TObject);

begin

DM\_f.ADOTable2.Prior;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton11Click(Sender: TObject);

begin

DM\_f.ADOTable3.Prior;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton2Click(Sender: TObject);

begin

DM\_f.ADOTable2.Next;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton3Click(Sender: TObject);

begin

DM\_f.ADOTable3.Next;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton1Click(Sender: TObject);

begin

Glavnaya\_f.Show;

Oformlenie\_poezdki\_f.Hide;

end;

procedure TOformlenie\_poezdki\_f.FormShow(Sender: TObject);

begin

DBEdit9.Text:=DateToStr(date);

DM\_f.ADOTable3.Filtered:=False; //Деактивация фильтра

DM\_f.ADOTable3.Filter:='Статус=True'; //Задание фильтра

DM\_f.ADOTable3.Filtered:=True; //Активация фильтра

DM\_f.ADOTable2.Filtered:=False; //Деактивация фильтра

DM\_f.ADOTable2.Filter:='Статус=True' ;//Задание фильтра;

DM\_f.ADOTable2.Filtered:=True;//Деактивация фильтра

end;

procedure TOformlenie\_poezdki\_f.Button3Click(Sender: TObject);

begin

DBEdit1.ReadOnly:=true;

DBEdit2.ReadOnly:=true;

DBEdit3.ReadOnly:=true;

DBEdit4.ReadOnly:=true;

Button1.Enabled:=true;

Button2.Enabled:=false;

Button3.Enabled:=false;

SpeedButton1.Enabled:=true;

DM\_f.ADOTable5.Cancel;

end;

procedure TOformlenie\_poezdki\_f.Button2Click(Sender: TObject);

begin

if (DBEdit1.Text='') or (DBEdit2.Text='') or

(DBEdit3.Text='') or (DBEdit4.Text='')

then

ShowMessage('Заполните все поля')

else begin

DM\_f.ADOTable5.Post;

DBEdit1.ReadOnly:=true;

DBEdit2.ReadOnly:=true;

DBEdit3.ReadOnly:=true;

DBEdit4.ReadOnly:=true;

Button1.Enabled:=true;

Button2.Enabled:=false;

Button3.Enabled:=false;

SpeedButton1.Enabled:=true;

end;

end;

procedure TOformlenie\_poezdki\_f.Button1Click(Sender: TObject);

begin

DBEdit1.ReadOnly:=false;

DBEdit2.ReadOnly:=false;

DBEdit3.ReadOnly:=false;

DBEdit4.ReadOnly:=false;

Button1.Enabled:=false;

Button2.Enabled:=true;

Button3.Enabled:=true;

SpeedButton1.Enabled:=False;

DM\_f.ADOTable5.Insert;

DBEdit1.SetFocus;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton5Click(Sender: TObject);

begin

DM\_f.ADOTable5.Next;

end;

procedure TOformlenie\_poezdki\_f.SpeedButton4Click(Sender: TObject);

begin

DM\_f.ADOTable5.Prior;

end;

procedure TOformlenie\_poezdki\_f.Button4Click(Sender: TObject);

begin

DM\_f.ADOTable6.Insert;

SpeedButton1.Enabled:=False;

Button4.Enabled:=false;

Button5.Enabled:=true;

SpeedButton7.Enabled:=true;

end;

procedure TOformlenie\_poezdki\_f.Button5Click(Sender: TObject);

begin

DM\_f.ADOTable6.Cancel;

Button4.Enabled:=true;

Button5.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton1.Enabled:=true;

end;

procedure TOformlenie\_poezdki\_f.Button6Click(Sender: TObject);

begin

if

CheckBox2.Checked then

DM\_f.ADOTable5.Filtered:=false

else

begin

DM\_f.ADOTable5.Filtered:=false;

DM\_f.ADOTable5.Filter:='ФИО\_Клиента='''

+ (Edit1.Text)+'''';

DM\_f.ADOTable5.Filtered:=true;

end;

end;

procedure TOformlenie\_poezdki\_f.GroupBox5MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

