СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

1.1 Построение концептуальной модели

1.2 Построение логической модели

1.3 Выбор программного обеспечения для разработки интерфейса информационной системы

1.4 Экономическое обоснование оплаты труда

2 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ УЧИТЕЛЯМ ШКОЛЫ»

2.1 Создание базы данных с помощью Access

2.2 Подключение базы данных к Delphi

2.3 Описание программной реализации

2.4 Элементы интерфейса информационной системы "Начисление заработной платы работникам средне-специальных учебных заведений"

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЕ A

ПРИЛОЖЕНИЕ Б

ПРИЛОЖЕНИЕ В

ВВЕДЕНИЕ

Для осуществления планирования любой фирмы на любом этапе работы осуществляются некоторые расчеты для автоматизации, которых удобно применять вычислительную технику.

Основой для учета, контроля и планирования служат всевозможные картотеки, регистрационные журналы, списки и т.д. Они постепенно накапливаются и обновляются. При большом объеме информации поиск и обобщение необходимых сведений, осуществляемых вручную, представляют собой довольно трудоемкий процесс.

С появлением ЭВМ и использованием их для обработки информации появилась возможность автоматизировать решение многих информационно-справочных и расчетных задач.

Постепенно с развитием программного обеспечения ЭВМ появились идеи создания управляющих систем, которые позволяли бы накапливать, хранить и обновлять взаимосвязанные данные по целому комплексу решаемых задач, например при автоматизации бухгалтерского учета, на предприятии. Эти идеи нашли свое воплощение в системах управления базами данных (СУБД). С появлением персональных компьютеров, СУБД становится наиболее популярным средством обработки табличной информации. Они являются инструментальным средством проектирования банков данных при обработке больших объемов информации [1, c.94].

Программное обеспечение для работы с базами данных используется на персональных компьютерах уже довольно давно. К сожалению, эти программы либо были элементарными диспетчерами хранения данных и не имели средств разработки приложений, либо были на столько сложны и трудны, что даже хорошо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

Система управления базами данных предоставляет возможность контролировать задание структуры и описание своих данных, работу с ними и организацию коллективного пользования этой информацией. СУБД также существенно увеличивает возможности и облегчает каталогизацию и ведение больших объемов хранящейся в многочисленных таблицах информации. На практике, как правило, после подсчета или анализа данных необходимо их представить в виде определенной формы или шаблона. В итоге пользователю приходится комбинировать программные продукты для получения необходимого результата. В этом смысле все существенно упростят возможности, предоставляемые Borland Delphi. Возможность создавать программы, обслуживающие базы данных, - важная отличительная особенность Delphi.

В данной выпускной работе требуется произвести расчет заработной платы сотрудникам средне-специальных учебных заведений, организовать выход необходимой информации на печать в виде отчетов, создать удобный интерфейс, для информационной системы «Начисление заработной платы сотрудникам средних учебных заведений», ориентированный на комфортную работу пользователя.

Целью выпускной работы является создание программного обеспечения для расчета заработной платы сотрудникам средне-специальных учебных заведений, организация выходной информации в виде отчетов, создание установочного файла.

Актуальность. Расчет заработной платы сотрудникам средне-специальных учебных заведений производиться бухгалтерами либо с помощью программы «1С-бухгалтерия», либо в ручную. Так как программа «1С-бухгалтерия» очень сложна в применении, и ее может освоить не каждый бухгалтер, то расчет заработной платы производится с помощью электронных таблиц. Это довольно трудоемкий процесс, который занимает очень много времени и ресурсов компьютера. Данная выпускная работа рассматривает принципы создания информационной системы «Начисление заработной платы сотрудникам школы», ориентированная на комфортную работу бухгалтеров средне – специальных заведений.

Новизна выпускной работы состоит в том, что был создан интерфейс пользователя для базы данных Access, с помощью современной системы программирования Delphi 7, которая является на сегодняшний день новейшей версией.

Практическая ценность – предназначена для бухгалтеров средне – специальных учебных заведений.

Данная выпускная работа состоит из двух разделов, содержит 37 рисунков и одну таблицу.

1 ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

1.1 Построение концептуальной модели

Требование пользователей к разрабатываемой базе данных представляет собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики базы данных получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Перечень вопросов при анализе различных предметных областей базы данных для начисления заработной платы учителям школы:

* Сколько учителей работает в школе?
* Сколько часов в неделю, и в каких классах проводится тот или иной предмет?
* Сколько дополнительных часов, часов замены было проведено?
* Имеется ли классное руководство?
* Сколько было отработанных дней из рабочих дней в месяце?
* Как часто обновляется информация в базе данных? и т.д.

При описании предметной области гораздо удобнее и эффективнее использовать такое её представление, которое было бы понятным для большинства категорий пользователей, такой обобщенный взгляд на информационную систему и на её функционирование реализуется через концептуальное проектирование [1, c.94].

Концептуальная модель представляет объекты предметной области и их взаимосвязи без указания способов их физического хранения. Таким образом, концептуальная модель является, по существу, моделью предметной области. При проектировании концептуальной модели все усилия разработчика должны быть направлены на структуризацию данных и выявлений взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки. Проектирование концептуальной модели основано на анализе решаемых, на этом предприятии задач по обработке данных [1, c.94-95].

Концептуальная модель включает описание объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных [1, c.96].

Результатом концептуального проектирования является некоторая концептуальная схема, которая чаще всего выражается в виде ER диаграммы (диаграмма «сущности связи») [2, c.48-52], в которой приняты следующие обозначения:

- объекты; - связи;

Для решения поставленной задачи дипломной работы была построена концептуальная схема (рисунок 1) информационной системы «Начисления заработной платы».

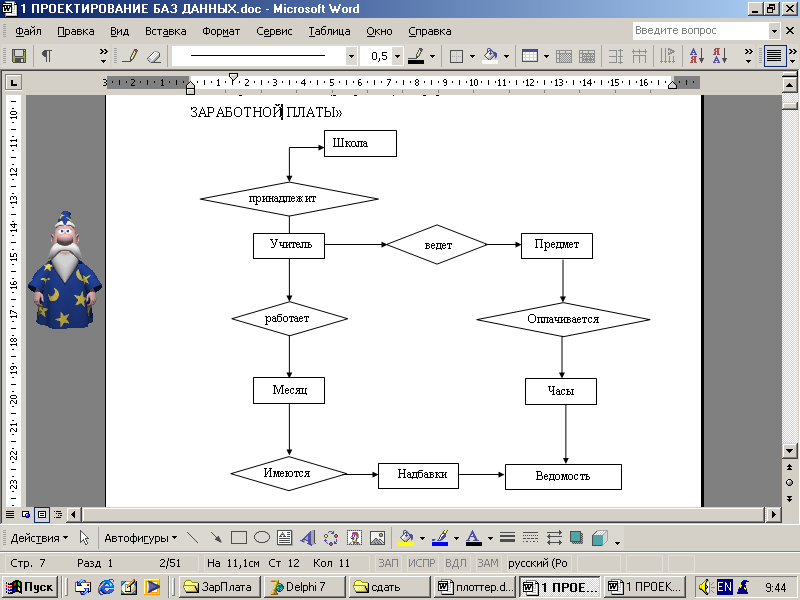


Рисунок 1 Концептуальная схема информационной системы

Концептуальная модель транспонируется в модель данных, совместимую с выбранной СУБД. Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называемой логической моделью.

1.2 Построение логической модели

Логическое проектирование представляет собой необходимый этап при создании базы данных. Основной задачей логического проектирования является разработка логической схемы, ориентированной на выбранную систему управления базами данных (СУБД). Этап логического проектирования в отличие от концептуального проектирования полностью ориентирован на инструментальные средства компьютера [2, c.95].

Этапы логического проектирования:

* Выбор конкретной СУБД.
* Отображение концептуальной схемы на логическую схему.
* Выбор ключей.
* Описание языков запросов.

Выбор СУБД зависит от того, на сколько эффективна внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему. Существующие СУБД делятся по типам моделей данных на реляционные, иерархические и сетевые. СУБД, ориентированные на персональные компьютеры, как правило, поддерживают реляционную модель данных. Если выбрана реляционная система, то концептуальную схему базы данных предстоит отображать на реляционную модель.

При отображении концептуальной схемы базы данных на реляционную модель данных следует учитывать ограничения на размеры таблиц, которая накладывает выбранная СУБД. По размеру полей можно определить требуемый объем памяти для хранения данной таблицы [2,c. 110-119].

Аналогично поступают со всеми остальными объектами концептуальной схемы.

База данных, созданная для реализации поставленной задачи дипломной работы, была реализована с помощью Access. Поэтому логическая модель имеет следующий вид (таблица 1).

Таблица 1 Логическая модель информационной системы «Начисление заработной платы учителям школы»

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование таблицы | Ключевое поле | Наименование поля | Описание поля |
| Кадры | Ключ | Код сотрудника | Числовой |
|  | ФИО сотрудника | Текстовый |
|  | Стаж | Текстовый |
|  | Адрес | Текстовый |
|  | Разряд | Числовой |
|  | БДО | Денежный |
|  | РабДн | Числовой |
|  | ПропДн | Числовой |
| Сведения о классах (Часы) | Ключ | Код начисления | Числовой |
|  | Код сотрудника | Числовой |
|  | Код должности | Числовой |
|  | Код предмета | Числовой |
|  | Дата начисления | Дата |
|  | 0-ые классы | Числовой |
|  | 1-4 классы | Числовой |
|  | 5-9 классы | Числовой |
|  | 10-11 классы | Числовой |
|  | Надомники (0) класс | Числовой |
|  | Надомники (1-4) класс | Числовой |
|  | Надомники (5-9) класс | Числовой |
|  | Надомники (10-11) класс | Числовой |
|  | ИндГрКонс (0) класс | Числовой |
|  | ИндГрКонс (1-4) класс | Числовой |
|  | ИндГрКонс (5-9) класс | Числовой |
|  | ИндГрКонс (10-11) класс | Числовой |
|  | Замена (0) класс | Числовой |
|  | Замена (1-4) класс | Числовой |
|  | Замена (5-9) класс | Числовой |
|  | Замена (10-11) класс | Числовой |
|  | Коэффиц. классного руковод. | Числовой |
|  | Коэффиц. внеклассной работы | Числовой |
| Сведения о классах (Часы) |  | Коэффиц. Углублен. изучение | Числовой |
|  | Коэффиц. проверки тетрадей | Числовой |
|  | Коэффиц. вредности | Числовой |
|  | Коэффиц. расчета | Числовой |
|  | G(группа) | Числовой |
|  | Коэффиц. ОПВ | Числовой |
|  | Сумма ИПН | Числовой |
|  | К выплате | Денежный |
| Должности | Ключ | Код должности | Числовой |
|  | наименов. Должности | Текстовый |
| Предметы | Ключ | Код предмета | Числовой |
|  | наименов. Предмета | Текстовый |
| Coff |  | Коэффиц. Нед. Час. (1-4) класс | Числовой |
|  | Коэффиц. Нед. Час. (5-11) класс | Числовой |
|  | Коэффиц. раз. Час. (1-4) класс | Числовой |
|  | Коэффиц. раз. Час. (5-11) класс | Числовой |
|  | Экология | Числовой |
| Год | Ключ | Код сотрудника | Числовой |
|  | Дата\_начисления | Дата |
|  | Начислено экологии | Денежный |
|  |  | Начислено за внеклассную работу | Денежный |
|  | Начислено за проверку тетрадей | Денежный |
|  |  | Начислено за вредность | Денежный |
| Год |  | Начисленно за классное руководство | Денежный |
|  | Начисленно за замену часов | Денежный |
|  | Начисленно за углубленное изучение | Денежный |
|  | Начисленно за замену углубленного изучения | Денежный |
|  | Заработная плата | Денежный |
|  | Итого | Денежный |

Логическая модель отражает логические связи между элементами данных вне зависимости их содержания и среды хранения.

Логическая модель данных может быть реляционной, иерархической или сетевой. Логическая модель отображается в логическую память, такую как диск, лента или какой-либо другой носитель информации.

1.3 Выбор программного обеспечения для разработки интерфейса информационной системы

В последнее время резко возрос интерес к программированию. Это связано с развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий. Если человек имеет дело с компьютером, то рано или поздно у него возникает желание, а иногда и необходимость, программировать. Среди пользователей персональных компьютеров в настоящее время наиболее популярно семейство операционных систем Windows и, естественно, что тот, кто собирается программировать, стремится писать программы, которые будут работать в этих системах. Несколько лет назад рядовому программисту оставалось только мечтать о создании собственных программ, работающих в среде Windows, т.к. единственным средством разработки был Borland C++ for Windows, явно ориентированный на профессионалов, обладающих серьезными знаниями и опытом [3, c.2].

Бурное развитие вычислительной техники, потребность в эффективных средствах разработки программного обеспечения привели к появлению систем программирования, ориентированных на так называемую "быструю разработку", среди которых можно выделить Borland Delphi и Microsoft Visual Basic.

В основе систем быстрой разработки (RAD-систем, Rapid Application Development — среда быстрой разработки приложений) лежит технология визуального проектирования и событийного программирования, суть которой заключается в том, что среда разработки берет на себя большую часть рутинной работы, оставляя программисту работу по конструированию диалоговых окон и функций обработки событий. Производительность программиста при использовании RAD-систем -фантастическая! [3, c.3-4]

Delphi — это среда быстрой разработки, в которой в качестве языка программирования используется язык Delphi. Язык Delphi — строго типизированный объектно-ориентированный язык, в основе которого лежит хорошо знакомый программистам Object Pascal [4, c. 3-4].

Мощность и гибкость языка программирования Delphi – безусловное достоинство Delphi, отличающую эту среду от других инструментов RAD. Ядром языка Delphi является язык Pascal. За долгие годы язык Pascal превратился в мощный современный объектно-ориентированный язык, пригодный для эффективного создания программ любой сложности.

От языка Visual Basic Delphi отличает строгая типизированность, позволяющая компилятору еще на этапе компиляции обнаружить многие ошибки, а также наличие средств работы с указателями. Таким образом, в Delphi используется так называемое ранее связывание с библиотеками типов в технологии COM, в то время как в Visual Basic используется более медленное позднее связывание при обращении к COM.

Синтаксис С++ провоцирует создание запутанных программ, в которых трудно разобраться даже автору, в то время как простой и ясный синтаксис Delphi позволяет ему претендовать на роль языка, идеально подходящего для описания алгоритма. Во всех случаях Delphi имеет самый быстрый среди продуктов подобного рода оптимизирующий компилятор, позволяющий создавать быстрые и относительно компактные программы [4, c.5].

Система Delphi известна как самое эффективное средство разработки приложений баз данных, то есть программ, обслуживающих электронные хранилища информации. Это определяется по трем обстоятельствам:

* Высокопроизводительная машина доступа к данным разного формата(Borland Database Engine, BDE).
* Наличие многочисленных компонентов и технологий, ориентированных на эту сферу применения.
* Поставка вместе с Delphi компактного, мощного и простого в администрировании сервера баз данных InterBase.

Машина BDE является средством доступа к данным разного формата. Особенностью BDE является использование собственных высокопроизводительных драйверов для доступа к локальным таблицам dBase, Paradox, FoxPro, а также SQL-ориентированных драйверов для доступа к серверам баз данных (этот набор называется SQL Links). При желании можно использовать драйверы сторонних производителей, соответствующие стандарту ODBC(Open Database Connectivity Interface – открытый интерфейс взаимодействия с базами данных). Более того в версии 6 и 7 появилась возможность использования альтернативных BDE средств доступа по технологиям ADO, InterBase Express, dbExpress.

Многочисленные компоненты, поддерживающие разработку приложений баз данных, обеспечивают обслуживание самых разных задач, таких как выборка и сортировка данных, их наглядное представление, изменение и публикация данных в виде отчетов (документов) или HTML-страниц в Интернете и т.д. [4, c.6].

В состав пакета включены разнообразные утилиты, обеспечивающие работу с базами данных, XML-документами, создание справочной системы, решение других задач. Отличительной особенностью седьмой версии является поддержка технологии NET. Borland Delphi 7 Studio может работать в среде операционных систем от Windows 98 до Windows XP [4, c.10].

Последняя на сегодня версия 7 сохраняет почти полную преемственность по языку программирования и интегрированной среде разработчика с версией 6. Ее характерное отличие – введение новых технологий, облегчающих (и упрощающих) создание программ для баз данных и Интернета [4, c.13-14].

1.4 Экономическое обоснование оплаты труда

В процессе хозяйственной деятельности предприятия происходит прием и увольнение работников, перемещение по работе, меняется численность персонала. Учет личного состава ведет отдел кадров, который обеспечивает получение данных о численности персонала на предприятии в целом, его структурных подразделениях, причинах ее изменения, о составе персонала по полу, возрасту, категориям, профессиям, должностям, специальности, квалификации, стажу работы, образованию и другим признакам.

При приеме на работу работодатель и работник заключают между собой индивидуальный трудовой договор о месте работы, сроке действия договора, об условиях режима труда, отдыха, оплаты, охраны труда, иных вопросах социально-бытового обеспечения работника, материальной ответственности сторон договора и др. [5, c.407]

Учет использования рабочего времени всех категорий работающих ведут в табеле учета рабочего времени, для контроля за соблюдением рабочими и служащими установленного режима рабочего времени, получение данных об отработанном времени, расчета заработной платы. Составляется табель уполномоченным лицом; после соответствующего оформления он передается в бухгалтерию. Отметки в табеле о причинах неявок на работу или о работе неполных рабочий день, о работе в сверхурочное время и других отступлениях от нормальных условий должны быть сделаны только на основании документов, оформленных надлежащим образом. Учет использования рабочего времени осуществляю в табелях методом сплошной регистрации явок и неявок на работу либо путем регистрации только отклонений (неявок, опозданий).

Работодатель самостоятельно разрабатывает систему оплаты труда в форме положения «Об оплате труда» с учетом конкретных условий труда, категорий работников и режимов рабочего времени, установленных в положении «О персонале» [5, c.410-413].

В практике организации оплаты труда используют две основные формы:

* При повременной форме оплаты труда заработную плату начисляют работнику по установленной тарифной ставке или окладу за фактически отработанное время.
* При сдельной форме оплаты труда заработную плату работнику начисляют в заранее установленном размере за каждую единицу выполненной работы.

