МОСКОВСКИЙ ФИНАНСОВО-ЮРИДИЧЕСКИЙ УНИВЕРСИТЕТ МФЮА

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Студента**

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

(фамилия, имя, отчество)

**На тему:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **Автор работы:** |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О.) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| **Научный руководитель:** |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (ученая степень, звание, Ф.И.О.) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

|  |  |
| --- | --- |
| **Консультант:** |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (ученая степень, звание, Ф.И.О.) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

«Допустить к защите»

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_» \_\_\_\_\_\_\_\_20\_\_ г.

Москва 20\_\_

МОСКОВСКИЙ ФИНАНСОВО-ЮРИДИЧЕСКИЙ УНИВЕРСИТЕТ МФЮА

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«УТВЕРЖДАЮ»

Зав. кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_» \_\_\_\_\_\_\_\_20\_\_ г.

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ**

**ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**Студенту \_\_**

(фамилия, имя, отчество)

**Тема:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Исходные данные к работе: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Содержание пояснительной записки (перечень подлежащих  
разработке вопросов): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Срок предоставления студентом законченной работы** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ученая степень, звание, ФИО)

Задание принял к исполнению\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ученая степень, звание, ФИО)

МОСКОВСКИЙ ФИНАНСОВО-ЮРИДИЧЕСКИЙ УНИВЕРСИТЕТ МФЮА

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| «СОГЛАСОВАНО» | «УТВЕРЖДАЮ» |
| Научный руководитель | Зав. кафедрой |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| «\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. | «\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_г. |

**ГРАФИК**

**написания и оформления выпускной квалификационной работы**

**Ф.И.О. студента(ки) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Тема выпускной квалификационной работы:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Мероприятия | Сроки выполнения | Отметка  руководи­теля  о вы­полнении |
| 1. | Подбор литературы, ее изучение и обработка. Составление библио­графии по основным источникам | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 2. | Составление плана ВКР и согласование его с руководителем | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 3. | Разработка и представление на проверку первой главы | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 4. | Накопление, систематизация анализ практических материалов | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 5. | Разработка и представление на про­верку второй главы | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 6. | Согласование с руководителем вы­водов и предложений | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 7. | Переработка (доработка) ВКР в соот­ветствии с замечаниями и представ­ление ее на кафедру | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
|  |
| 8. | Разработка тезисов доклада для защиты | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 9. | Ознакомление с отзывом и рецензией | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |
| 10. | Завершение подготовки к защите с учетом отзыва и рецензии | до «\_\_» \_\_\_\_\_\_\_\_ 20\_\_г. |  |

График составил «\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

[Введение 6](#_Toc451815622)

[1. Аналитическая часть 9](#_Toc451815623)

[1.1. Характеристика предприятия 9](#_Toc451815624)

[1.2. Анализ производственных процессов требующих автоматизации 11](#_Toc451815625)

[1.3. Обзор существующих разработок в сфере автоматизации документооборота полиграфического предприятия 18](#_Toc451815626)

[1.4. Постановка задачи проектирования 21](#_Toc451815627)

[1.5. Обоснование необходимости использования информационных технологий в полиграфической деятельности 23](#_Toc451815628)

[1.6. Формальная модель системы автоматизации работы с клиентами 26](#_Toc451815629)

[1.7. Обоснование выбора СУБД 28](#_Toc451815630)

[1.8. Обоснование выбора среды разработки 29](#_Toc451815631)

[2. Проектная часть 30](#_Toc451815632)

[2.1. Информационное обеспечение деятельности полиграфического предприятия 30](#_Toc451815633)

[2.1.1. Проектирование концептуальной модели базы данных 30](#_Toc451815634)

[2.1.2. Проектирование физической модели базы данных 32](#_Toc451815635)

[2.1.3. Описание связей и полей 33](#_Toc451815636)

[2.2. Программная реализация приложения 36](#_Toc451815637)

[2.2.1 Проектирование интерфейса 36](#_Toc451815638)

[2.2.2. Взаимодействие с базой данных 44](#_Toc451815639)

[2.2.3. Особенности программной реализации функционала приложения 47](#_Toc451815640)

[2.3. Структурная схема приложения 47](#_Toc451815641)