DBEdit6.Text:=TimeToStr(DateTimePicker1.Time);

end;

procedure TOformlenie\_poezdki\_f.Button7Click(Sender: TObject);

begin

if (DBEdit16.Text='') and (DBEdit15.Text='') then

begin

ShowMessage('В данный момент нет свободных машин'); //Вывод сообщения

Glavnaya\_f.Show;

Oformlenie\_poezdki\_f.Hide;

end

else

if (DBEdit10.Text='') and (DBEdit8.Text='') then

begin

ShowMessage('В данный момент нет свободных водителей'); //Вывод сообщения

Glavnaya\_f.Show;

Oformlenie\_poezdki\_f.Hide;

end;

end;

procedure TOformlenie\_poezdki\_f.GroupBox1MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.GroupBox2MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.FormMouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.GroupBox6MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.DBGrid1MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.Button6MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.Edit1MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.Label10MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

procedure TOformlenie\_poezdki\_f.CheckBox2MouseMove(Sender: TObject;

Shift: TShiftState; X, Y: Integer);

begin

Button7.Click;

end;

end.

unit Operator

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, DBCtrls, StdCtrls, Mask, Grids, DBGrids, Buttons;

type

TOperatory\_f = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

SpeedButton7: TSpeedButton;

DBGrid1: TDBGrid;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

SpeedButton3: TSpeedButton;

SpeedButton8: TSpeedButton;

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton2: TSpeedButton;

SpeedButton1: TSpeedButton;

Label5: TLabel;

DBEdit5: TDBEdit;

procedure SpeedButton7Click(Sender: TObject);

procedure SpeedButton3Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton8Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton6Click(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Operatory\_f: TOperatory\_f;

implementation

uses Glavnaya, DM;

{$R \*.dfm}

procedure TOperatory\_f.SpeedButton7Click(Sender: TObject);

begin

Operatory\_f.Hide;

Glavnaya\_f.Show;

end;

procedure TOperatory\_f.SpeedButton3Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DBEdit4.ReadOnly:=False;

DBEdit5.ReadOnly:=False;

DBEdit1.SetFocus;

DM\_f.ADOTable4.Insert;

end;

procedure TOperatory\_f.SpeedButton2Click(Sender: TObject);

begin

DM\_f.ADOTable4.Prior;

end;

procedure TOperatory\_f.SpeedButton8Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DBEdit4.ReadOnly:=False;

DBEdit5.ReadOnly:=False;

DM\_f.ADOTable4.Edit;

DBEdit1.SetFocus;

end;

procedure TOperatory\_f.SpeedButton4Click(Sender: TObject);

begin

DM\_f.ADOTable4.Delete;

end;

procedure TOperatory\_f.SpeedButton5Click(Sender: TObject);

begin

if (DBEdit1.Text='') or (DBEdit2.Text='') or

(DBEdit3.Text='') or (DBEdit4.Text='')

then

ShowMessage('Заполните все поля')

else begin

DM\_f.ADOTable4.Post;

SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

DBEdit4.ReadOnly:=true;

DBEdit5.ReadOnly:=true;

end;

end;

procedure TOperatory\_f.SpeedButton6Click(Sender: TObject);

begin

SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

DBEdit4.ReadOnly:=true;

DBEdit5.ReadOnly:=true;

DM\_f.ADOTable4.Cancel;

end;

procedure TOperatory\_f.SpeedButton1Click(Sender: TObject);

begin

DM\_f.ADOTable4.Next;

end;

end.

unit Status

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Mask, DBCtrls, Buttons;

type

TStatus\_f = class(TForm)

GroupBox2: TGroupBox;

GroupBox1: TGroupBox;

SpeedButton10: TSpeedButton;

SpeedButton16: TSpeedButton;

DBEdit33: TDBEdit;

ComboBox2: TComboBox;

DBEdit32: TDBEdit;

SpeedButton18: TSpeedButton;

SpeedButton17: TSpeedButton;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label31: TLabel;

SpeedButton20: TSpeedButton;

SpeedButton21: TSpeedButton;