В соответствии с действующим законодательством работодатели удерживают из заработной платы работников: обязательные пенсионные взносы (ОПВ) в накопительные пенсионные фонды, индивидуальный подоходный налог (ИПН), суммы по исполнительным листам, а также производят любые другие удержания по письменному заявлению работников, либо по инициативе самого работодателя, без согласия работника: для возращения неизрасходованного аванса, выданного в счет заработной платы. При увольнении работника по его вине за неотработанные им дни полученного отпуска; для возмещения вреда, причиненного по вине работника предприятию, если размер вреда не превышает месячного заработка работника [5, c.425-426].

Удержание индивидуального подоходного налога. Заработная плата, а также любые другие доходы работников, выплачиваемые работодателями в денежной или натуральной форме, включая доходы, предоставленные работодателем в виде материальных, социальных благ или иной материальной выгоды подлежат налогообложению индивидуальным подоходным налогом по ставкам. Исчисление и удержания индивидуального подоходного налога производят ежемесячно нарастающим итогом в установленном законодательством порядке [5, c.429].

При увольнении работника до истечения налогового года производится перерасчет индивидуального подоходного налога, исходя из полученного дохода за фактически проработанный период с учетом коэффициента перерасчета, равного единице. Согласно перерасчету работнику предоставляется справка о доходах и налогах за проработанный период. При изменении места работы физическое лицо должно представить по новому месту работы расчет с прежнего места работы о полученном доходе и уплаченном налоге за проработанный период [5, c.425-426].

Обязательные пенсионные взносы в накопительные пенсионные фонды удерживаются и перечисляются работодателем за работников в размере 10% из доходов, выплачиваемых работникам в виде оплаты труда, включая все виды заработной платы, а также различные премии, доплаты, надбавки.

Вычеты из оплаты труда членских профсоюзных взносов осуществляют по письменным заявлениям работников. Членские взносы, удержанные из оплаты труда работников, предприятие перечисляет на банковский счет соответствующей профсоюзной организации [5, c.429].

Самым волнующим для всех бухгалтеров страны на сегодняшний день стал вопрос правильного расчета социального налога (СН) в связи с введением с 2005 года социальных отчислений (СО) [6, c.6].

Для расчета социального налога необходимо [6, c.7]:

* Указать доходы, не подлежащие налогообложению в соответствии со ст.144 Налогового кодекса.
* Указать обязательные пенсионные взносы.
* Указать сумму налоговых вычетов, определяемых в соответствии с пп. 1,2 и 4 – 6 пункта 1ст. 152 Налогового кодекса.
* Коэффициент перерасчета.
* Расчетная сумма облагаемых ИПН доходов.
* Сумма ИПН с расчетной суммы облагаемых ИПН доходов.
* Сумма ИПН за отчетный месяц.
* Доходы, облагаемые социальным налогом за отчетный месяц.
* Расчетная сумма дохода, облагаемая социальным налогом.
* Сумма социального налога с расчетной суммы дохода, облагаемый социальным налогом.
* Сумма социального налога за отчетный месяц.
* Сумма социального налога за подлежащего уплате.
* Доходы физических лиц, с которых исчисляются социальные исчисления.
* Сумма социальных отчислений.
* Сумма социального налога к уплате и отражению в декларации.

Коэффициент перерасчета сотрудника определяется на основании пункта 11 пункта 2 Правил исчисления индивидуального подоходного налога и исчисления социального налога. То есть, если работник был принят с начала года, то коэффициент перерасчета равен 12. К примеру, если работник начал работать с марта месяца, то его коэффициент будет равен 10, так как он проработает до конца налогового периода 10 месяцев.

2. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ «НАЧИСЛЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ СОТРУДНИКАМ ШКОЛЫ»

2.1 Создание базы данных с помощью Access

Для осуществления планирования любой фирмы на любом этапе работы осуществляются некоторые расчеты, для автоматизации которых удобно и рационально применять вычислительную технику.

В данной задаче требуется рассчитать заработную плату сотрудникам школы на месяц, с учетом того, что известны все необходимые данные, такие как разряд, от которого зависит коэффициент учителя, базовый должностной оклад, количество отработанных дней из рабочих дней в месяце, количество плановых часов, дополнительных часов, часов замены, коэффициенты классного руководства, внеклассной работы и прочие кадровые данные.

Для решения данной задачи первоначально был создан документ для начисления заработной платы бухгалтером школы №39 г. Усть-Каменогорска, с помощью электронной таблицы Microsoft Excel (рисунок 2) , в котором на каждого сотрудника был заведен отдельный лист. Microsoft Excel позволяет представлять данные в различных форматах и производить с ними некоторые арифметические действия с помощью формул, а также хранить, изменять и предоставлять информацию по первому требованию и в профессиональном виде.

В связи с тем, что электронная таблица содержит большое количество страниц, то работать с ним становится очень затруднительно. С ростом сотрудников в школе может возникнуть такая проблема как превышение установленных электронной таблицей ограничений на память или же вообще исчерпаны возможности компьютерной системы. Разработанная электронная таблица предназначена для многих пользователей, следовательно, проконтролировать ввод новых и использования уже имеющихся данных невозможно.

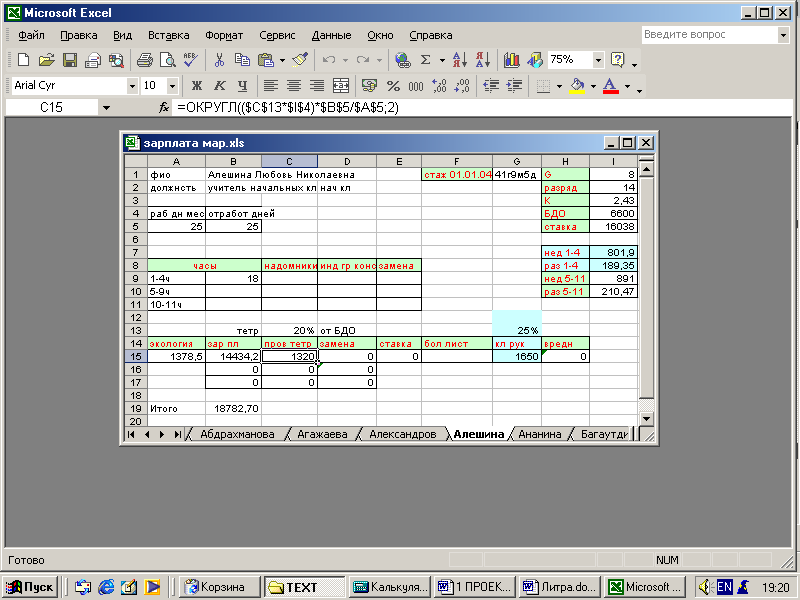


Рисунок 2. Расчет заработный платы в Microsoft Excel

Учтивая все выше перечисленное встала необходимость создания базы данных. При разработке информационной системы выбор остановился на Access 2000. Так как Microsoft Access – это функционально полная СУБД. В ней предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации [7, c.27-29]. СУБД Access очень легка в использовании и входит в состав пакета Microsoft Office, что очень важно для компьютеров, установленных в школах, а также обладает таким достоинством как интегрированность с электронными таблицами Excel, Word и другими программами пакета Microsoft Office. Microsoft Access значительно упрощает задачу обработки данных. Предназначенная для коллективного пользования СУБД имеет средства, не позволяющие нескольким пользователям одновременно корректировать одни и те же данные, а также ее установка не требует дополнительных усилий и больших мощностей компьютера [8, c.87].

С помощью Microsoft Access 2000 была создана база данных «ЗарПлата», в которой содержаться таблицы с данными каждого сотрудника, используемые при начислении заработной платы. Таблицы «Кадры», «Должности», «Предметы», «Начисления», «Коэффициенты» и «Год» связанны между собой связями «один – ко многим».

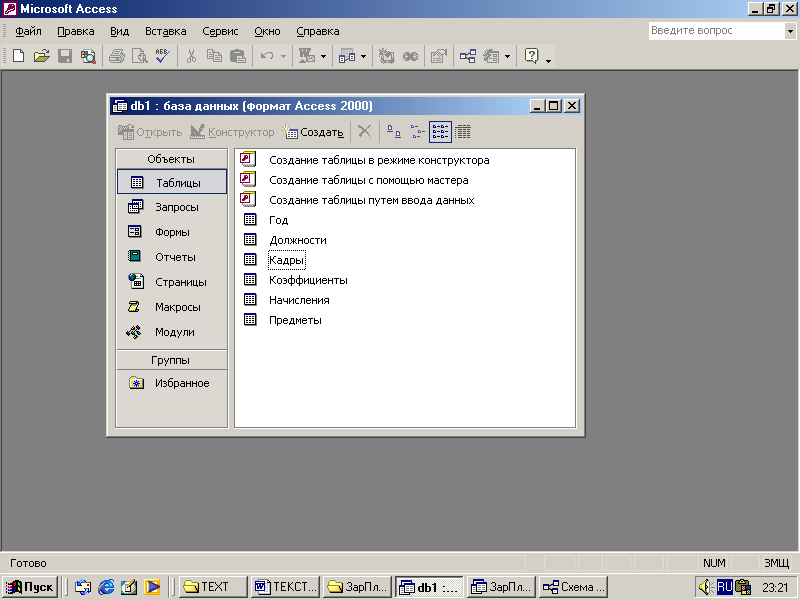


Рисунок 3. База данных «Начисление заработной платы сотрудникам школы» в формате Access

2.2 Подключение базы данных к Delphi

Создание пользовательского интерфейса производилось с помощью языка программирования Delphi. При подключении базы данных Access к среде Delphi были использованы основные таблицы, содержащие информацию о работниках школы. Подключение производилось с помощью технологии ActiveX Data Object (ADO – объекты данных, построенные как объекты ActiveX) и компонентов ADOConnection, ADOTable, DataSource.

Основные особенности использования технологии ADO не зависят от архитектуры баз данных: эта технология характерна не только для файл-серверных баз данных, но также и для клиент-серверных и трехзвенных баз данных.

Основным достоинством технологии ADO является ее естественная ориентация на создание «облегченного» клиента. В рамках этой технологии на машине разработчика баз данных устанавливаются базовые объекты MSADO и соответствующие компоненты Delphi, обеспечивающие использование технологии ADO. На машине сервера данных устанавливается так называемый провайдер данных – некоторая надстройка над специальной технологией OLE DB, понимающая запросы объектов ADO и умеющая переводить эти запросы в нужные действия с данными. Взаимодействия компонентов ADO и провайдера осуществляется на основе технологии ActiveX, причем провайдер – COM сервер, а ADO-компоненты – COM клиенты.

На машине сервера создается и размещается источник данных. В случае файл-серверных систем отдельные таблицы должны управляться соответствующим ODBC-драйвером. В данной выпускной работе были использованы таблицы в формате Access, данный файл управляется машиной баз данных Microsoft Jet 4.0 Database Engine, а в роли провайдера используется Microsoft Jet4.0 OLE DB Provider [9, c.65-68].

На начальном этапе создания пользовательского интерфейса был создан проект и к нему добавлен модуль данных. В модуль данных были помещены один компонент ADOConnection, четыре компонент ADOTable и четыре компонент DataSource. Таблицы были названы Person, Predmety, Dolgnosty, Coff, Move, Year и связаны с источниками данных DataSource, которые получили соответствующие названия Person1, Predmety1, Dolgnosty1, Coff1, Move1, Year1, (рисунок 4).

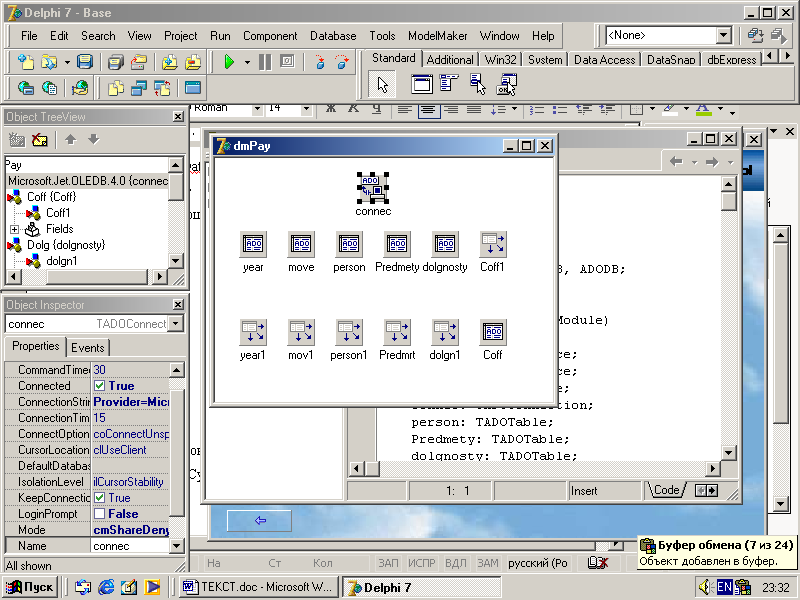


Рисунок 4ю Модуль данных

Установление связи с объектом ADO является ключевым моментом всей технологии.

Существует два пути установления связи:

* Диалоговый способ формирования связи.
* Формирование связного файла.

Для того чтобы настроить связь объектов ADO с соответствующим провайдером, необходимо в свойстве ConnectionString компонента ADOConnection1 вызвать окно настройки [9, c.73-78]. На экране появиться диалоговое окно с четырьмя вкладками:

* «Поставщик данных», где указывается провайдер Microsoft Jet4.0 OLE DB Provider, предназначенный для работы с базами данных Access.
* «Подключение», где указывается путь доступа к файлу базы данных.
* «Дополнительно» позволяет задать дополнительные свойства связи.
* «Все» здесь приводятся все параметры связи, заданные как явно так и назначенные по умолчанию.

Далее в свойстве TableName компонентов ADOTable были указаны соответствующие названия таблиц из базы данных Access и установлены связи между данными таблицами на вкладке Diagram по ключевому полю Код, (рисунок 5).

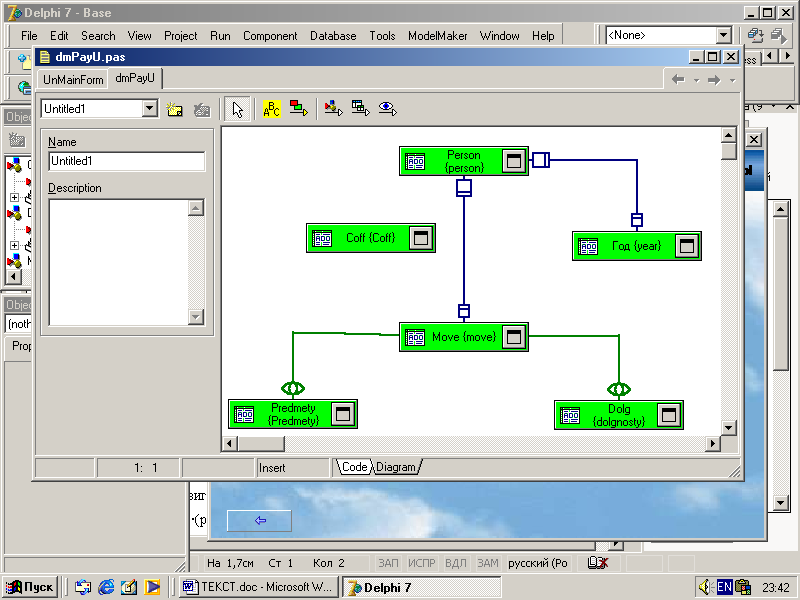


Рисунок 5 Отображение реляционной связи

2.3 Описание программной реализации

Результатом дипломной работы явилось создание программного обеспечения, предназначенного для расчета заработной платы сотрудникам средне-специальных учебных заведений.

Работа выполнялась на основании индивидуального задания на дипломное проектирование по специальности 010240 “Прикладная математика и информатика” на тему “Автоматизированная система начисления заработной платы сотрудникам средне-специальных учебных заведений».

Назначение разрабатываемого программного обеспечения – расчет заработной платы сотрудникам средне-специальных учебных заведений.

Автоматизации подлежат следующие функции:

* ввод данных и контроль за вводом данных;
* редактирование и сохранение данных;
* расчет заработной платы;
* создание отчётов;

Положительными результатами использования программного продукта являются:

* сокращение потерь времени за счет автоматизации получения документации;
* повышение производительности труда бухгалтера школы, при выполнении расчетных операций;

Целью создания программного обеспечения является автоматизация расчётов, высвобождение времени, затрачиваемого на реализацию данных расчетов вручную, а также реализация конкретной задачи в рамках программного комплекса с учетом всех требований пользовательского интерфейса и систем требований для оформления приложений Windows.

Разработанное программное обеспечение должен обеспечивать выполнение следующих функций:

* ведение данных;
* расчет заработной платы сотрудникам школы за текущий месяц с соответствующими налоговыми вычетами;
* представление выходной информации в виде отчетов;
* вывод заработной платы сотрудникам школы за год;

Проектирование интерфейса пользователя осуществлялось в соответствии со стандартом SAA, разработанным фирмой IBM. Были реализованы требования подсистемы CUA (Common User Access), определяющей компоненты пользовательского интерфейса, которые должны присутствовать во всех программах. Эти компоненты строятся на двух основных принципах:

* пользователи могут разрабатывать концепцию интерфейса;
* пользователь может и должен управлять диалогом;

С технической точки зрения для реализации стандарта использовались следующие приемы:

* разработка развитой системы меню;
* упрощение режимов обработки данных;
* создание многооконного интерфейса;
* система всплывающих подсказок;
* выдача дополнительных сообщений при осуществлении пользователем некорректных действий;
* используется система всплывающего меню, активизация происходит при нажатии правой кнопки мыши;
* использование стандартных названий в меню;
* система помощи;
* функциональные клавиши для быстрого доступа к командам меню;

Обеспечена унификация и непротиворечивость интерфейса. Продуманный, последовательный интерфейс программной системы и пользователя определяет скорость адаптации к данной системе и может значительно сократить время на выполнение функций данным пользователем. Интерфейс с пользователем поддерживает стандартные устройства ввода информации (клавиатура и мышь). При этом предусмотрена реализация большинства функций программной системы с помощью манипулятора “мышь”, так как это значительно ускоряет процесс работы и уменьшает количество вводимых ошибок.

Диалоги программных систем должны удовлетворяют следующему минимальному набору требований:

* время ответа системы на запрос, как правило, не должно превышать 4-5 секунды; при длительных вычислениях необходимы комментарии;
* ошибки пользователя не должны приводить к сбоям системы;
* изображение должно быть как можно более наглядным – во всех случаях желательна замена таблиц графиками, диаграммами, гистограммами, а черно-белого изображения – цветным;
* для повышения доверия к получаемым с помощью ПЭВМ результатам необходимо предусмотреть возможность эффективного контроля промежуточных результатов и при необходимости – изменения порядка вычислений;
* результаты работы должны отображаться в содержательных, привычных пользователю категориях;
* система должна позволять пользователю в любой момент перейти к другому виду работы, воспользоваться справочными таблицами.