[2.4. Схемы основных алгоритмов 48](#_Toc451815642)

[2.5. Организация процессов обработки информации с использованием приложения 49](#_Toc451815643)

[3. Расчет экономической эффективности проекта 54](#_Toc451815644)

[3.1. Выбор методики расчета экономической эффективности 54](#_Toc451815645)

[3.2. Расчет затрат на разработку программного продукта 57](#_Toc451815646)

[3.3. Оценка экономической эффективности внедрения автоматизации дипломной документации 63](#_Toc451815647)

[3.4. Оценка периода окупаемости проекта 65](#_Toc451815648)

[Заключение 67](#_Toc451815649)

[Список использованной литературы 69](#_Toc451815650)

[Приложение. Код программы 72](#_Toc451815651)

РЕФЕРАТ

Пояснительная записка 74 страниц, 35 рисунков, 8 таблиц, 35 источников, 1 приложения.

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, БАЗА ДАННЫХ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ.

Объект исследования –

Цель работы –

Метод исследования –

Полученные результаты и их новизна –

Основные конструктивные, технологические и технологические технико-эксплуатационные характеристики –

Степень внедрения –

Рекомендации по внедрению –

Область применения –

Экономическая эффективность или значимость работы –

Прогнозные предположения о развитии объекта исследования –

Введение

**Актуальность темы.** С каждым днем во всех сферах жизнедеятельности человека возрастает объём и оборот информации, что влечет за собой ускорение процессов накопления, обработки и использования знаний. В связи с этим возникает необходимость использования автоматических средств, позволяющих эффективно хранить, обрабатывать и распределять накопленные данные. Эффективная деятельность любого современного предприятия в соответствии с все большими требованиями, предъявляемыми к качеству его работы, полностью зависит от уровня оснащения компании информационными средствами на базе компьютерных систем автоматизированного учета.

Полиграфическая промышленность – одна из немногих, стабильно развивающихся сегодня в нашей стране отраслей, которая вошла во все отрасли и направления экономической деятельности. К полиграфии обращается все больше людей и организаций, которые ищут наиболее рациональные современные методы размножения и распространения информации – от однокрасочных бланков и листовок до многокрасочных журналов, буклетов, проспектов, открыток и другой высококачественной полиграфической продукции. Качественные полиграфические услуги играют важную роль в продвижении товаров или услуг на рынке. Современная типография – это полиграфическое предприятие, которое осуществляет полный процесс создания печатной продукции. Оборудованные высокоэффективным и современным оборудованием полиграфические компании, фирмы и типографии могут издавать самые разные виды полиграфической продукции. Сегодня большинство полиграфических предприятий выполняют полный комплекс технологических операций, необходимых для создания высококачественного печатного продукта и при необходимости предоставляют заказчикам услуги дизайнера, что значительно облегчает процесс создания качественной полиграфической продукции. Современную типографию характеризуют быстрые сроки печати и высокая степень гибкости, ее продукция является более индивидуальной, привлекательной и доступной для потребителя.

В условиях современной жизни требуется ускорение процессов обработки информационных потоков в полиграфической промышленности, поэтому особую актуальность приобретают вопросы повышения эффективности работы менеджеров по работе с клиентами, в частности за счет программного обеспечения и автоматизации рабочего места. Используя преимущества компьютерных технологий, современная полиграфическая компания уверенно и успешно работает сегодня, что закладывает фундамент для будущего процветания.

**Цель дипломного проекта** - разработка автоматизированного рабочего места менеджера по работе с клиентами в компании ООО «Неопринт», занимающейся предоставлением полиграфических услуг.

**Объект исследования** - процесс функционирования ООО «Неопринт», который после исследования формализуется и реализуется в виде автоматизированной информационной системы.

**Предмет исследования** - предметная область деятельности ООО «Неопринт», принятый на предприятии стандарт и механизм документооборота, установившаяся система отчетности. В процессе исследования предметной области предполагается выявить недостатки существующей организации процесса функционирования предприятия и устранить их в реализуемом рабочем месте.