DBEdit35: TDBEdit;

ComboBox3: TComboBox;

DBEdit34: TDBEdit;

SpeedButton19: TSpeedButton;

SpeedButton22: TSpeedButton;

Label4: TLabel;

Label5: TLabel;

Label50: TLabel;

Label7: TLabel;

SpeedButton1: TSpeedButton;

procedure SpeedButton10Click(Sender: TObject);

procedure SpeedButton16Click(Sender: TObject);

procedure SpeedButton17Click(Sender: TObject);

procedure SpeedButton18Click(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure SpeedButton19Click(Sender: TObject);

procedure SpeedButton22Click(Sender: TObject);

procedure SpeedButton20Click(Sender: TObject);

procedure SpeedButton21Click(Sender: TObject);

procedure ComboBox3Change(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Status\_f: TStatus\_f;

implementation

uses DM, Glavnaya;

{$R \*.dfm}

procedure TStatus\_f.SpeedButton10Click(Sender: TObject);

begin

ComboBox2.Enabled:=True; //активация поля ввода информации

DM\_f.ADOTable3.Edit;// Изменение записи в таблице

Label31.Font.Color:=ClGray; //Изменение цвета надписи

if DBEdit33.Text='True' then

Label31.Caption:='Свободен' //Установка значения

else

Label31.Caption:='Занят';//Установка значения

SpeedButton16.Enabled:=True; //Активация кнопки

SpeedButton10.Enabled:=False;//Деактивация кнопки

SpeedButton17.Enabled:=False; //Деактивация кнопки

SpeedButton18.Enabled:=False; //Деактивация кнопки

end;

procedure TStatus\_f.SpeedButton16Click(Sender: TObject);

begin

if DBEdit33.Text='True' then

Label31.Caption:='Свободен' //Установка значения

else

Label31.Caption:='Занят';//Установка значения

DM\_f.ADOTable3.FieldByName('Статус').AsString:=DBEdit33.Text; //Добавление строки в таблицу

DM\_f.ADOTable3.Post; //Сохранение записи в таблицу

Label31.Font.Color:=ClRed;//Изменение цвета надписи

SpeedButton16.Enabled:=False; //Деактивация кнопки

SpeedButton10.Enabled:=True; //Активация кнопки

SpeedButton17.Enabled:=True; //Активация кнопки

SpeedButton18.Enabled:=True; //Активация кнопки

ComboBox2.Text:=''; //Очистка поля ввода информации

ComboBox2.Enabled:=False; //Деактивация поля ввода информации

ShowMessage('Статус машины успешно изменен'); //Вывод сообщения

end;

procedure TStatus\_f.SpeedButton17Click(Sender: TObject);

begin

DM\_f.ADOTable3.Next;

if DBEdit33.Text='True' then

Label31.Caption:='Свободен' //Установка значения

else

begin

Label31.Caption:='Занят';//Установка значения

ComboBox2.Text:=''; //Очистка поля ввода информации

end;

end;

procedure TStatus\_f.SpeedButton18Click(Sender: TObject);

begin

DM\_f.ADOTable3.Prior;

if DBEdit33.Text='True' then

Label31.Caption:='Свободен'//Установка значения

else

begin

Label31.Caption:='Занят';//Установка значения

ComboBox2.Text:=''; // Очистка поля ввода информации

end;

end;

procedure TStatus\_f.ComboBox2Change(Sender: TObject);

begin

if ComboBox2.Text='Свободен' then

DBEdit33.Text:='True' //Установка значения

else

DBEdit33.Text:='False' //Установка значения

end;

procedure TStatus\_f.FormShow(Sender: TObject);

begin

DM\_f.ADOTable3.Filtered:=False; //Деактивация фильтра

DM\_f.ADOTable2.Filtered:=False; //Деактивация фильтра

if DBEdit33.Text='true' then

Label31.Caption:='Свободен' //Установка значения

else

Label31.Caption:='Занят';//Установка значения

if DBEdit35.Text='true' then

Label50.Caption:='Свободен'

else

Label50.Caption:='Занят';