Практика показала, что использование стандарта SAA сокращает время необходимое конечным пользователям для освоения приложения. Причем это касается как опытных, так и начинающих пользователей.

Временные характеристики выполняемых действий, прежде всего, определяют комфортность работы с программным продуктом. К показателям, описывающим реактивность системы, относят все временные характеристики, которые должны обеспечивать комфортную работу пользователя. Просчёт модели может занимать до нескольких минут в зависимости от класса ПЭВМ и количества заданных параметров.

На практике осуществляют различные виды защиты данных: защита данных от разрушений, некорректных обновлений, несанкционированного доступа и др.

Защита данных от разрушения (имеется в виду потеря данных при сбоях) – этот вид защиты называется обеспечение физической целостности данных и обеспечивается созданием копий и архивов данных.

Защита данных от некорректных обновлений (обеспечение логической целостности, основывается на предупреждении использования данных не основными пользователями и обеспечивается путем ограничения доступа к данным с помощью программных фильтров). Второй способ обеспечения логической целостности состоит в том, что пользователю выдается не вся запись, а только ее часть, необходимая для использования. Третий способ защиты заключается в проверки новых значений на принадлежность его значений области допустимых значений.

Защита данных от несанкционированного доступа, обеспечивающаяся путем создания программных фильтров, использованием замков доступа (пароли) и введением системы кодирования.

В связи с тем, что программное обеспечение содержит экономические данные, изменение которых могут привести к ошибочным расчетам, то доступ к ним должен быть защищён. Для решения этой задачи на запуск программной системы установлен пароль.

Одним из важнейших характеристик качества программного изделия является надежность.

Надежность – это свойство программного изделия сохранять работоспособность в течение определенного периода времени, в определенных условиях эксплуатации с учетом последствий для пользователя каждого отказа.

Работоспособным называется такое состояние программного изделия, при котором оно способно выполнять заданные функции с параметрами, с установленными требованиями технического задания. С переходом программного изделия в неработоспособное состояние связано состояние событие отказа.

Причины отказа программного изделия и технических систем различны. Если для технических систем причиной отказа может быть физический износ узлов и деталей, то программные изделия физическому износу не подвержены. Моральный износ, характерный для программного изделия, не может быть причиной нарушения работоспособности.

Причиной отказа программного изделия является невозможность его полной проверки в процессе тестирования и испытаний. При эксплуатации программного изделия в реальных условиях может возникнуть такая комбинация входных данных, которая вызывает отказ. Таким образом, работоспособность программного изделия зависит от входной информации, и чем меньше эта зависимость, тем выше уровень надежности.

В спроектированном программном обеспечении были реализованы функции, повышающие общую надежность системы.

Обработка исключительных ситуаций. Данная функция является наиболее значимой, поскольку как сама природа численных методов, так и действия пользователя предполагают возможность возникновения некорректных ситуаций. К примеру, наиболее распространенной, при проведении вычислений, является попытка деления на ноль. Основная проблема здесь заключается в том, что при возникновении подобных ситуаций происходит автоматический выход из текущего блока программы и операторы, стоящие за местом возникновения ошибки, которые, при нормальном выполнении программы должны были бы выполниться, не выполняются. Результатом этого может являться непредсказуемость дальнейшей работы программы. Для избежания подобных последствий, критические участки программы были включены в защищаемый блок кода. Данный блок характеризуется тем, что он позволяет либо соответствующим образом обработать исключительную ситуацию, либо выполнить завершающие действия для обеспечения ликвидации последствий, вызванных данной ошибкой.

Обеспечено блокирование недоступных в данный момент действий пользователя. Это проявляется в реализации системы меню и других элементов управления. Общая схема данного принципа заключается в том, что в каждый момент времени отслеживается возможность выполнения той или иной команды. Если выполнение команды возможно – то соответствующий элемент управления является доступным, и наоборот.

Обеспечена автоматическая проверка корректности данных, введённых пользователем. В случае, когда пользователь вводит неверные данные, например, точку вместо запятой в качестве десятичного разделителя, или литеру вместо цифры, программа сообщает ему об этом и предлагает повторить ввод.

Следует отметить, что часть функций обеспечения надежности системы реализуется средствами Delphi, поскольку он обеспечивает механизм стандартной обработки исключительных ситуаций.

2.4 Элементы интерфейса информационной системы «Начисление заработной платы сотрудникам средне-специальных учебных заведений»

При запуске программы пользователю предлагается указать пароль, окно пароля представлено на рисунке 6.



Рисунок 6. Окно пароля

При правильном вводе пароля (del) происходит загрузка программы, открывается главное окно приложения (рисунок 7).

Окно состоит из пяти объектов:

1. Кадры;
2. Расчет;
3. Справка, при щелчке мышью открывается справочная система. Здесь можно получить информацию о методах эксплуатации программы.
4. Архив, при щелчке мышью открывается таблица начислений за год. Здесь можно получить информацию о начислениях по конкретному сотруднику.
5. Выход, при нажатии кнопки в левом нижнем углу с пиктограммой стрелки происходит выход из программы.

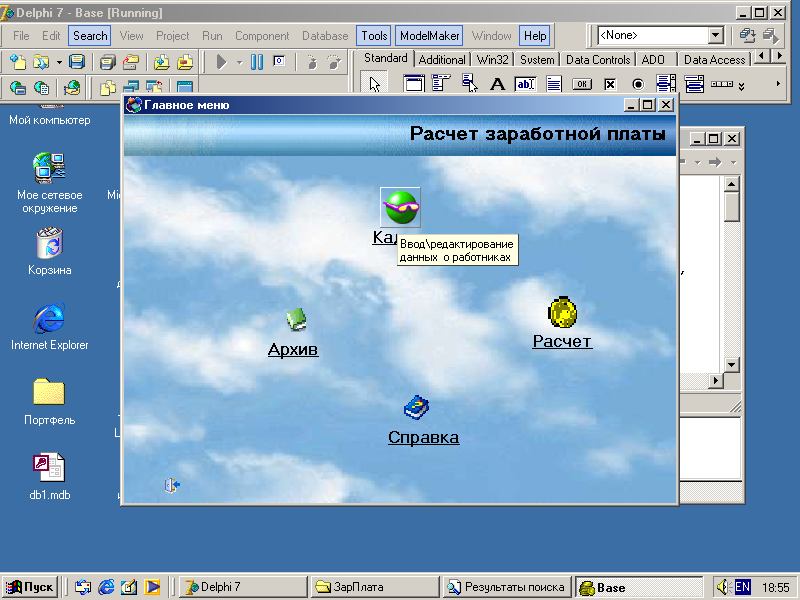


Рисунок 7. Главное окно

При щелчке мышью кнопке «Кадры» открывается окно «Окно категорий ввода и редактирования данных о сотруднике» (рисунок 8).

Данное окно содержит четыре объекта:

* Ввод, при щелчке мышью открывается окно «Ввода и редактирования кадровых данных» (рисунок 9).

Данное окно обладает всеми условиями для правильного ввода данных о принятом на работу сотруднике. Ввод данных о сотруднике может проводиться как в таблицах, так и с помощью диалоговых окон. На форме имеются кнопки «Кадры…» и «Должность…», при нажатии которых вызываются диалоговые окна «Ввода личных данных сотрудника» (рисунок 10) и «Ввода должностных данных» (рисунок 11). Также здесь организован поиск необходимой информации о сотруднике по его фамилии. Опции «без учета регистра» и «по части строки» производят поиск по части фамилии и без учета заглавных букв и меню в котором предусмотрены операции вставки новой записи и удаление существующих записей, сохранение и отмена изменении, обновление, навигация по данным, с помощью меню «Окно» можно вызвать необходимое диалоговое окно.

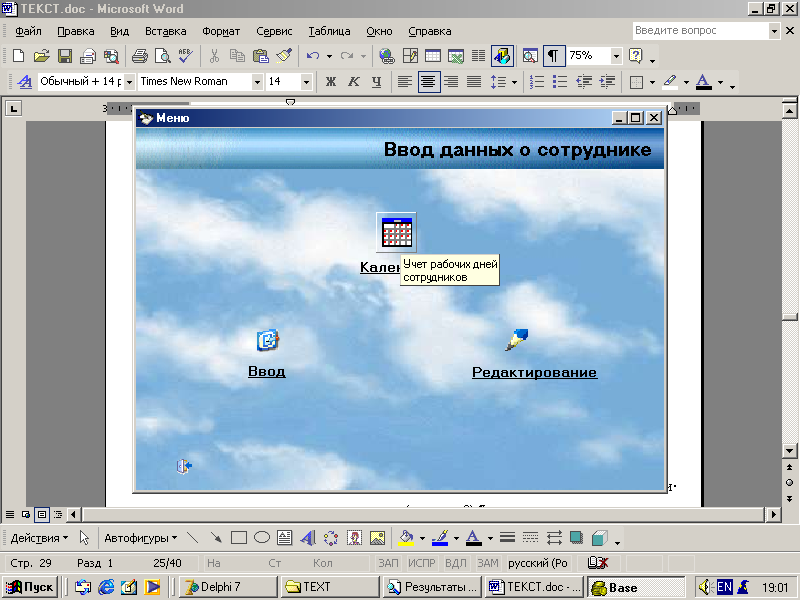
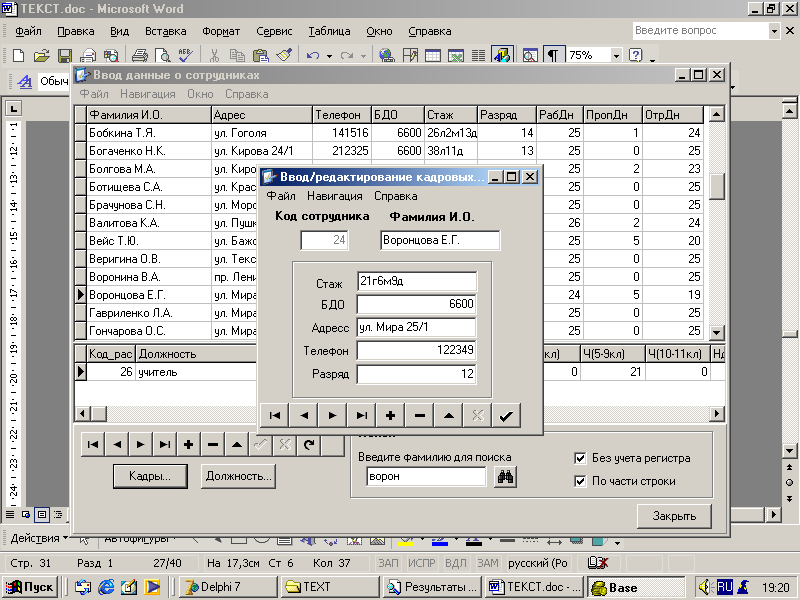


Рисунок 8. Окно «Категорий ввода и редактирования данных о сотрудниках»

Для ввода данных о принятом сотруднике необходимо нажать кнопку с пиктограммой «+», окно ввода станет пустым. Ввод начинается с Фамилии И.О., код сотруднику присваивается автоматически. Перемещения по объектам ввода осуществляется с помощью клавиши Tab или мыши. После заполнения всех данных необходимо нажать на кнопку с пиктограммой .



Ввод должностных данных о сотруднике может производиться как в таблице, так и через форму «Должности», должность и преподаваемый предмет выбираются в соответствующих списках, далее указываются G (группа) коэффициент сотрудника и коэффициент обязательных пенсионных взносов. В случае если сотрудник имеет несколько должностей, то опция «расчет» определяет по каким данным, будут производиться расчеты. В списке «ставка» указывается коэффициент ставки. Для учителей ставка при оплате не учитывается, но учитывается при начислениях, следовательно, коэффициент ставки равен нулю, а для остальных он равен от 0,5 и далее. После заполнения всех данных необходимо нажать кнопку .

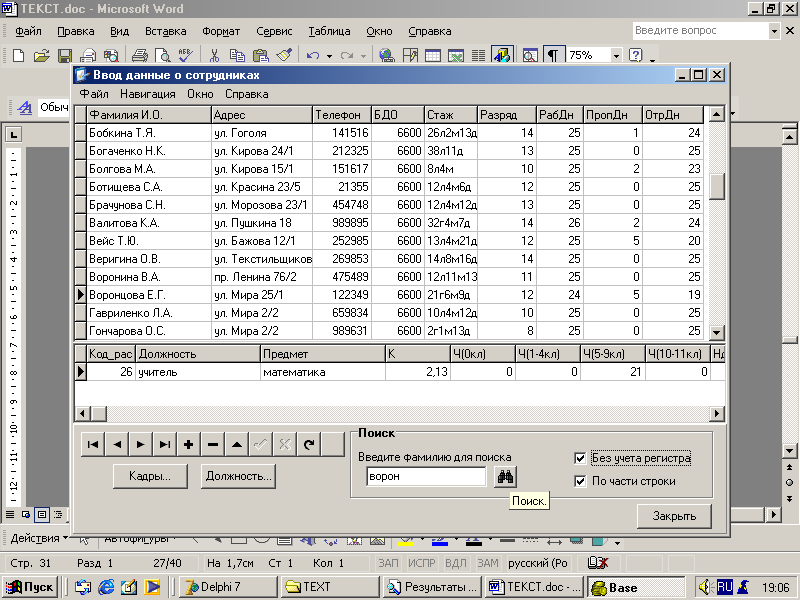
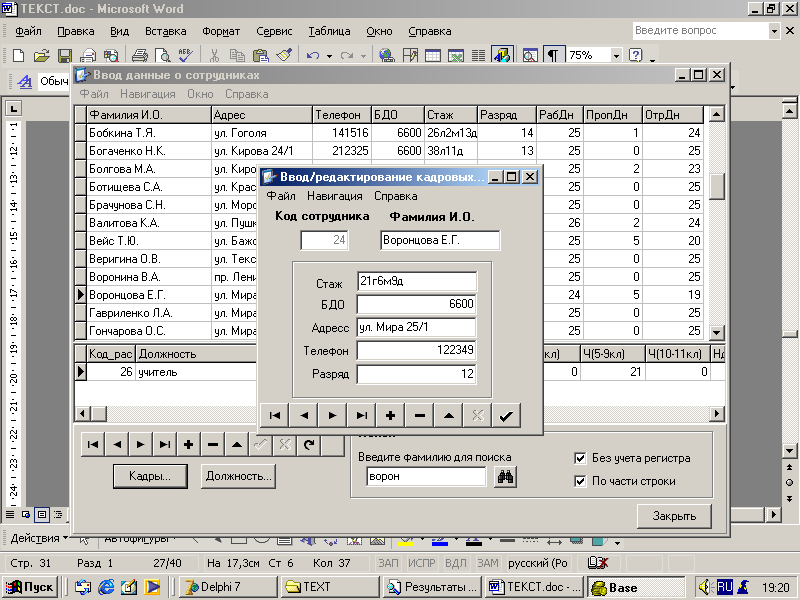


Рисунок 9. Окно ввода данных о сотруднике

Если сотрудник занимает более одной должности или преподает более одного предмета, то необходимо вызвать повторно окно программы «Должности» и повторить действия. Вся информации о сотруднике отобразится подчиненной таблице окна «Ввод данных».

Редактирование, при щелчке мышью открывается окно программы «редактирование академических часов по классам» (рисунок 12).

В данном окне для удобства редактирования записей предусмотрены дополнительные диалоговые окна:

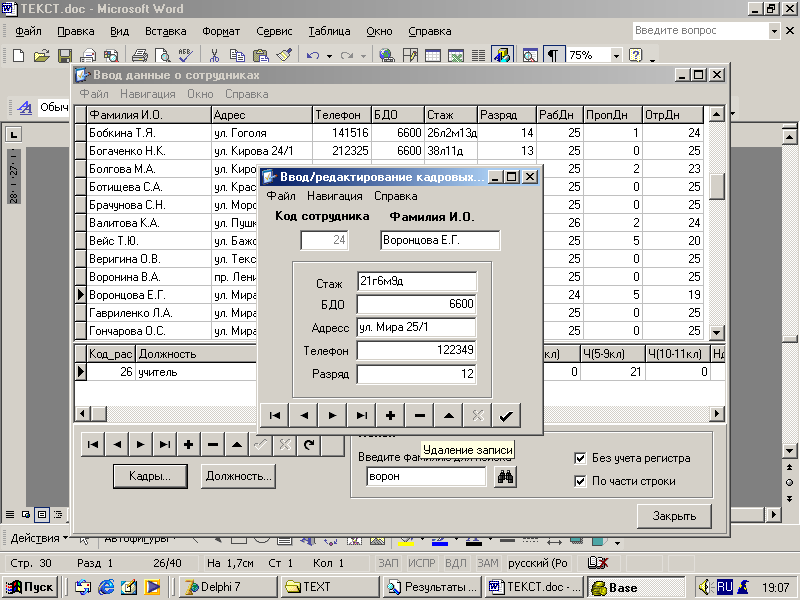


Рисунок 10. Окно ввода личных данных сотрудника

При щелчке на кнопке «Должности» появляется окно для редактирования должностных данных (рисунок 13).

При щелчке на кнопке «Часы» появляется окно для изменения количества часов преподаваемого предмета (рисунок 14).

Учет рабочих дней и пропущенных дней производятся с помощью вызова диалогового окна программы «Учет рабочих дней», при нажатии кнопки столбца таблицы «РабДн».

Также в данном окне программы организован поиск работников по Фамилии. Установлений опций «без учета регистра» позволяет вводить фамилию без заглавных букв, опция «по части строки», позволяет вводить только часть фамилии. При указании обеих опций поиск производиться по двум параметрам одновременно.

Окно программы «Сведения о должностях» открывается при нажатии кнопки «Должности». Окно программы делиться на две области, если работник занимает более одной должности или преподает боле одного предмета. Данное диалоговое окно гарантирует безошибочный ввод данных о сотруднике.