**Задачи исследования**. Для достижения указанной цели в рамках дипломной работы необходимо решить ряд следующих задач:

- рассмотреть и проанализировать основные технические аспекты создания автоматизированного рабочего места менеджера по работе с клиентами в сфере полиграфии, и в частности в компании ООО «Неопринт»;

- определить основные цели и задачи автоматизации рабочего места менеджера по работе с клиентами компании ООО «Неопринт»;

- проанализировать существующие разработки для решения поставленной задачи автоматизации;

- формализовать задачу, выбрать стратегию автоматизации;

- разработать информационное и программное обеспечение автоматизации рабочего места менеджера

- обосновать необходимость и экономическую эффективность проекта.

Теоретико-методической основной исследования являются современные подходы и разработки в области управления предприятием. Эти подходы были взяты за основу последующей разработки дипломного проекта.

**Новизна работы** - разработанная система автоматизации рабочего места менеджера по работе с клиентами позволяет производить быстрый расчет стоимости полиграфической продукции, что способствует более качественному обслуживанию клиентов, быстрому предоставлению информации о стоимости того или иного варианта заказа, своевременному выполнению заказа в найкратчайшие сроки.

**Практическое значение** – разработанная система автоматизации для расчета стоимости полиграфической продукции может быть использована менеджерами по работе с клиентами в компании ООО «Неопринт».

**Структура работы.** Дипломная работа состоит из введения, четырех разделов, которые делятся на параграфы, заключения, списка литературы и приложений. Работа изложена на 74 странице, включает 8 таблиц и 35 рисунков. Библиографический список состоит из 28 источников.

1. Аналитическая часть

1.1. Характеристика предприятия

Типография "Неопринт" работает на рынке самоклеящихся этикеток с 2005 г. и предлагает весь спектр работ по флексографической печати самоклеящихся этикеток и изготовлению термочеков. Типография делает ставку на высокое качество выпускаемой продукции, поэтому работает на самом современном и качественном оборудовании фирмы HEIDELBERG. В апреле 2006 года компания приобрела первый станок HEIDELBERG Gallus 280, который имел возможность печатать шестикрасочные заказы. Со временем, вместе с развитием фирмы, ростом количества клиентов, партнеров, их потребностей, в 2008 году компанией был приобретен второй станок HEIDELBERG Gallus 280.

Компания старается не останавливаться на достигнутом и двигаться вперед вместе с развитием современного рынка полиграфической продукции, идти в ногу с технологическими новшествами в области предоставления полиграфических услуг. Компания работает со всем ассортиментом самоклеящихся материалов и ведущими мировыми поставщиками бумаги, пленки, термобумаги, красок и всевозможных лаков. Новые станки оборудованы в соответствии с новыми технологическими разработками в области полиграфического производства и имеют возможность печати 9-ти красочных заказов с использованием тиснения (холодного и горячего), трафарета, конгрева, ламинации, переворота полотна, печати на клеевом слое и многих других функций. Основными преимуществами этих машин является широкий диапазон запечатываемых материалов; точное совмещение красок; качество печати до 175 Lpi; использование УФ-красок, обладающих хорошей насыщенностью и глянцем (УФ-краски являются стандартом качества во флексографии). Компания также использует расходные материалы и краски от Fasson, Raflatac, Sericol, Akzo Nobel.

Исходя из главной заботы компании о клиенте, можно определить ее миссию - стать надёжным долговременным партнёром, основывая отношения на тесном взаимовыгодном сотрудничестве, используя индивидуальный подход к каждому клиенту.

Основная цель ООО «Неопринт» - удовлетворение спроса на качественную рулонную этикетку в минимальный срок и по разумной цене. Специалисты компании постоянно изучают и анализируют смежные рынки, используют собственный опыт и стараются воплотить в жизнь идеи партнёров и клиентов. Организационная структура компании представлена на рис.1.1.