end;

procedure TStatus\_f.SpeedButton19Click(Sender: TObject);

begin

ComboBox3.Enabled:=True; //Активация поля ввода информациии

DM\_f.ADOTable2.Edit; //Изменение записи в таблице

Label50.Font.Color:=ClGray; //Изменение цвета надписи

if DBEdit35.Text='True' then

Label50.Caption:='Свободен' //Установка значения

else

Label50.Caption:='Занят'; //Установка значения

SpeedButton22.Enabled:=True; //Активация кнопки

SpeedButton19.Enabled:=False; //Деактивация кнопки

SpeedButton20.Enabled:=False; //Деактивация кнопки

SpeedButton21.Enabled:=False; //Деактивация кнопки

ComboBox2.Text:=''; //Очистка поля ввода информации

end;

procedure TStatus\_f.SpeedButton22Click(Sender: TObject);

begin

if DBEdit35.Text='True' then

Label50.Caption:='Свободен'//Установка значения

else

Label50.Caption:='Занят';//Установка значения

DM\_f.ADOTable2.FieldByName('Статус').AsString:=DBEdit35.Text; //Добавление записи в таблицу

DM\_f.ADOTable2.Post; //Сохранение записи в таблице

Label50.Font.Color:=ClRed; //Изменение цвета надписи

SpeedButton22.Enabled:=False; //Деактивация кнопки

SpeedButton19.Enabled:=True; //Активация кнопки

SpeedButton20.Enabled:=True; //Активация кнопки

SpeedButton21.Enabled:=True; //Активация кнопки

ComboBox3.Text:=''; // Очистка поля ввода информации

ComboBox3.Enabled:=False; //Деактивация поля ввода информации

ShowMessage('Статус водителя успешно изменен');// Вывод сообщения

end;

procedure TStatus\_f.SpeedButton20Click(Sender: TObject);

begin

DM\_f.ADOTable2.Prior; //Навигация по таблице

if DBEdit35.Text='True' then

Label50.Caption:='Свободен'

else

begin

Label50.Caption:='Занят';

ComboBox3.Text:=''; //Очистка поля ввода информации

end;

end;

procedure TStatus\_f.SpeedButton21Click(Sender: TObject);

begin

DM\_f.ADOTable2.Next; //Навигация по таблице

if DBEdit35.Text='True' then

Label50.Caption:='Свободен'

else

begin

Label50.Caption:='Занят';

ComboBox3.Text:='';//Очистка поля ввода информации

end;

end;

procedure TStatus\_f.ComboBox3Change(Sender: TObject);

begin

if ComboBox3.Text='Свободен' then

DBEdit35.Text:='True' //Установка значения

else

DBEdit35.Text:='False' //Установка значения

end;

procedure TStatus\_f.SpeedButton1Click(Sender: TObject);

begin

Glavnaya\_f.Show; //открытие формы главная

Status\_f.Hide;// скрытие формы

end;

end.

unit Voditeli

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Buttons, StdCtrls, ExtCtrls, DBCtrls, DB, ADODB, Grids, DBGrids,

Mask;

type

TVoditeli\_f = class(TForm)

DBGrid1: TDBGrid;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

SpeedButton7: TSpeedButton;

DBEdit5: TDBEdit;

DBEdit7: TDBEdit;

Label5: TLabel;

Label7: TLabel;

SpeedButton3: TSpeedButton;

SpeedButton8: TSpeedButton;

SpeedButton4: TSpeedButton;

SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;

SpeedButton1: TSpeedButton;

SpeedButton2: TSpeedButton;

procedure SpeedButton7Click(Sender: TObject);

procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit2KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit3KeyPress(Sender: TObject; var Key: Char);

procedure SpeedButton3Click(Sender: TObject);

procedure SpeedButton8Click(Sender: TObject);

procedure SpeedButton4Click(Sender: TObject);

procedure SpeedButton5Click(Sender: TObject);

procedure SpeedButton6Click(Sender: TObject);

procedure SpeedButton2Click(Sender: TObject);

procedure SpeedButton1Click(Sender: TObject);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Voditeli\_f: TVoditeli\_f;

implementation

uses Glavnaya, DM;