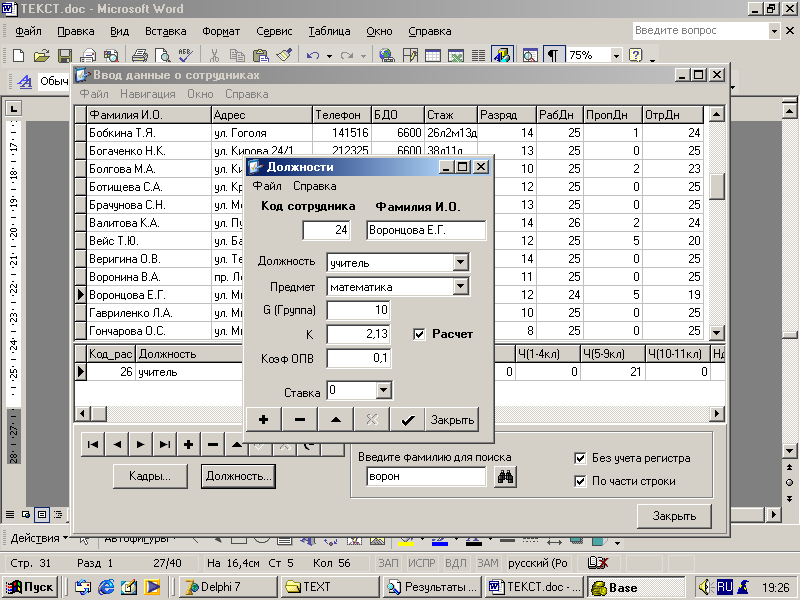


Рисунок 11. Окно ввода должностных данных сотрудника

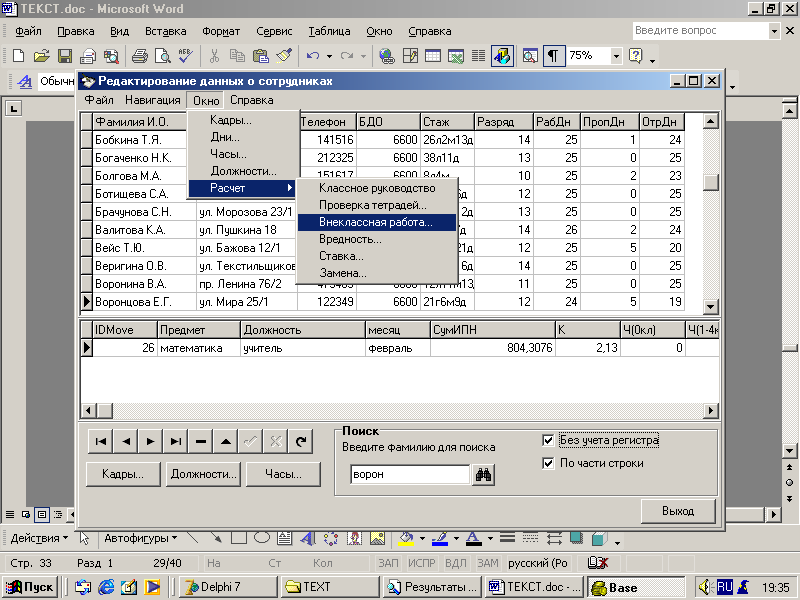


Рисунок 12. Окно редактирования данных

Окно программы «Редактирования данных» (рисунок 13) наглядно отображает все должности и предметы, которые соответствуют данному сотруднику. В данном окне удобно производить соответствующие изменения данных о сотруднике.

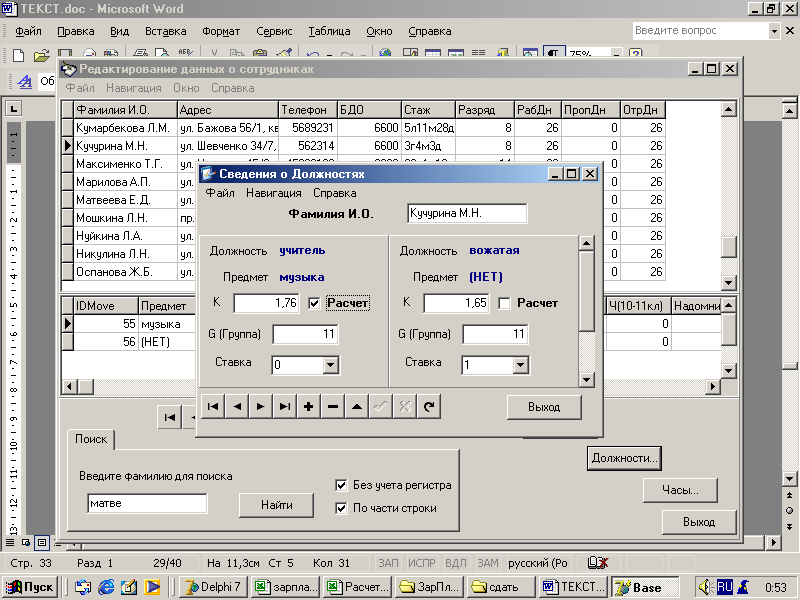


Рисунок 13. Окно редактирования должностных данных

Окно программы «Распределение учебных часов» (рисунок 14) наглядно отображает все предметы, которые соответствуют данному сотруднику. В данном окне удобно производить соответствующие изменения академических часов, закрепленные за данным учителем.

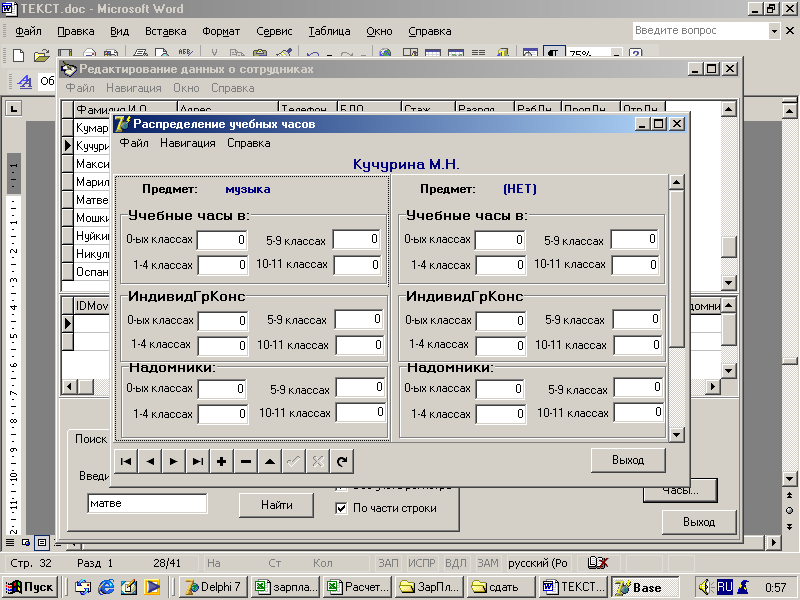


Рисунок 14. Окно редактирования данных

Календарь, при щелчке мышью открывается окно программы «Учет рабочих дней» (рисунок 15).

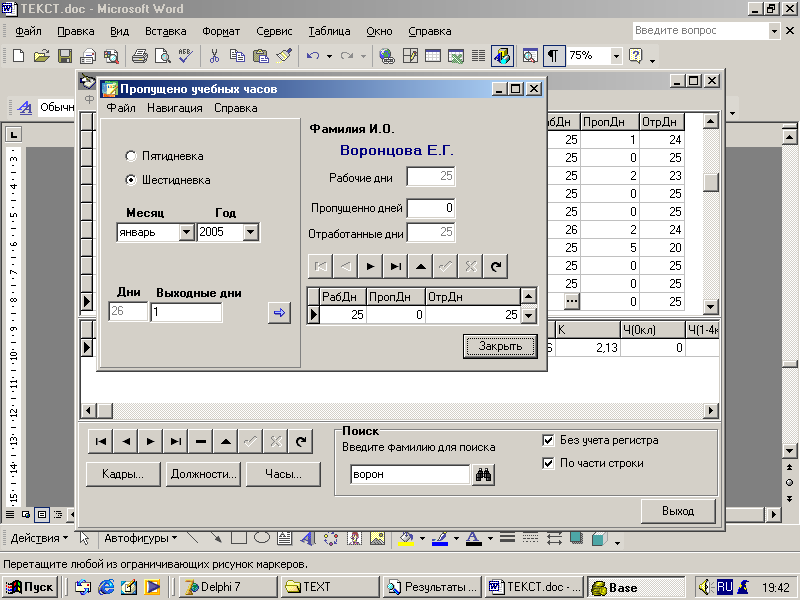
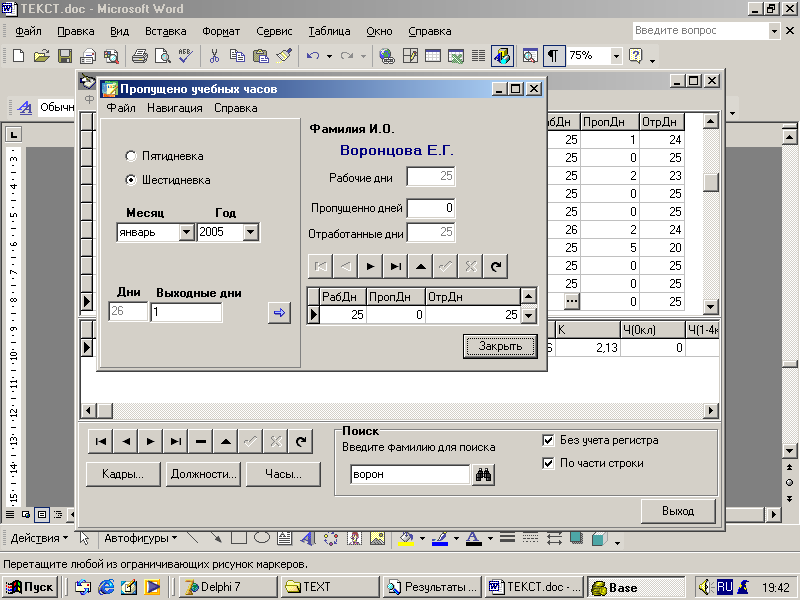


Рисунок 15. Окно программы «Учет рабочих дней»

Если никаких изменений данных о сотрудниках не производиться, то для того чтобы произвести учет рабочих дней сотрудника вовсе не обязательно открывать окно программы «Редактирование». Окно программы «Учет рабочих дней» можно вызвать из окна категорий или любого окна программы из меню «Окно». Оно обеспечивает бухгалтеру быстрый ввод рабочих и пропущенных дней сотрудника. Учет рабочих дней состоит из нескольких этапов:

* Устанавливается одна из опций «Пятидневка» или «Шестидневка».
* Затем устанавливается текущий год и месяц.

В объекте «Дни» автоматически в зависимости от установленного месяца и года отображается количество дней в месяце без воскресенья. В объекте «Выходные дни» устанавливаются праздничные дни, относящиеся к выходным. После введения всех параметров необходимо нажать кнопку с пиктограммой . Нажатие этой кнопки делает активной правую часть окна программы. Автоматически в объекты «рабочие дни» и «отработанные дни» помещается количество рабочих дней в месяце и отработанные дни сотрудником.



Если сотрудник по каким-либо причинам пропустил рабочий день, то это фиксируется в объекте «Пропущенные дни» и автоматически происходит изменение значения отработанных дней.

В данном окне все данные левой части окна программы считаются общими для всех сотрудников, их устанавливают однажды, а правая часть окна индивидуальна для каждого сотрудника. Пропущенные дни устанавливаются для каждого сотрудника отдельно. Навигация по записям о сотрудниках происходит с помощью навигатора или поиска, который можно вызвать из меню «Файл».

При щелчке мышью по кнопке «Расчет» открывается окно программы «Категории расчета» (рисунок 16).

Данное окно состоит из девяти объектов, при нажатии на которые выходят диалоговые окна программы, в которых указывается коэффициент расчета по соответствующей категории расчета. Навигация по записям производится с помощью навигатора или поиска, вызываемы через меню «Окно».

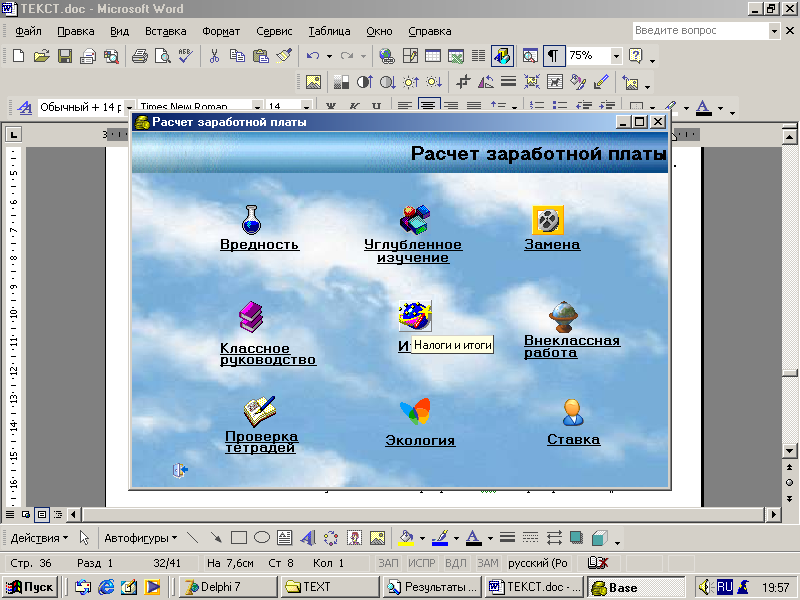


Рисунок 16. Окно программы «Категории расчета»

Категории расчета:

Вредность, Вредность рассчитывается работникам, которые в ходе своей деятельности наносят вред своему здоровью. Строка «Коэффициент» становится активной только для учителей химии и ИВТ (рисунок 17).

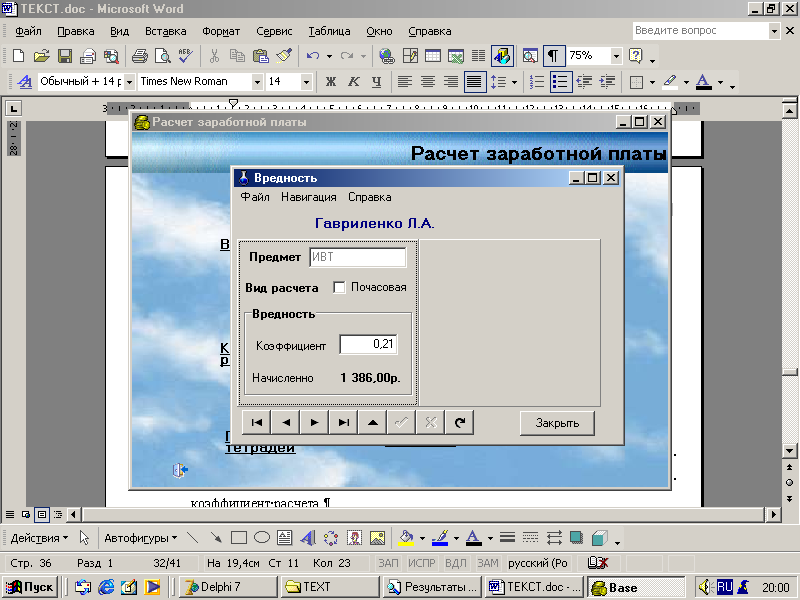


Рисунок 17. Окно программы расчета вредности

Классное руководство, при щелчке мышью открывается окно программы расчета классного руководства (рисунок 18), в котором указывается коэффициент расчета.

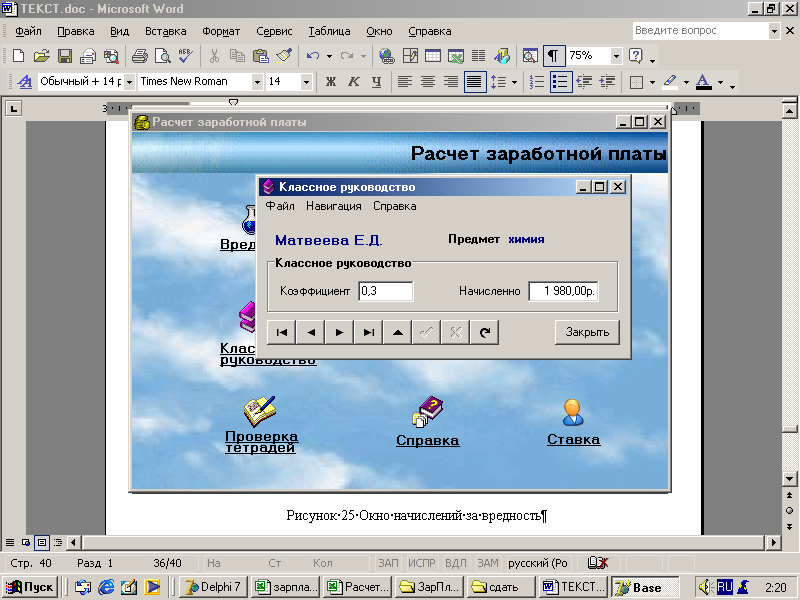


Рисунок 18. Окно программы расчета классного руководства

Проверка тетрадей, при щелчке мышью открывается окно программы расчета классного руководства (рисунок 19), в котором указывается коэффициент расчета.

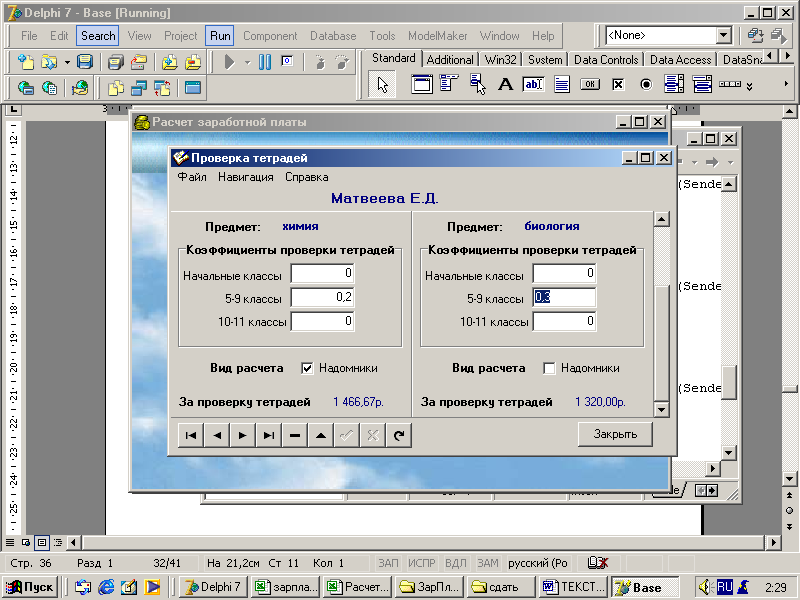


Рисунок 19. Окно программы расчета проверки тетрадей

Расчет проверки тетрадей происходит в зависимости от количества академических часов и от коэффициента расчета.

Замена (рисунок 20), в левой части окна указываются часы замены, а в правой части подсчитывается начисления.