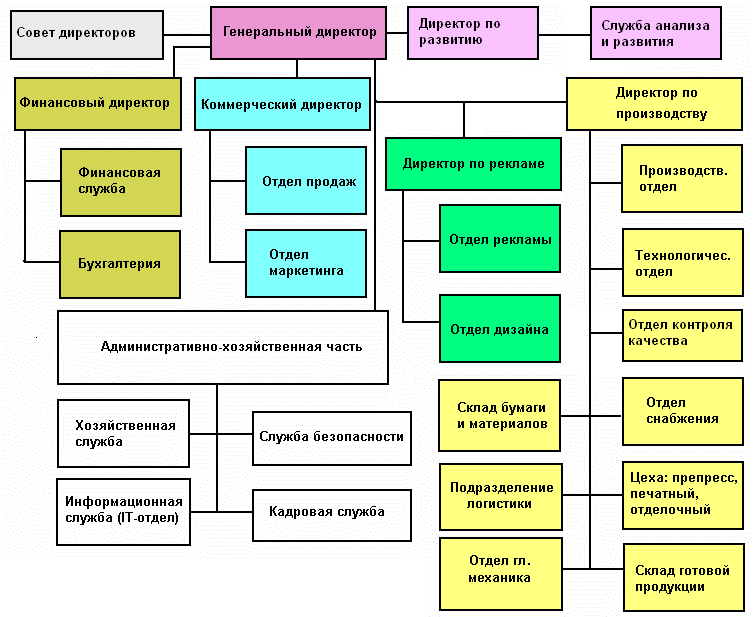


Рис1.1. Организационная структура компании ООО «Неопринт»

Благодаря широким техническим возможностям полиграфического производства, индивидуальному подходу к каждому клиенту и гибкой политике цен, компания сотрудничает с большим количеством партнеров в самых разнообразных отраслях промышленности и сферах деятельности. Четкая специализация позволяет менеджерам сосредоточиться на собственных и новейших технологиях в производстве этикеток своего направления для того, чтобы разрабатывать продукцию с учётом современных требований рынка. Направления курируют высокопрофессиональные менеджеры, которые могут предложить оригинальные решения исполнения и печати самоклеящихся этикеток для каждого определенного продукта.

За несколько лет успешной работы компания приобрела достаточный опыт во всем цикле работ по изготовлению печатной продукции: начиная от разработки дизайна самоклеящихся этикеток и заканчивая отгрузкой и доставкой. В компании работают высокопрофессиональные сотрудники, которые имеют за плечами многолетний опыт работы в крупных печатных компаниях. Профессионализм и ответственный подход к выполнению задач позволяет компании успешно развиваться и расширять техническую базу и производство. Типография выполняет заказы по печати самоклеящихся этикеток для многих известных предприятий, имеет собственный логистический отдел, который обеспечивает быструю доставку готовой продукции в любую точку страны. Для осуществления доставки в отдаленные регионы России (Сибирь, Дальний Восток, Сахалин) компания сотрудничает с лучшими транспортными логистическими компаниями.

1.2. Анализ производственных процессов требующих автоматизации

В ООО «Неопринт» полностью независимое производство, начиная от собственных зданий, электрической подстанции, флексографических печатных машин до собственного цеха для производства флексоформ, поэтому типография является технически независимой и осуществляет полный цикл производства самоклеящейся этикетки, флексографических полимерных форм, резки ролей и упаковки готовой продукции.

ООО «Неопринт» предоставляет следующие виды услуг на рынке полиграфической продукции:

1. *Изготовление самоклеящихся этикеток*. Это направление является одним из приоритетных в деятельности компании «Неопринт». Самоклеящиеся этикетки – незаменимый атрибут любого товара, представляющий компанию-производителя, ее логотип, поэтому дизайн и материал изготовления играют далеко не последнюю роль в создании впечатления о фирме.

2. *Конгревное тиснение (конгрев)*. С помощью конгревного тиснения получают многоуровневое рельефное (выпукло-вогнутое) изображение. Данный вид тиснения придает особую привлекательность и элегантный вид любому виду печатной продукции, будь то открытка, календарь, брошюра или этикетка. Конгревом можно достичь дополнительных задач в художественном и дизайнерском смысле. Методом декоративного оформления горячим тиснением достигается эффект трехмерности, позволяющей создать нечто отличное от плоского оттиска. Конгрев помогает выделить необходимые детали рисунка. Технологический прием конгревной обработки печатной продукции позволяет ощутимо отличить носитель информации от изделий с плоской поверхностью.