{$R \*.dfm}

procedure TVoditeli\_f.SpeedButton7Click(Sender: TObject);

begin

Voditeli\_f.Hide;//скрытие формы водители

Glavnaya\_f.Show;//открытие формы главная

end;

procedure TVoditeli\_f.DBEdit1KeyPress(Sender: TObject; var Key: Char);

begin

if key=#13 then //нажатие Enter

DBEdit2.SetFocus;//Переход на следующую строку при нажатии Enter

end;

procedure TVoditeli\_f.DBEdit2KeyPress(Sender: TObject; var Key: Char);

begin

if key=#13 then //нажатие Enter

DBEdit3.SetFocus;//Переход на следующую строку при нажатии Enter

end;

procedure TVoditeli\_f.DBEdit3KeyPress(Sender: TObject; var Key: Char);

begin

if key=#13 then //нажатие Enter

DBEdit4.SetFocus;//Переход на следующую строку при нажатии Enter

end;

procedure TVoditeli\_f.SpeedButton3Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DBEdit4.ReadOnly:=False;

DBEdit5.ReadOnly:=False;

DBEdit7.ReadOnly:=False;

DBEdit1.SetFocus;

DM\_f.ADOTable2.Insert;

end;

procedure TVoditeli\_f.SpeedButton8Click(Sender: TObject);

begin

SpeedButton3.Enabled:=false;

SpeedButton4.Enabled:=false;

SpeedButton2.Enabled:=false;

SpeedButton1.Enabled:=false;

SpeedButton7.Enabled:=false;

SpeedButton8.Enabled:=false;

SpeedButton5.Enabled:=true;

SpeedButton6.Enabled:=true;

DBEdit1.ReadOnly:=False;

DBEdit2.ReadOnly:=False;

DBEdit3.ReadOnly:=False;

DBEdit4.ReadOnly:=False;

DBEdit5.ReadOnly:=False;

DBEdit7.ReadOnly:=False;

DM\_f.ADOTable2.Edit;

DBEdit1.SetFocus;

end;

procedure TVoditeli\_f.SpeedButton4Click(Sender: TObject);

begin

DM\_f.ADOTable2.Delete;

end;

procedure TVoditeli\_f.SpeedButton5Click(Sender: TObject);

begin

if (DBEdit1.Text='') or (DBEdit2.Text='') or

(DBEdit3.Text='') or (DBEdit4.Text='') or

(DBEdit5.Text='') or (DBEdit7.Text='')

then

ShowMessage('Заполните все поля')

else begin

DM\_f.ADOTable2.Post;

SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

DBEdit4.ReadOnly:=true;

DBEdit5.ReadOnly:=True;

DBEdit7.ReadOnly:=True;

end;

end;

procedure TVoditeli\_f.SpeedButton6Click(Sender: TObject);

begin SpeedButton3.Enabled:=true;

SpeedButton4.Enabled:=true;

SpeedButton2.Enabled:=true;

SpeedButton1.Enabled:=true;

SpeedButton7.Enabled:=true;

SpeedButton8.Enabled:=true;

SpeedButton5.Enabled:=False;

SpeedButton6.Enabled:=False;

DBEdit1.ReadOnly:=True;

DBEdit2.ReadOnly:=True;

DBEdit3.ReadOnly:=True;

DBEdit4.ReadOnly:=true;

DBEdit5.ReadOnly:=True;

DBEdit7.ReadOnly:=True;

DM\_f.ADOTable2.Cancel;

end;

procedure TVoditeli\_f.SpeedButton2Click(Sender: TObject);

begin

DM\_f.ADOTable2.Prior;

end;

procedure TVoditeli\_f.SpeedButton1Click(Sender: TObject);

begin

DM\_f.ADOTable2.Next;

end;

procedure TVoditeli\_f.FormShow(Sender: TObject);

begin

DM\_f.ADOTable2.Filtered:=false;

end;

end.