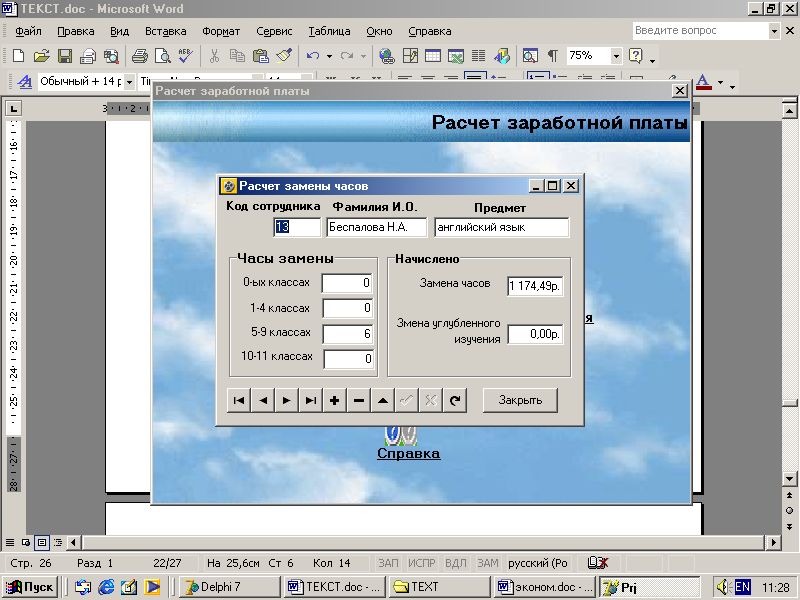


Рисунок 20. Окно расчета «Замены часов»

Внеклассная работа (рисунок 21), в данном окне указывается расчетный коэффициент внеклассной работы.

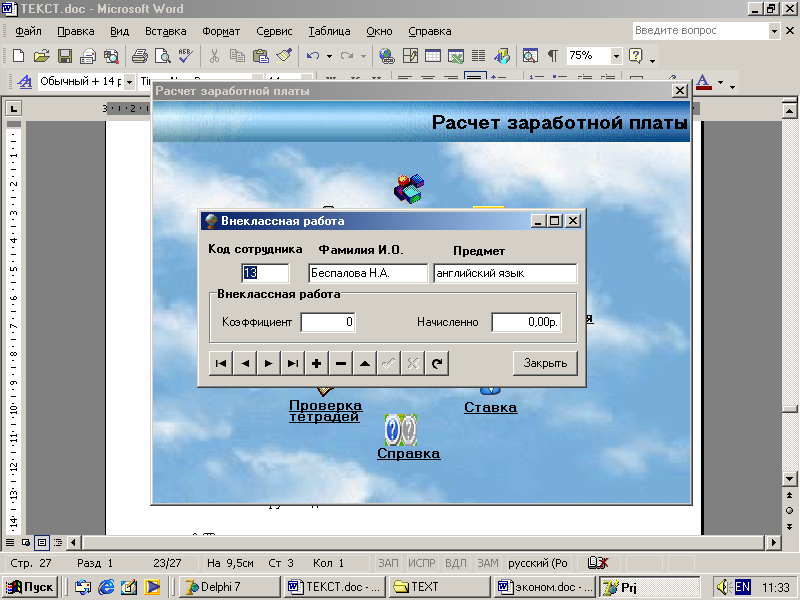


Рисунок 21. Окно расчета за «Внеклассную работу»

Экология (рисунок 22), в данном окне рассчитывается экологические надбавки.

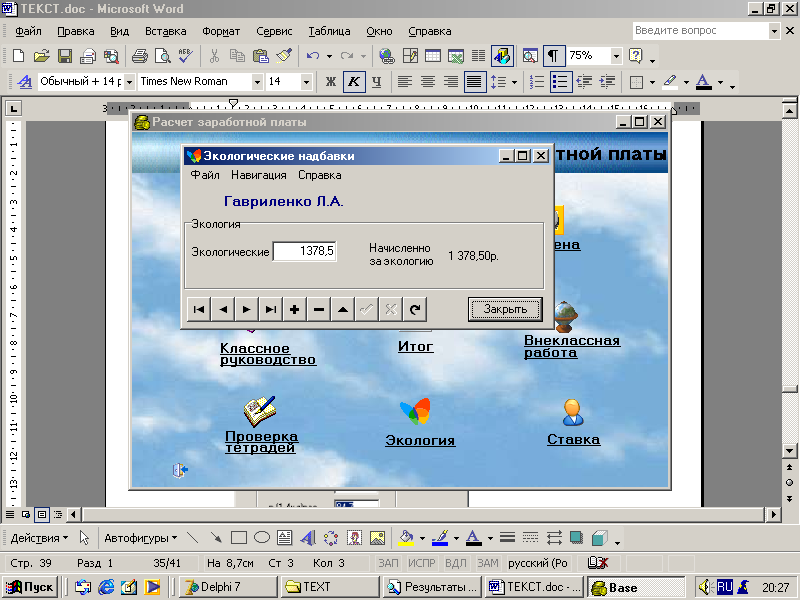


Рисунок 22. Окно расчета экологических надбавок

Углубленное изучение (рисунок 23).

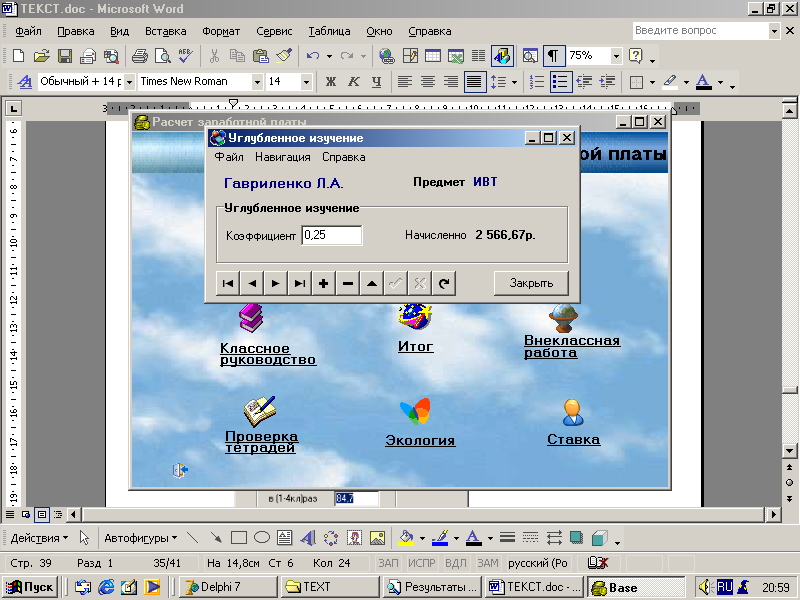


Рисунок 23. Окно расчета экологических надбавок

Ставка (рисунок 24), при открытии данного диалогового окна можно получить сведения о расчетной ставке сотрудника, по которой производятся расчеты заработной платы и ставке, которая учитывается при выплате заработной платы.

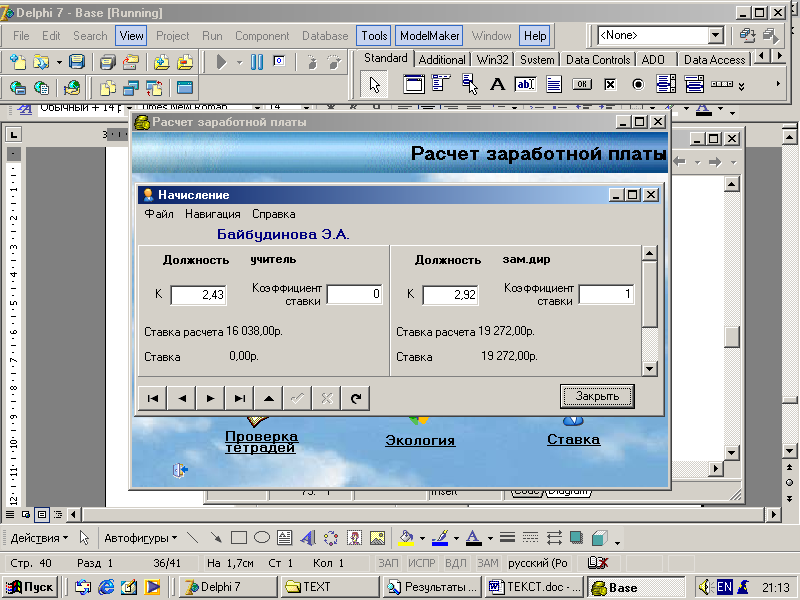
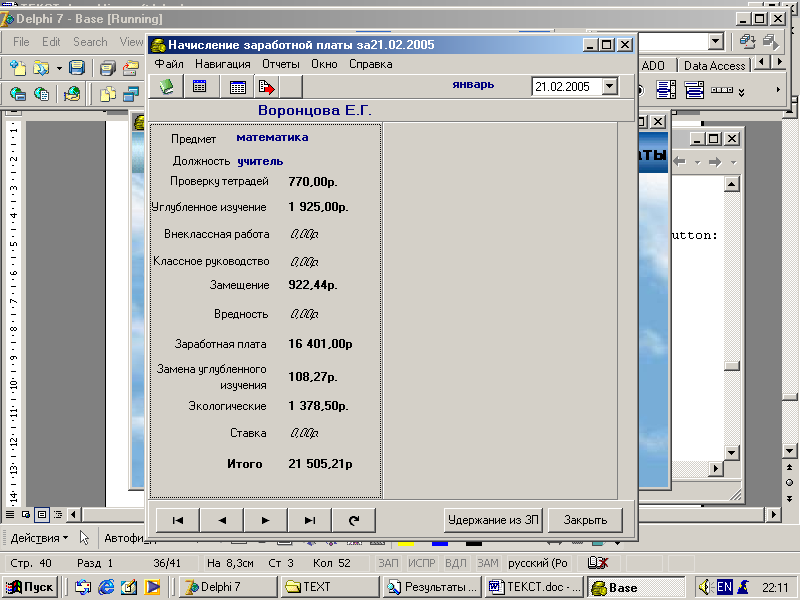
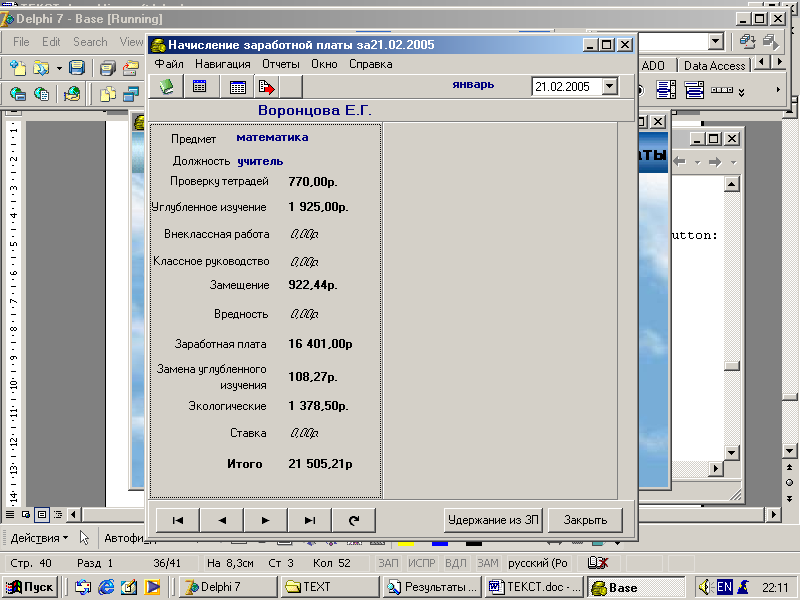
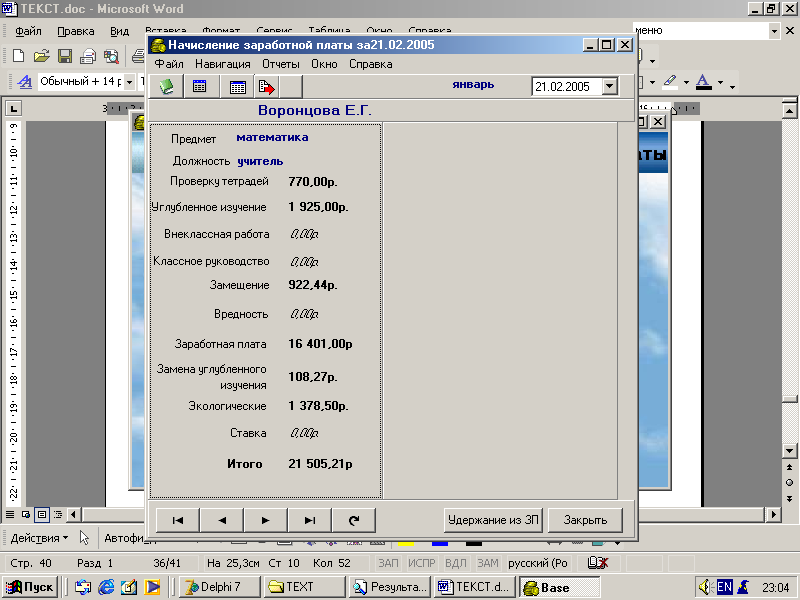


Рисунок 24. Окно расчета ставки

При нажатии кноп «Итог» откроется диалоговое окно «Начисление заработной платы за текущий месяц» (рисунок 25).

Данное диалоговое окно предоставляет сведения о произведенных начислениях. С помощью кнопки производится вставка записи в таблицу «Год», где хранятся все начисления за проработанные года. Кнопки «Таблица месяц» и «Таблица год» позволяют просмотреть данные о начислениях за текущий месяц и проработанные года соответственно. При нажатии кнопки или кнопки «Удержание из ЗП» открывается окно программы (рисунок 26), в котором отображаются начисления сотрудника с начала года и доходы, не подлежащие налогообложению.



В левой части окна программы отображаются доходы начисленные за текущий месяц, а в правой части указываются доходы не подлежащие налогообложению. При щелчке по кнопке «Данные» отображаются данные начисленных доходов и доходов, не подлежащих налогообложению с начала года.

При нажатии кнопки «Далее…» открывается окно программы «Сумма ОПВ и налоговые вычеты» (рисунок 27).

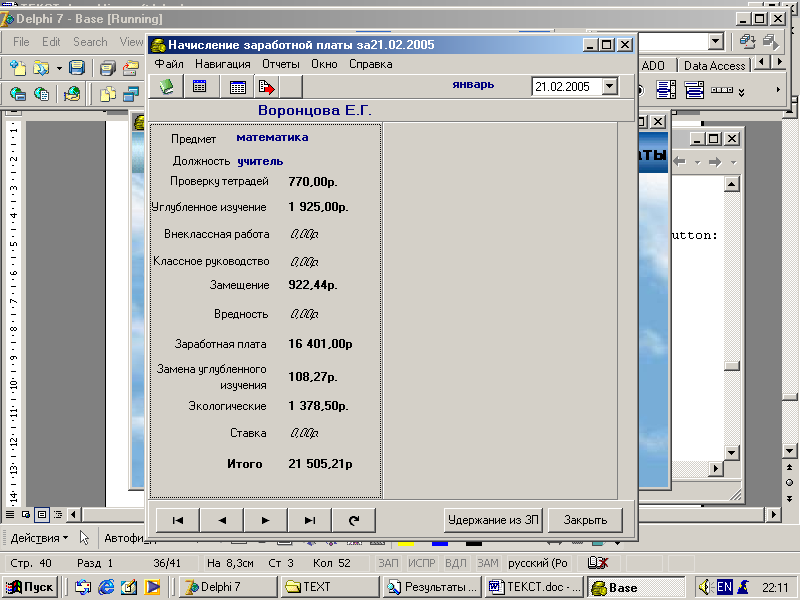


Рисунок 25. Окно программы «Начисление заработной платы за текущий месяц»

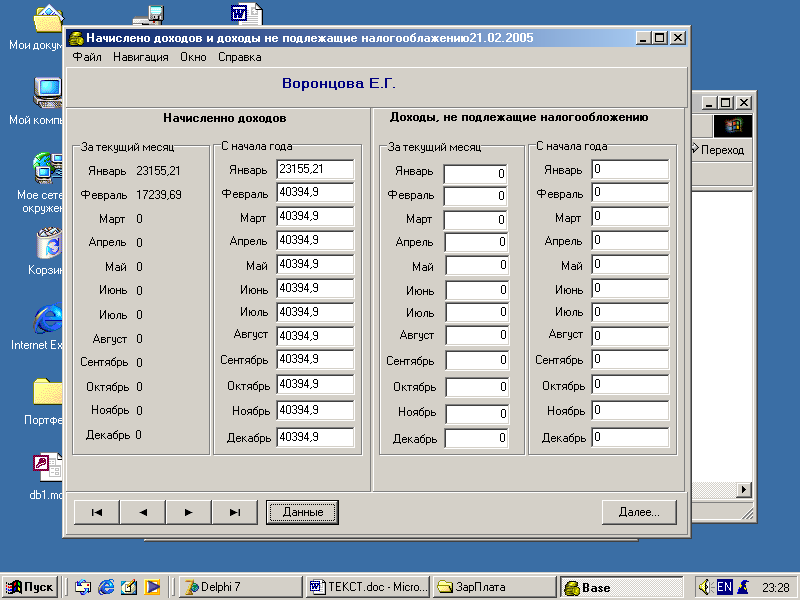


Рисунок 26 Окно программы «Начисление доходов»

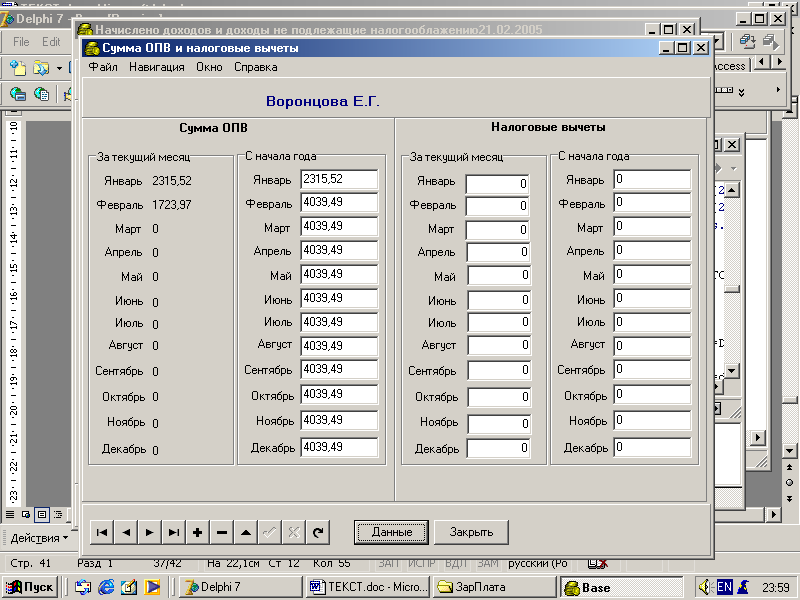


Рисунок 27. Окно программы «Сумма ОПВ и налоговые вычеты»

В левой части окна программы отображаются обязательные пенсионные взносы за текущий месяц в размере 10% от начисленной суммы, в правой части окна необходимо указать налоговые вычеты за текущий месяц. При щелчке по кнопке «Данные…» отображаются сумма ОПВ и налоговые вычеты с начала года.