3. *Горячее тиснение фольгой*. С помощью структурного горячего тиснения возможно создание фактурных оттисков, имитирующих лен, пергамент, кожу и другие популярные природные материалы. Одной из разновидностей этой технологии является горячее тиснение фольгой, которое обычно не применяют на фактурных материалах, поскольку расплавленную с помощью высоких температур фольгу довольно сложно равномерно нанести на неровную поверхность.

3. *Холодное тиснение фольгой*. Дешевый процесс, который позволяет улучшить оформление печатной продукции без значительного увеличения ее стоимости, помогая полиграфистам, не предоставлявшим ранее услуги тиснения, выходить на новый рынок.

4. *Ламинирование*. это процесс покрытия полиграфической продукции полипропиленовой плёнкой. Покрытие может осуществляться, как с одной стороны листа, так и с двух. При ламинировании могут использоваться глянцевые и матовые пленки различной плотности.

5. *Печать УФ лаком*. Один из наиболее оригинальных методов производства. Покрытие УФ-лаком придает полиграфической продукции глянец, создает престижный и деловой образ, привлекающий внимание клиентов. Печать УФ лаком представляет собой покрытие изделия пленкой лака, который, подвергаясь обработке УФ-лучами, полимеризуется и затвердевает, что придает высокое качество отделки, сравнимое с ламинированием.

6. *Печать на оборотной стороне*. Компания «Неопринт» предлагает своим клиентам также печать на оборотной стороне самоклеющихся пленок. Следует отметить, что данная услуга не имеет большой популярности, но чаще всего печать на оборотной стороне используется для этикеток на прозрачных бутылках (шампунь, водка, минеральная вода или иная продукция в прозрачной упаковке).

7. *Печать термочеков*. Термочек, как правило, не содержит никакой информации, она наносится заказчиком с помощью термовесов или термопринтеров. Печать термочеков значительно облегчает работу крупных магазинов, в частности во время большого потока клиентов. Типография «Неопринт» предлагает печать термочеков различных размеров, форм, дизайна по желанию заказчика.

8. *Изготовление оригинал-макета и флексоформ*. **Оригинал-макетом** в полиграфии называется текстовый, либо графический материал, который прошел обработку у дизайнера или специалиста по допечатной подготовке (верстальщика). Этот макет представляет собой основу для создания печатного изделия. Печатная форма — это важный элемент печатного процесса, очень сильно влияющий на результат печати. Выбор материала, на основе которого изготавливается печатная форма, зависит от технологии печати, применяемой на производстве. Изготовление флексоформ осуществляется на фотополимерном материале на пленочной подложке.

9. *Этикетки со штрих кодом*. Компания «Неопринт» предлагает изготовление этикеток со штрих кодом на высокоточном оборудовании, которое легко справится с задачей нанесения информации в закодированном виде. Материал и краска готовой продукции может существенно варьироваться в зависимости от того, на какую поверхность необходимо поместить этикетки со штрих кодом.

10. *Флексографическая печать*. Благодаря особым краскам (на водной основе, на основе летучих веществ и т.д.) флексографическая печать прекрасно подходит для совершенно невпитывающих материалов, таких как пленка, фольга и др. Флексографическое оборудование позволяет сэкономить время, ресурсы и бюджет, тем самым делая флексопечать оптимальным решением.

11. *Создание этикеток собственного производства и доработка макета*. В типографии можно купить любой вид печатной продукции. Специалисты компании разрабатывают оригинальный дизайн за короткий срок, гарантируя создание красивой разноцветной этикетки. Доработка макета необходима для выпуска качественной продукции, т.к. позволяет максимально точно смоделировать будущий печатный оттиск.

Для постоянных клиентов и в случае крупных заказов компания «Неопринт» предлагает скидки.

Как говорилось ранее, главной заботой компании является стремление стать надёжным долговременным партнёром, основывая отношения на тесном взаимовыгодном сотрудничестве и используя индивидуальный подход к каждому клиенту. Поэтому немаловажное значение для типографии имеет организация работы с клиентами, поддержание отношений с важными для нее партнерами. Эти обязанности выполняет менеджер по работе с клиентами, и от эффективной организации его труда во многом зависит успешная деятельность всего предприятия.