С помощью меню «Окно» можно получить доступ к окнам просмотра по расчету индивидуального подоходного налога, социального налога и окну доходов, подлежащих к выплате.

Пункт «Налоги» меню «Окно» состоит из шести подпунктов:

Коэффициент перерасчета, при щелчке мышью открывается окно просмотра (рисунок 28). При нажатии кнопки «Данные» отображаются данные: сумма облагаемого ИПН дохода с начала года, расчетная сумма облагаемого ИПН дохода, сумма ИПН с расчетной суммы облагаемого ИПН доходов и коэффициент перерасчета.

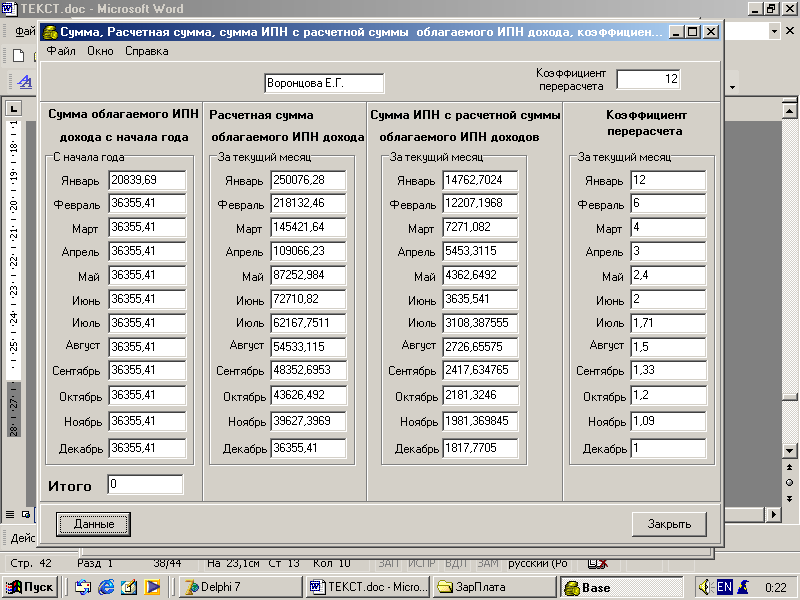
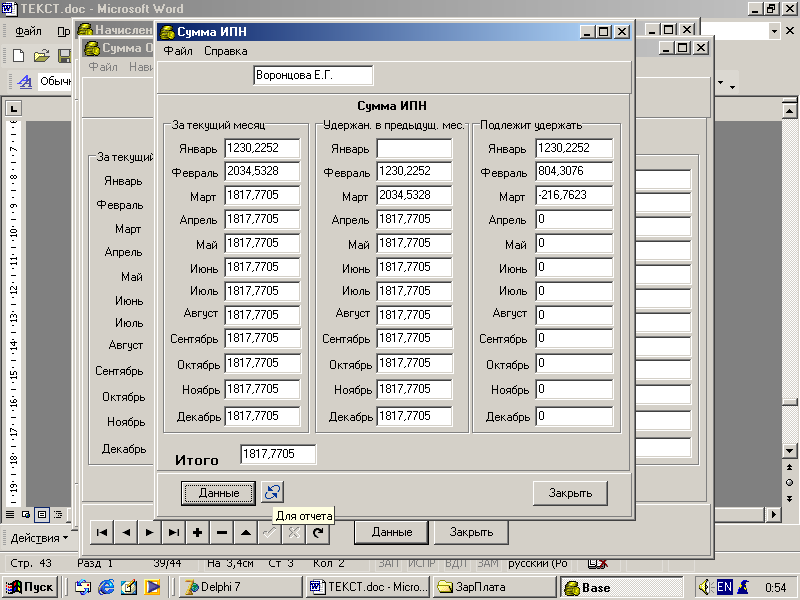


Рисунок 28. Окно просмотра

Сумма ИПН, при щелчке мышью открывается окно просмотра «Сумма ИПН» (рисунок 29). Кнопка «Данные» возвращает значения суммы ИПН за текущий месяц, удержанные в предыдущем месяце и подлежащие удержанию. Так как сумма ИПН подлежащая удержанию используется в отчетах, то ее необходимо сохранить в таблице «Месяц» с помощью кнопки .



Сумма СО, при щелчке мышью открывается окно просмотра «Сумма СО» (рисунок 30). Кнопка «Данные» возвращает значения суммы социального налога с расчетной суммы дохода, облагаемого социальным налогом и значения суммы социального налога за текущий месяц, удержанные в предыдущем месяце и подлежащие удержанию.

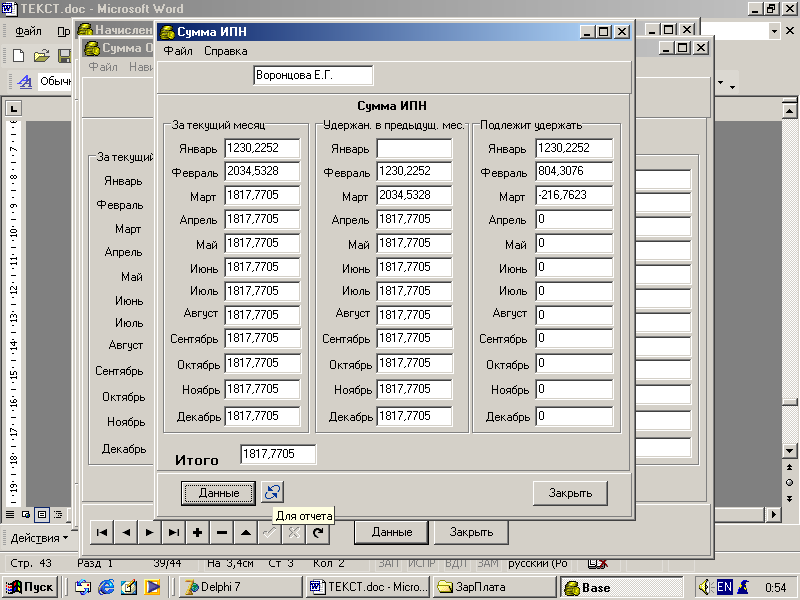


Рисунок 29. Окно просмотра «Сумма ИПН»

Социальные отчисления и социальный налог, при щелчке мышью открывается окно просмотра «Социальные отчисления и социальный налог» (рисунок 31). Кнопка «Данные» возвращает значения доходов физических лиц, с которых исчисляются социальные отчисления, суммы социальных отчислений и значения социального налога за минусом социальных отчислений с расчетной суммы дохода, облагаемого социальным налогом и значения суммы социального налога за текущий месяц, удержанные в предыдущем месяце и подлежащие удержанию.

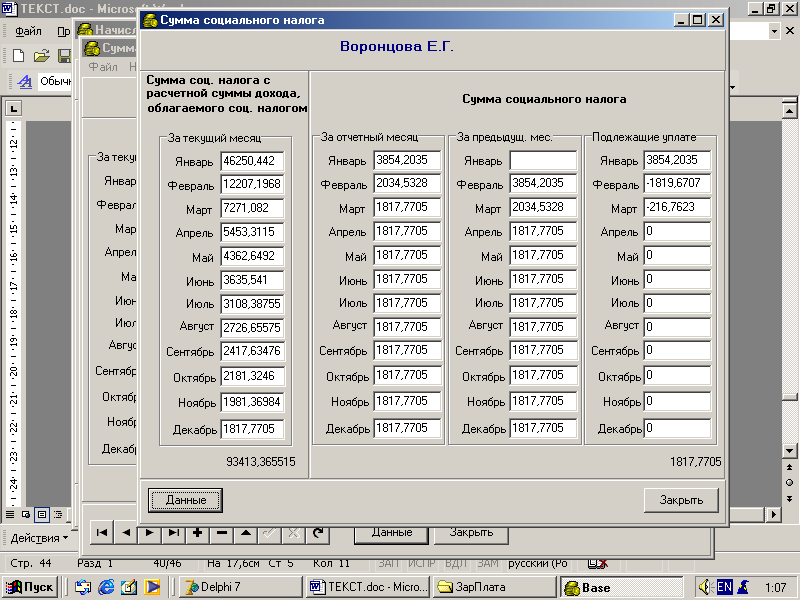


Рисунок 30. Окно просмотра «Сумма социальных отчислений»

Облагаемые и необлагаемые доходы, при щелчке мышью открывается окно просмотра (рисунок 32). Кнопка «Данные» возвращает значения доходов, не являющиеся объектом обложения социальным налогом, за исключением ОПВ, доходов, облагаемые социальным налогом и значения расчетной суммы дохода, облагаемого социальным налогом.

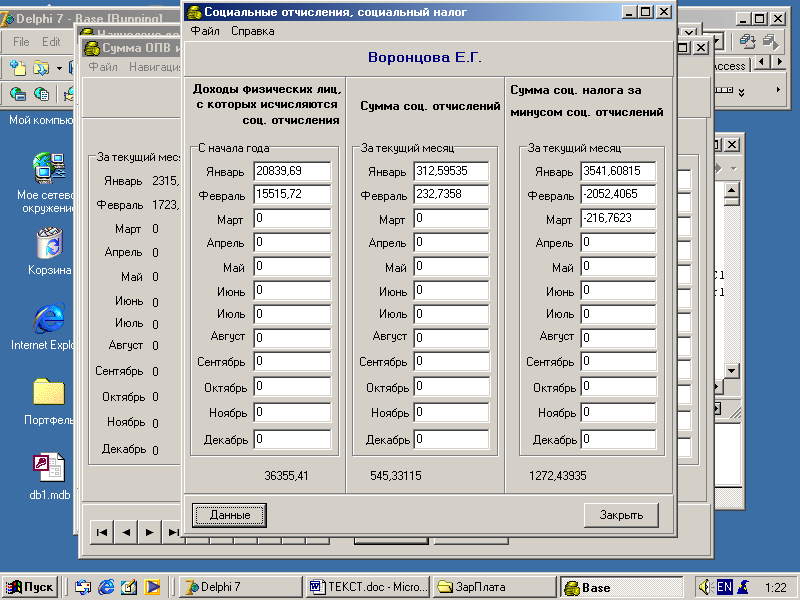


Рисунок 31. Окно просмотра «Социальные отчисления и социальный налог»

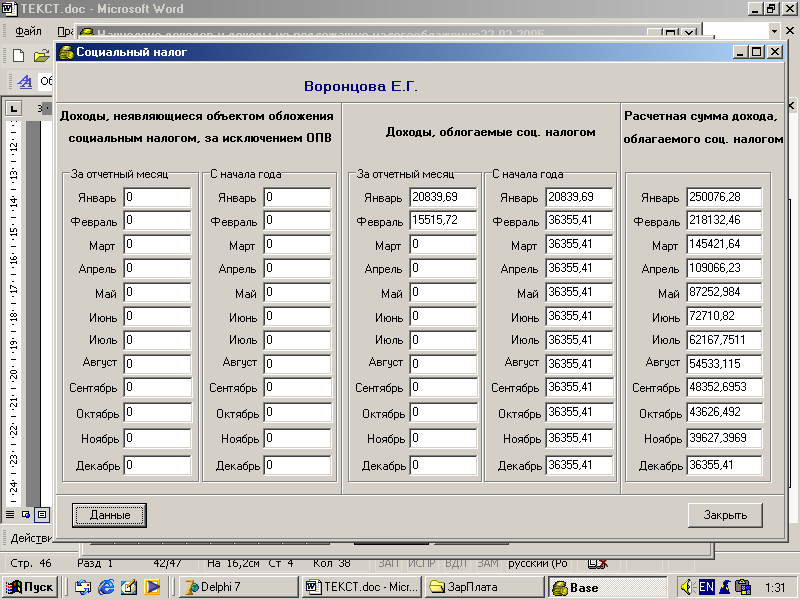


Рисунок 32. Окно просмотра облагаемых и необлагаемых доходов

К выдаче, при щелчке мышью открывается окно просмотра доходов, подлежащих к выплате (рисунок 33). Кнопка «Данные» возвращает значения начисленной суммы без учета ИПН и ОПВ за текущий месяц. Так как сумма, подлежащая к выплате используется в отчетах, то ее необходимо сохранить в таблице «Месяц» с помощью кнопки .

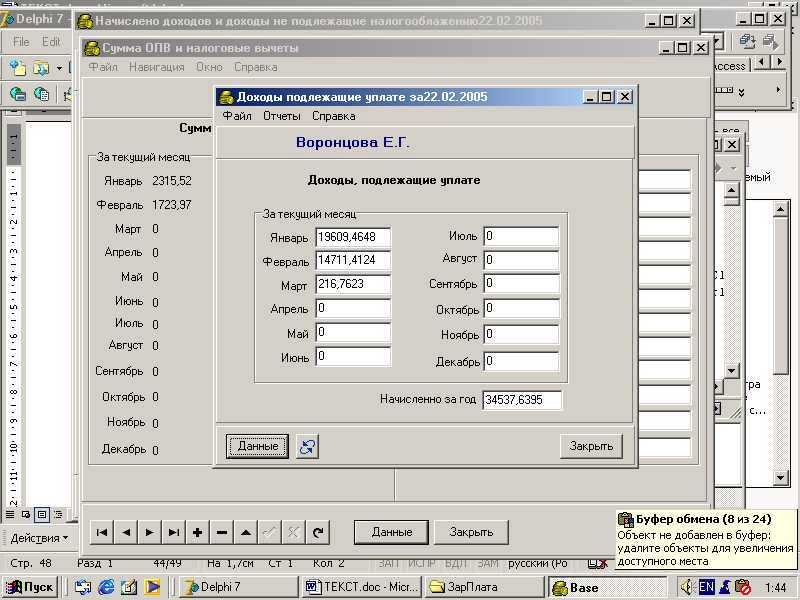
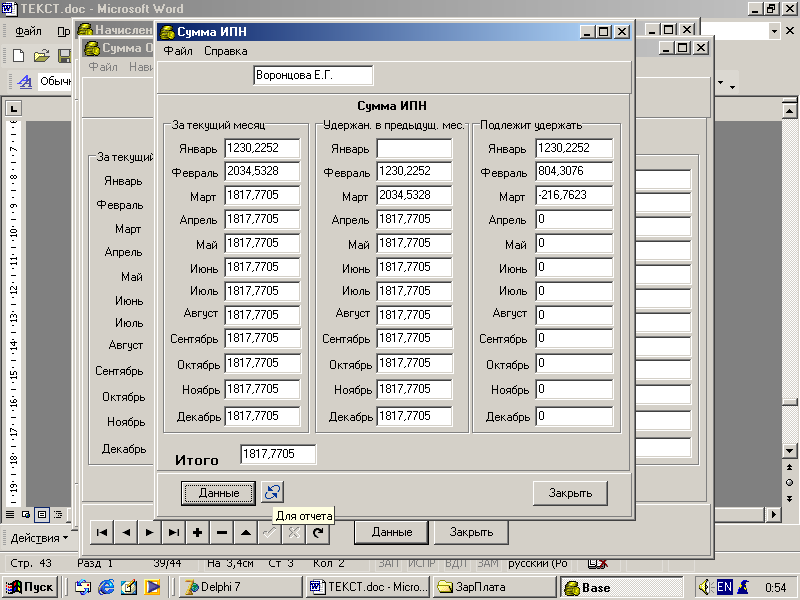


Рисунок 33. Окно просмотра доходов, подлежащих к выплате

С помощью меню «Отчеты» можно получить доступ к отчетам за налоговый период (рисунок 35) и к квиткам (рисунок 34)

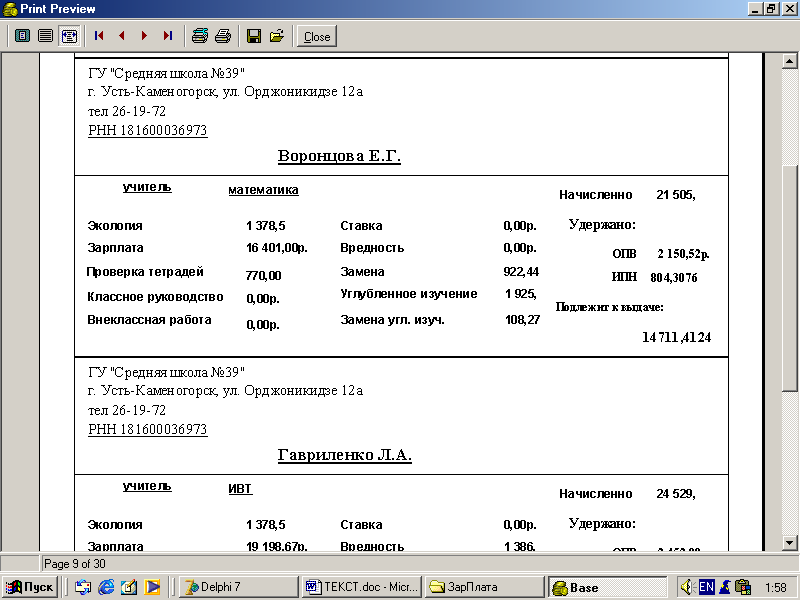


Рисунок 34 Квитки

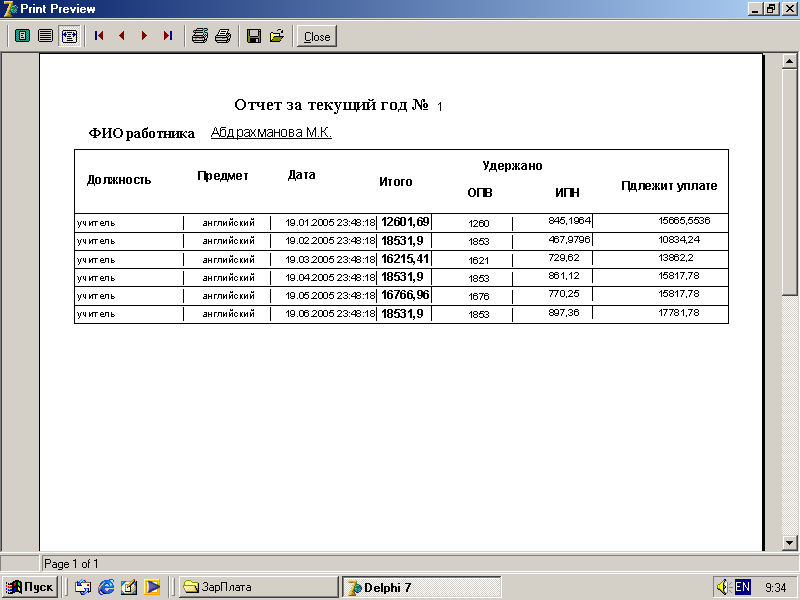


Рисунок 35. Отчет за налоговый период

Каждое из вышеперечисленных окон программы имеет меню, с помощью которого можно выполнять операции ввода, редактирования и удаления записей, навигацию по записям, получить доступ к отчетам, перемещаться по необходимым диалоговым окнам из любого активного окна программы.

В меню «Окно» находиться вызов окна программы общих коэффициентов (рисунок 36). Они используются при расчетах и являются одинаковыми для всех сотрудников. При изменении коэффициентов расчета их достаточно изменить в соответствующих объектах и сохранить эти изменения. Эти изменения будут равносильны для каждого работника школы.

В меню любого диалогового окна программы присутствует пункт «Справка», через которое организовывается вызов справки. Здесь можно получить необходимую подробную информацию об эксплуатации программы (рисунок 37).

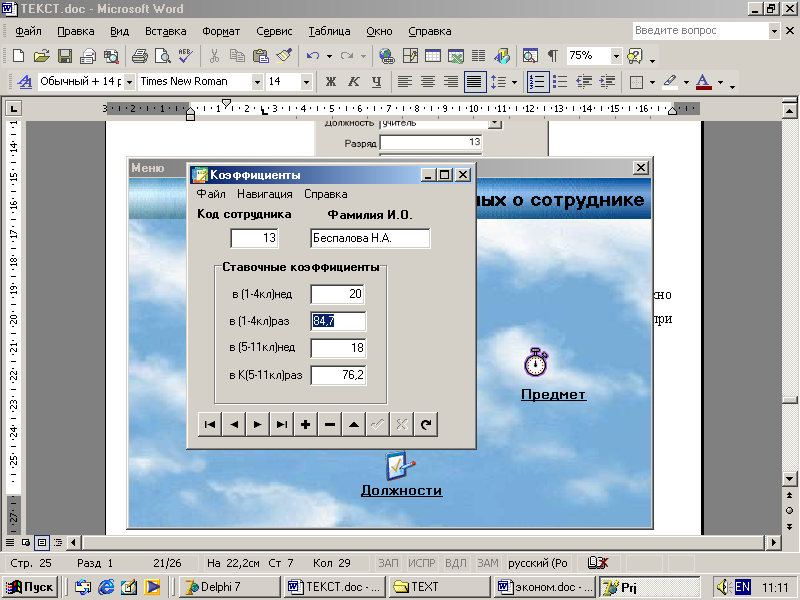


Рисунок 36. Окно ввода расчетных коэффициентов

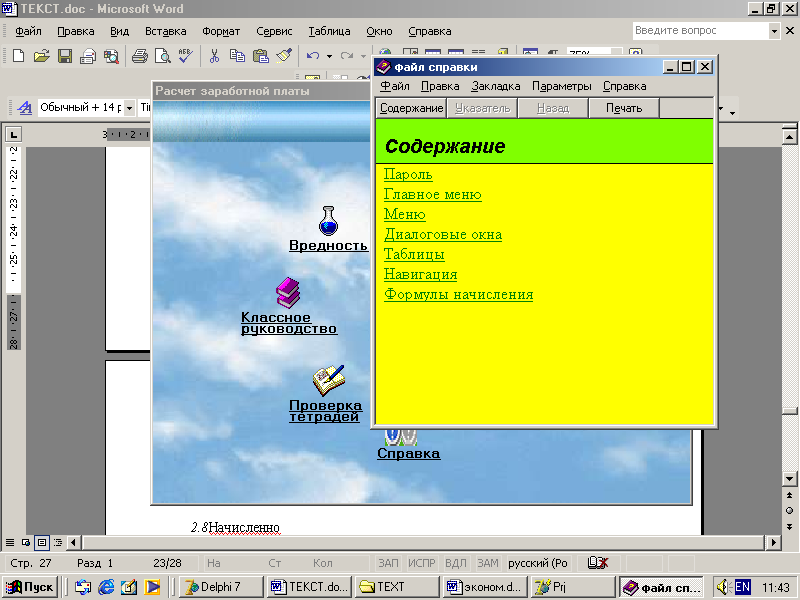


Рисунок 37 Окно справки

Формулы, используемые в данной курсовой работе для начисления заработной платы учителям школы:

1) расчетная ставка

Ставка = БДО\*К

где базовый должностной оклад (БДО) у всех сотрудников одинаковый, а коэффициент (К) зависит от разряда;

2) экологические надбавки

Экология = 1378.5\*



3) вредность оплачивается учителям Химии и ИВТ за нанесенный вред здоровью в ходе уроков

Вредность = БДО \* Надбавочный коэффициент



4) внеклассная работа

Внеклассная работа = БДО \* Надбавочный коэффициент\*



5) классное руководство

Классное руководство в начальных классах = БДО \* Надбавочный коэффициент\*



Классное руководство в старших классах = БДО \* Надбавочный коэффициент\*



Для определения оплаты часов замены необходимо вычислить размер оплаты разового и недельного часа в начальных классах и старших классах.

6) недельные и разовые часы оплаты

Разовый час = ;



Недельный час = ;



Разовый час = ;



Недельный час = ;



8) замена углубленного изучения

Замена углубленного изучения = Часы замены в 1-4 классах \* надбавочный коэффициент\*\*



Замена углубленного изучения = Часы замены в 5-11 класса\* надбавочный коэффициент\*\*



9) Оплата заменяемых часов в начальных классах

Замена = часы замены в 0-4 классах \* разовый час1-4класса+замена углубленного изучения в начальных классах;

10) Оплата заменяемых часов в старших классах

Замена = часы замены в 5-11 классах \* разовый час5-11класса+замена углубленного изучения в старших классах;

11) Углубленное изучение

Углубленное изучение в начальных классах = количество часов + часы индивидуальной групповой консультации \* надбавочный коэффициент\*\*



Углубленное изучение в старших классах = количество часов + часы индивидуальной групповой консультации в 5-11 классах + Надомное изучение в 5-11 классах\* надбавочный коэффициент\*\*



12) проверка тетрадей

Проверка тетрадей в подготовительных классах = БДО \*надбавочный коэффициент\*



Проверка тетрадей в начальных классах =количество часов в 1-4 кл. \* процентный показатель\* \*\*



Проверка тетрадей в старших классах = количество часов в 5-11 классах \* процентный показатель\*\*



13) заработная плата начисляется в зависимости от нагрузки

Заработная плата в начальных классах = общее количество часов + индивидуальные групповые консультации в 1-4 классах\* Недельный час в 1-4 классах.\*



Заработная плата в старших классах = общее количество часов + +Надомное изучение в 5-11 классах\* индивидуальные групповые консультации в 5-11 классах\* Недельный час в 5-11 классах\*



14) общая заработная плата в месяц учителей школ

Общая заработная плата в месяц учителей школ = Проверка тетрадей + Заработная плата + Углубленное изучение + Замещение + Экологические надбавки + Вредность + Внеклассная работа + классное руководство;

Формулы удержания из заработной платы налоговых отчислений:

15) начислено доходов с начала года

Начислено доходов с начала года = доходы, начисленные в текущем месяце + доходы, начисленные в предыдущем месяце с начала года;

16) доходы, не подлежащие налогообложению с начала года

Доходы, не подлежащие налогообложению с начала года = доходы, не подлежащие налогообложению в текущем месяце + доходы, не подлежащие налогообложению в предыдущем месяце с начала года;

17) обязательные пенсионные взносы за текущий месяц

Обязательные пенсионные взносы за текущий месяц = Общая заработная плата за текущий месяц / процент ОПВ (10%);

18) сумма Обязательных пенсионных взносов

Сумма ОПВ = Обязательные пенсионные взносы за текущий месяц + сумма обязательных пенсионных взносов в предыдущем месяц с начала года;

19) налоговые вычеты с начала года

Налоговые вычеты с начала года = Налоговые вычеты за текущий месяц + Налоговые вычеты в предыдущем месяце с начала года;

20) сумма облагаемого ИПН дохода с начала года

Сумма облагаемого ИПН дохода с начала года = Начислено доходов с начала года - Доходы, не подлежащие налогообложению с начала года - Сумма ОПВ с начала года - Налоговые вычеты с начала года;

21) коэффициент перерасчета

Коэффициент перерасчета = количество месяцев до конца налогового периода / месяц по порядку его деятельности на данном предприятии;

22) расчетная сумма облагаемого ИПН дохода

Расчетная сумма облагаемого ИПН дохода = Сумма облагаемого ИПН дохода с начала года \* Коэффициент перерасчета;

23) сумма ИПН с расчетной суммы облагаемого ИПН доходов

Сумма ИПН с расчетной суммы облагаемого ИПН доходов= =ЕСЛИ(Расчетная сумма облагаемого ИПН дохода >=6991200, ТО 973525+(Расчетная сумма облагаемого ИПН дохода -6991200)\*20%; ИНАЧЕ ЕСЛИ ( Расчетная сумма облагаемого ИПН дохода >=2330400, ТО 274405+(Расчетная сумма облагаемого ИПН дохода -2330400)\*15%; ИНАЧЕ ЕСЛИ(Расчетная сумма облагаемого ИПН дохода >=466080, ТО 32043+(Расчетная сумма облагаемого ИПН дохода -466080)\*13%; ИНАЧЕ ЕСЛИ (Расчетная сумма облагаемого ИПН дохода >=174780, ТО 8739+(Расчетная сумма облагаемого ИПН дохода -174780)\*8%; ИНАЧЕ Расчетная сумма облагаемого ИПН дохода \*5%))));

24) сумма ИПН налога за отчетный месяц

Сумма ИПН налога за отчетный месяц = Сумма ИПН с расчетной суммы облагаемого ИПН доходов / Коэффициент перерасчета;

25) сумма ИПН, удержанные в предыдущие месяцы

Сумма ИПН, удержанные в предыдущие месяцы = Сумма ИПН налога за предыдущий отчетный месяц;

26) сумма ИПН, подлежащая удержанию

Сумма ИПН, подлежащая удержанию = Сумма ИПН налога за отчетный месяц - Сумма ИПН, удержанные в предыдущие месяцы;

27) доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ за отчетный месяц

Доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ за отчетный месяц = доходы, не подлежащие налогообложению в текущем месяце;

28) доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ с начала года

Доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ с начала года = Доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ за отчетный месяц + Доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ в предыдущем месяце с начала года;

29) доходы, облагаемые социальным налогом за отчетный период

Доходы, облагаемые социальным налогом за отчетный период = доходы, начисленные в текущем месяце - Обязательные пенсионные взносы за текущий месяц - Доходы, не являющиеся объектом обложения социальным налогом, за исключение ОПВ за отчетный месяц;

30) доходы, облагаемые социальным налогом с начала года

Доходы, облагаемые социальным налогом с начала года = Доходы, облагаемые социальным налогом за отчетный месяц + Доходы, облагаемые социальным налогом в предыдущем месяце с начала года;

31) расчетная сумма дохода, облагаемого социальным налогом

Расчетная сумма дохода, облагаемого социальным налогом = Доходы, облагаемые социальным налогом с начала года \* Коэффициент перерасчета;

32) сумма социального налога с расчетной суммы дохода, облагаемого социальным налогом

Сумма социального налога с расчетной суммы дохода, облагаемого социальным налогом = ЕСЛИ(Расчетная сумма дохода, облагаемого социальным налогом >=6991200, ТО 973525+( Расчетная сумма дохода, облагаемого социальным налогом -6991200)\*20%; ИНАЧЕ ЕСЛИ(Расчетная сумма дохода, облагаемого социальным налогом >=2330400, ТО 274405+(Расчетная сумма дохода, облагаемого социальным налогом -2330400)\*15%; ИНАЧЕ ЕСЛИ (Расчетная сумма дохода, облагаемого социальным налогом >=466080, ТО 32043+( Расчетная сумма дохода, облагаемого социальным налогом -466080)\*13%; ИНАЧЕ ЕСЛИ (Расчетная сумма дохода, облагаемого социальным налогом >=174780, ТО 8739+(Расчетная сумма дохода, облагаемого социальным налогом -174780)\*8%; ИНАЧЕ Расчетная сумма дохода, облагаемого социальным налогом \*5%))));

33) сумма социального налога за отчетный месяц

Сумма социального налога за отчетный месяц = Сумма социального налога с расчетной суммы дохода, облагаемого социальным налогом / /Коэффициент перерасчета;

34) сумма социального налога за предыдущие месяцы года

Сумма социального налога за предыдущие месяцы года = Сумма социального налога за предыдущий отчетный месяц;

35) сумма социального налога подлежащего уплате

Сумма социального налога подлежащего уплате = Сумма социального налога за отчетный месяц - Сумма социального налога за предыдущие месяцы года;

36) доходы физических лиц, с которых исчисляются социальные отчисления

Доходы физических лиц, с которых исчисляются социальные отчисления = доходы, начисленные в текущем месяце - доходы, не подлежащие налогообложению в текущем месяце - Обязательные пенсионные взносы за текущий месяц;

37) сумма социальных отчислений

Сумма социальных отчислений = Доходы физических лиц, с которых исчисляются социальные отчисления\*1,5% / 100%;

38) сумма социального налога за минусом социальных отчислений

Сумма социального налога за минусом социальных отчислений = Сумма социального налога подлежащего уплате - Сумма CО;

39) доходы подлежащие к выплате

Доходы подлежащие к выплате = доходы, начисленные в текущем месяце – ОПВ за текущий месяц – Сумма ИПН, подлежащая удержанию.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе, при проектировании информационной системы «Начисление заработной платы сотрудникам школы» были рассмотрены принципы проектирования концептуальной модели, логической модели и были рассмотрены основные причины, по которым данный выбор программного обеспечения Delphi был целесообразным.

Во второй главе дипломной работы было представлено непосредственно само подключение базы данных Access к среде Delphi. В данной главе был рассмотрен перенос логической модели данных в Access, принципы учета и контроля рабочего времени для всех категорий работающих, принципы удержания из заработной платы налоговых отчислений, этапы и формулы начисления заработной платы. Так же были представлены рисунки пользовательского интерфейса.

При разработке пользовательского интерфейса для информационной системы «Начисления заработной платы сотрудникам средне-специальных учебных заведений» расчет заработной платы производился для администрации и преподавательского состава. Также производились налоговые вычеты, по каждому месяцу, были созданы отчеты за текущий месяц по всему персоналу и квитки по каждому работнику.

Данная программа была создана по просьбе бухгалтеров школы №31, так как программа «1С – бухгалтерия» предназначена для различных сфер применения, и для того чтобы ее освоить необходимо очень много времени, сил и средств. Очень не многие бухгалтера могут освоить программу «1С - бухгалтерия», и поэтому они прибегают к другим способам расчета заработной платы к таким как бумажный способ или расчет заработной платы с помощью электронных таблиц Excel. В данном случае расчеты производятся в ручную по каждому сотруднику индивидуально. Это очень трудоемкий процесс и занимает очень много времени и ресурсов компьютера.

Созданная программа очень проста в применении и дополнительных знаний она не потребует. Данная программа удовлетворяет всем требованиям пожеланиям и замечаниям бухгалтеров школы №39 г. Усть-Каменогорска.

СПИСОК ЛИТЕРАТУРЫ

1. Симонович С. Специальная информатика /С. Симонович, Г. Евсеев, А. Алексеев. – М.: АСТпресс, 2001. – 478с.
2. Бакаревич Ю. Самоучитель Microsoft Access 2002 /Ю. Бакаревич, Н. Пушкина. - СПб.: БХВ – Петербург, 2004. – 720 с.
3. Бобровский С. Delphi7 учебный курс /С. Бобровский. – СПб.: Питер, 2004. – 834с.
4. Фаронов В. Система программирования Delphi /В. Фаронов. – СПб.: БХВ-Петербург, 2004. – 888с.
5. Родостовец В.П. Бухгалтерский учет для предприятий /В.П. Родостовец. – Алматы: Каржы – Каражат, 2002. – 978с.
6. Смурыгина И. Бюллетень бухгалтера: В помощь бухгалтеру /И. Смурыгина, К. Жанбыршиева // БИКО – 2005. - №6. – с.6-9
7. Гончаров А.Ю. Access 2003 /А.Ю. Гончаров. – М.: Кудиц – Образ, 2004. – 964с.
8. Михеева В. Microsoft Access 2002 / В. Михеева, И. Харитонова. - СПб.: БХВ – Петербург, 2003. – 1040с.
9. Фаронов В. Программирование баз данных в Delphi /В. Фаронов. – СПб.: Питер, 2004. – 759с.
10. Гофман В. Работа с базами данных в Delphi /В. Гофман. – СПб.: БХВ – Петербург, 2003. – 958с.
11. Роб П. Системы баз данных: проектирование, реализация и управление /П. Роб. - СПб.: БХВ – Петербург, 2004. – 1018с.
12. Гофман В. Работа с базами данных в Delphi /В. Гофман, А. Хомоненко. - СПб.: БХВ – Петербург, 2003. – 624с.
13. Драхвелидзе П. Программирование в Delphi 7 /П. Драхвелидзе, Е. Марков. - СПб.: БХВ – Петербург, 2004. – 784 с.
14. Хомоненко А. Самоучитель Delphi /А. Хомоненко, В. Гофман. - СПб.: БХВ – Петербург, 2003. – 576с.
15. Михеев С.Л. Access 2000 /С.Л. Михеев.– М.: Кудиц – Образ, 2000. – 964с.
16. Роббинс Д. Отладка приложений /Д. Робинс. – СПб.: БХВ – Петербург, 2003. – 512с.
17. Бекаревич Ю. Самоучитель Microsoft Access 2000 /Ю. Бекаревич, Н. Пушкина.- СПб.: БХВ – Петербург, 2000. – 480с.
18. Бекаревич Ю. Microsoft Access 2000 за 30 занятий / Ю. Бекаревич, Н. Пушкина.- СПб.: БХВ – Петербург, 2000. – 480с.
19. Мамаев Е. MS SQL Server: проектирование и реализация баз данных /Е. Мамаев.- СПб.: БХВ – Петербург, 2003. – 496с.
20. Понамарев В. COM и ActiveX в Delphi /В. Понамарев.- СПб.: БХВ – Петербург, 2000. – 320с.
21. Тихомиров Ю. Microsoft SQL Server /Ю. Тихомиров.- СПб.: БХВ – Петербург, 2000. – 320с.
22. Долженков В. Microsoft Excel 2002 /В. Долженков, Ю. Колесников.- СПб.: БХВ – Петербург, 2000. – 1072с.
23. Андерсен В. Разработка приложений в Access 97 /В. Андерсен.- СПб.: БХВ – Петербург, 2000. – 320с.
24. Гарнаев А. Microsoft Excel: разработка приложений /А. Гарнаев.- СПб.: БХВ – Петербург, 2000. – 320 с.
25. Родостовец В.П. Бухгалтерский учет: учебный курс /В.П. Родостовец. – Алматы: Каржы – Каражат, 2002. – 978с.
26. Фаронов В. Система программирования Delphi /В. Фаронов. – СПб.: БХВ-Петербург, 2004. – 888с.
27. Гофман В. Работа с базами данных в Delphi /В. Гофман, А. Хомоненко. - СПб.: БХВ – Петербург, 2003. – 624с.
28. Понамарев В. Механизм доступа ActiveX в Delphi /В. Понамарев.- СПб.: БХВ – Петербург, 2000. – 320с.
29. Бекаревич Ю. Microsoft Access 2000: Экспресс-курс /Ю. Бекаревич, Н. Пушкина.- СПб.: БХВ – Петербург, 2000. – 480с.
30. Погорелов В. Delphi: Быстрый старт /В. Погорелов.- СПб.: БХВ – Петербург, 2000. – 320с.

ПРИЛОЖЕНИЕ А

Листинг главной формы программы

unit UnMainForm;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, menu, jpeg, StdCtrls, Buttons;

var

MainForm: TMainForm;

implementation

uses menu\_ras, dmPayU, prjBase, grid\_Year;

{$R \*.dfm}

procedure TMainForm.FormCreate(Sender: TObject);

begin

SpeedButton1.Hint:='Выход';

SpeedButton2.Hint:='Справка';

SpeedButton3.Hint:='Расчет ЗП';

SpeedButton4.Hint:='Данные за прошедшие года';

SpeedButton5.Hint:='Ввод\редактирование '+chr(13)+ 'данных о работниках';

end;

procedure TMainForm.Label2Click(Sender: TObject);

begin

MainForm.Visible:=false;

frm\_menu.ShowModal;

end;

procedure TMainForm.Label5Click(Sender: TObject);

begin

winhelp(MainForm.Handle,'HLP1.HLP',HELP\_CONTEXT, 1);

end;

procedure TMainForm.Label3Click(Sender: TObject);

begin

MainForm.Visible:=false;

menu2.ShowModal;

end;

procedure TMainForm.Label7Click(Sender: TObject);

begin

TabYear.ShowModal;

end;

procedure TMainForm.SpeedButton1Click(Sender: TObject);

begin

mainform.Close;

end;

procedure TMainForm.SpeedButton5Click(Sender: TObject);

begin

MainForm.Visible:=false;

frm\_menu.ShowModal;

end;

procedure TMainForm.SpeedButton3Click(Sender: TObject);

begin

MainForm.Visible:=false;

menu2.ShowModal;

end;

procedure TMainForm.SpeedButton4Click(Sender: TObject);

begin

TabYear.ShowModal;

end;

procedure TMainForm.SpeedButton2Click(Sender: TObject);

begin

winhelp(MainForm.Handle,'HLP1.HLP',HELP\_CONTEXT, 1); end;

ПРИЛОЖЕНИЕ Б

Листинг модуля данных

unit dmPayU;

interface

uses

SysUtils, Classes, DB, ADODB, DdeMan, OleServer, ExcelXP;

procedure moveCalcFields(DataSet: TDataSet);

procedure personCalcFields(DataSet: TDataSet);

procedure yearCalcFields(DataSet: TDataSet);

var

dmPay: TdmPay;

implementation

{$R \*.dfm}

procedure TdmPay.moveCalcFields(DataSet: TDataSet);

begin

//ставка для расчета

with dmPay.move do

fields[33].AsCurrency:=fields[3].AsFloat\*person.Fields[4].AsCurrency;

//недельный час (1-4)классах

if dmpay.move.Fields[28].AsBoolean=true then

with dmpay.move do

fields[34].AsCurrency:=fields[33].AsCurrency/coff.Fields[1].AsFloat

else

with dmpay.move do

fields[34].AsCurrency:=0;

//недельный час (5-11)классах

if dmpay.move.Fields[28].AsBoolean=true then

with dmpay.move do

fields[35].AsCurrency:=fields[33].AsCurrency/coff.Fields[2].AsFloat

else

with dmpay.move do

fields[35].AsCurrency:=0;

//разовый час (1-4)

if dmpay.move.Fields[28].AsBoolean=true then

with dmpay.move do

fields[36].AsCurrency:=fields[33].AsCurrency/coff.Fields[3].AsFloat

else

with dmpay.move do

fields[36].AsCurrency:=0;

//разовый час (5-11)

if dmpay.move.Fields[28].AsBoolean=true then

with dmpay.move do

fields[37].AsCurrency:=fields[33].AsCurrency/coff.Fields[4].AsFloat

else

with dmpay.move do

fields[37].AsCurrency:=0;

//Экология

if dmpay.move.Fields[28].AsBoolean=true then

with dmPay.move do

fields[39].AsFloat:=coff.Fields[5].AsFloat\*person.fields[9].AsFloat/person.fields[7].AsFloat

else dmpay.move.fields[39].AsCurrency:=0;

//ЗП

with dmpay.move do

fields[40].AsCurrency:=((fields[5].AsFloat+fields[6].AsFloat+fields[9].AsFloat+fields[10].AsFloat+fields[13].AsFloat+fields[14].AsFloat)\*fields[34].AsFloat+(fields[7].AsFloat+fields[8].AsFloat+fields[11].AsFloat+fields[12].AsFloat+fields[15].AsFloat+fields[16].AsFloat)\*fields[35].AsFloat)\*person.Fields[9].AsInteger/person.Fields[7].AsInteger;

// Проверка тетрадей

if (dmPay.move.FieldByName('Предмет').AsString='начальные классы') then

begin

with dmpay.move do

fields[41].AsCurrency:=dmpay.person.Fields[4].AsCurrency\*fields[25].AsFloat\* person.Fields[9].AsInteger/person.Fields[7].AsInteger

end else

with dmpay.move do

if fields[29].AsBoolean=true then

begin

fields[41].AsCurrency:=((fields[5].AsFloat+fields[6].AsFloat+(fields[9].AsFloat+ fields[10].AsFloat+fields[13].AsFloat+fields[14].AsFloat)/2)\*fields[25].AsFloat/ coff.Fields[1].AsFloat+((fields[7].AsFloat+(fields[11].AsFloat+fields[15].AsFloat) /2)\*fields[26].AsFloat+(fields[8].AsFloat+(fields[12].AsFloat+fields[16].AsFloat)/2)\*fields[27].AsFloat/coff.Fields[2].AsFloat)\*person.Fields[4].AsCurrency\* person.Fields[9].AsInteger/person.Fields[7].AsInteger

end else

with dmpay.move do

fields[41].AsCurrency:=((fields[5].AsFloat+fields[6].AsFloat)\*fields[25].AsFloat/ coff.Fields[1].AsFloat+(fields[7].AsFloat\*fields[26].AsFloat+fields[8].AsFloat\* fields[27].AsFloat/coff.Fields[2].AsFloat)\*person.Fields[4].AsCurrency\* person.Fields[9].AsInteger/person.Fields[7].AsInteger;

// Замена

with dmpay.move do

fields[42].AsCurrency:= (fields[17].AsFloat+ fields[18].AsFloat) \* fields[36].AsFloat + ( fields[19].AsFloat+ fields[20].AsFloat)\* fields[37].AsFloat;

// Углубленное изучение

with dmpay.move do

fields[43].AsCurrency:=((fields[5].AsFloat+fields[6].AsFloat+fields[9].AsFloat+ fields[10].AsFloat+fields[13].AsFloat+fields[14].AsFloat)/coff.Fields[1].AsFloat+ (fields[7].AsFloat+fields[11].AsFloat+fields[15].AsFloat+fields[8].AsFloat+ fields[12].AsFloat+fields[16].AsFloat)/coff.Fields[2].AsFloat)\* person.Fields[4].AsCurrency\*fields[22].AsFloat\*person.Fields[9].AsInteger/ person.Fields[7].AsInteger;

// Замена углубленного изучения

with dmpay.move do

fields[44].AsCurrency:=((fields[17].AsFloat+fields[18].AsFloat)/coff.Fields[3].AsFloat + (fields[19].AsFloat+fields[20].AsFloat) / coff.Fields[4].AsFloat) \* fields[22].AsFloat\*person.Fields[4].AsFloat;

//Оплата ставки

with dmpay.move do

fields[45].AsCurrency:=fields[30].AsFloat\*fields[33].AsFloat\*person.Fields[9].AsInteger / person.Fields[7].AsInteger;

//Классное руководство

with dmpay.move do

fields[46].AsCurrency:=person.Fields[4].AsFloat\*fields[21].AsFloat\* person.Fields[9].AsInteger / person.Fields[7].AsInteger;

//Вредность

with dmpay.move do

fields[47].AsCurrency:=person.Fields[4].AsFloat\*fields[23].AsFloat\* person.Fields[9].AsInteger/person.Fields[7].AsInteger;

// Внеклассная работа

with dmpay.move do

fields[48].AsCurrency:=person.Fields[4].AsFloat\*fields[24].AsFloat\* person.Fields[9].AsInteger/person.Fields[7].AsInteger;

//Итого

with dmpay.move do

fields[49].AsCurrency:=fields[47].AsCurrency+fields[46].AsCurrency+ fields[45].AsCurrency+fields[44].AsCurrency+fields[42].AsCurrency+ fields[41].AsCurrency+fields[40].AsCurrency+fields[39].AsCurrency+ fields[48].AsCurrency+fields[43].AsCurrency;

//Обязательные пенсионные взносы

with dmpay.move do

fields[53].AsCurrency:=fields[54].AsFloat\*fields[49].AsFloat;

end;

procedure TdmPay.personCalcFields(DataSet: TDataSet);

begin

with dmPay.person do

fields[9].AsFloat:=fields[7].AsFloat-fields[8].AsFloat;

end;

Листинг окна поиска:

procedure Tfff.SpeedButton2Click(Sender: TObject);

var LO:TLocateOptions;

begin

LO:=[];

if CheckBox1.Checked then LO:=LO+[loCaseInsensitive];

if Checkbox2.Checked then LO:=[loPartialKey];

dmPay.person.Locate('P\_Name',edit1.Text,LO);

fff.Close;

end;

end.

ПРИЛОЖЕНИЕ В

Листинг окна программы «Сумма ИПН»

unit ipn;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, DBCtrls, Mask, ExtCtrls, ExcelXP, OleServer, Menus;

var

frm\_ipn: Tfrm\_ipn;

implementation

uses dmPayU, sum\_soc, soc, prjBase, tabTime, frm\_tim, frm\_kl\_ruk, frm\_propusk, PropD, frm\_vnekl, frm\_vred, frm\_ekolg, Zam, frm\_kn;

{$R \*.dfm}

procedure Tfrm\_ipn.Button1Click(Sender: TObject);

var n:OleVariant;

begin

n:='C:\ЗарПлата\nal.xls';

excelapplication1.Workbooks.Open(n,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,0);

ExcelWorkbook1.ConnectTo(ExcelApplication1.ActiveWorkbook);

ExcelApplication1.Cells.Item[13,11].value:=DBedit2.Text;

ExcelWorkbook1.Save;

// обмен данными

edit39.Text:=excelapplication1.Cells.Item[13,10];

edit40.Text:=excelapplication1.Cells.Item[14,10];

edit41.Text:=excelapplication1.Cells.Item[15,10];

edit42.Text:=excelapplication1.Cells.Item[16,10];

edit43.Text:=excelapplication1.Cells.Item[17,10];

edit44.Text:=excelapplication1.Cells.Item[18,10];

edit45.Text:=excelapplication1.Cells.Item[19,10];

edit46.Text:=excelapplication1.Cells.Item[20,10];

edit47.Text:=excelapplication1.Cells.Item[21,10];

edit48.Text:=excelapplication1.Cells.Item[22,10];

edit49.Text:=excelapplication1.Cells.Item[23,10];

edit50.Text:=excelapplication1.Cells.Item[24,10];

edit52.Text:=excelapplication1.Cells.Item[25,10];

edit27.Text:=excelapplication1.Cells.Item[13,12];

edit28.Text:=excelapplication1.Cells.Item[14,12];

edit29.Text:=excelapplication1.Cells.Item[15,12];

edit30.Text:=excelapplication1.Cells.Item[16,12];

edit31.Text:=excelapplication1.Cells.Item[17,12];

edit32.Text:=excelapplication1.Cells.Item[18,12];

edit33.Text:=excelapplication1.Cells.Item[19,12];

edit34.Text:=excelapplication1.Cells.Item[20,12];

edit35.Text:=excelapplication1.Cells.Item[21,12];

edit36.Text:=excelapplication1.Cells.Item[22,12];

edit37.Text:=excelapplication1.Cells.Item[23,12];

edit38.Text:=excelapplication1.Cells.Item[24,12];

Edit1.Text:=excelapplication1.Cells.Item[13,13];

edit2.Text:=excelapplication1.Cells.Item[14,13];

edit3.Text:=excelapplication1.Cells.Item[15,13];

edit4.Text:=excelapplication1.Cells.Item[16,13];

edit5.Text:=excelapplication1.Cells.Item[17,13];

edit6.Text:=excelapplication1.Cells.Item[18,13];

edit7.Text:=excelapplication1.Cells.Item[19,13];

edit8.Text:=excelapplication1.Cells.Item[20,13];

edit9.Text:=excelapplication1.Cells.Item[21,13];

edit10.Text:=excelapplication1.Cells.Item[22,13];

edit11.Text:=excelapplication1.Cells.Item[23,13];

edit12.Text:=excelapplication1.Cells.Item[24,13];

Edit13.Text:=excelapplication1.Cells.Item[13,11];

edit14.Text:=excelapplication1.Cells.Item[14,11];

edit15.Text:=excelapplication1.Cells.Item[15,11];

edit16.Text:=excelapplication1.Cells.Item[16,11];

edit17.Text:=excelapplication1.Cells.Item[17,11];

edit18.Text:=excelapplication1.Cells.Item[18,11];

edit19.Text:=excelapplication1.Cells.Item[19,11];

edit20.Text:=excelapplication1.Cells.Item[20,11];

edit21.Text:=excelapplication1.Cells.Item[21,11];

edit22.Text:=excelapplication1.Cells.Item[22,11];

edit23.Text:=excelapplication1.Cells.Item[23,11];

edit24.Text:=excelapplication1.Cells.Item[24,11];

end;

procedure Tfrm\_ipn.N3Click(Sender: TObject);

var n:OleVariant;

begin

n:='C:\ЗарПлата\nal.xls';

excelapplication1.Workbooks.Open(n,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,emptyParam,0);

ExcelWorkbook1.ConnectTo(ExcelApplication1.ActiveWorkbook);

ExcelApplication1.Cells.Item[13,11].value:=DBedit2.Text;

ExcelWorkbook1.Save;

edit39.Text:=excelapplication1.Cells.Item[13,10];

edit40.Text:=excelapplication1.Cells.Item[14,10];

edit41.Text:=excelapplication1.Cells.Item[15,10];

edit42.Text:=excelapplication1.Cells.Item[16,10];

edit43.Text:=excelapplication1.Cells.Item[17,10];

edit44.Text:=excelapplication1.Cells.Item[18,10];

edit45.Text:=excelapplication1.Cells.Item[19,10];

edit46.Text:=excelapplication1.Cells.Item[20,10];

edit47.Text:=excelapplication1.Cells.Item[21,10];

edit48.Text:=excelapplication1.Cells.Item[22,10];

edit49.Text:=excelapplication1.Cells.Item[23,10];

edit50.Text:=excelapplication1.Cells.Item[24,10];

edit52.Text:=excelapplication1.Cells.Item[25,10];

edit27.Text:=excelapplication1.Cells.Item[13,12];

edit28.Text:=excelapplication1.Cells.Item[14,12];

edit29.Text:=excelapplication1.Cells.Item[15,12];

edit30.Text:=excelapplication1.Cells.Item[16,12];

edit31.Text:=excelapplication1.Cells.Item[17,12];

edit32.Text:=excelapplication1.Cells.Item[18,12];

edit33.Text:=excelapplication1.Cells.Item[19,12];

edit34.Text:=excelapplication1.Cells.Item[20,12];

edit35.Text:=excelapplication1.Cells.Item[21,12];

edit36.Text:=excelapplication1.Cells.Item[22,12];

edit37.Text:=excelapplication1.Cells.Item[23,12];

edit38.Text:=excelapplication1.Cells.Item[24,12];

Edit1.Text:=excelapplication1.Cells.Item[13,13];

edit2.Text:=excelapplication1.Cells.Item[14,13];

edit3.Text:=excelapplication1.Cells.Item[15,13];

edit4.Text:=excelapplication1.Cells.Item[16,13];

edit5.Text:=excelapplication1.Cells.Item[17,13];

edit6.Text:=excelapplication1.Cells.Item[18,13];

edit7.Text:=excelapplication1.Cells.Item[19,13];

edit8.Text:=excelapplication1.Cells.Item[20,13];

edit9.Text:=excelapplication1.Cells.Item[21,13];

edit10.Text:=excelapplication1.Cells.Item[22,13];

edit11.Text:=excelapplication1.Cells.Item[23,13];

edit12.Text:=excelapplication1.Cells.Item[24,13];

Edit13.Text:=excelapplication1.Cells.Item[13,11];

edit14.Text:=excelapplication1.Cells.Item[14,11];

edit15.Text:=excelapplication1.Cells.Item[15,11];

edit16.Text:=excelapplication1.Cells.Item[16,11];

edit17.Text:=excelapplication1.Cells.Item[17,11];

edit18.Text:=excelapplication1.Cells.Item[18,11];

edit19.Text:=excelapplication1.Cells.Item[19,11];

edit20.Text:=excelapplication1.Cells.Item[20,11];

edit21.Text:=excelapplication1.Cells.Item[21,11];

edit22.Text:=excelapplication1.Cells.Item[22,11];

edit23.Text:=excelapplication1.Cells.Item[23,11];

edit24.Text:=excelapplication1.Cells.Item[24,11];

end;

end.