Менеджер по работе с клиентами в ООО «Неопринт» является «лицом» компании, прежде всего с ним общаются клиенты, партнеры, люди, заходящие в типографию. На него негласно возложена ответственность за поддержание имиджа предприятия. Немаловажную роль в работе на этой должности играет соблюдение менеджером этического кодекса поведения — соблюдение правил хорошего тона, вежливость, тактичность, уважительность в общении, т.е. необходимо убедить клиента в перспективах сотрудничества, не нарушая при этом основные принципы деловой этики: корректность, ненавязчивость и терпение. Именно поэтому особое значение имеет человеческий фактор, а именно личностные характеристики менеджера (эрудиция; умение принимать решения в нестандартных ситуациях, вести переговоры и анализировать полученную информацию о потенциальном клиенте, вникать в потребности клиентов с целью предложения ему наиболее оптимальных, взаимовыгодных условий сотрудничества; коммуникабельность; стремление к повышению своего профессионального уровня).

Основная его задача состоит в том, чтобы помочь клиенту выбрать и приобрести то, что предлагает компания. Менеджер более детально работает с заказчиками непосредственно по изготовляемой продукции, подсчитывает итоговую сумму заказа, а также является связующим звеном с другими подразделениями и специалистами типографии, непосредственно реализующим заказы – фотографами, рекламщиками, дизайнерами и др. В его обязанности входит реализация полиграфической продукции и услуг, консультирование клиентов, оформление договоров и счетов, оформление и ведение соответствующего документооборота.

Менеджер по работе с клиентами не только предлагает варианты форм сотрудничества, но и сопровождает совместную работу, участвует в обсуждении итогов, оценивает перспективу дальнейшего взаимодействия. Залог успешной работы менеджера по работе с клиентами — индивидуальный подход к каждому заказчику и нацеленность на долгосрочное сотрудничество, основанное на взаимном доверии и уважении.

Организация работы менеджера по работе с клиентами в ООО «Неопринт» представлена на рис. 1.2. Принимая заказ от клиента, менеджер обеспечивает контроль его выполнения и является связующим звеном между клиентом и другими подразделениями компании, ответственными за выполнение заказа – отделами управления запасами (складом), планирования производства и непосредственно производства (дизайнерский и печатный отделы). Основной задачей АРМ менеджера является то, что вся информация о заказе и пожеланиях клиента, занесенная менеджером в автоматизированную базу данных, должна сразу отображаться в базах данных этих отделов в порядке очереди по выполнению заказов, что будет значительно экономить время и ресурсы на подсчет затрат на выполнение заказа и передачу информации о нем между структурными подразделениями предприятия.



Рис. 1.2. Организация работы менеджера по работе с клиентами

Должностные обязанности менеджера по работе с клиентами зависят от специфики деятельности компании. В компании ОО «Неопринт» основными обязанностями менеджера по работе с клиентами являются:

1. осуществление анализа аудитории потенциальных клиентов, выявление потребностей клиентов, их категорий и направленности;
2. планирование работы с клиентами;
3. координация мероприятий по привлечению клиентов;
4. прогнозирование деловой надежности потенциальных клиентов;
5. предварительные переговоры с клиентами, заинтересовавшимися услугами компании, уточнение их потребностей и подготовка предложения для конкретных клиентов;
6. обсуждения проектов договоров с вышестоящим руководством, согласование разногласий в пределах своей компетенции;
7. помощь клиентам при решении несогласованных вопросов при переговорах или возникших после совершения юридически значимых действий;
8. заключение договоров от имени предприятия;
9. поддержание связи с постоянными клиентами, мотивация их для дальнейшего сотрудничества;
10. разработка особых схем работы с наиболее выгодными предприятию клиентами (особенные условия договоров, системы скидок, придание статуса «VIP», ускоренные сроки исполнения обязательств со стороны предприятия и т.д.);
11. предоставление клиентам своевременной информации о новых возможностях и предоставлении новых услуг типографии;
12. обеспечение соблюдения интересов клиентов при выполнении условий договоров подразделениями предприятия;
13. обеспечение обратной связи с клиентами (изучение их требований к товарам и услугам, установление причин неудовлетворенности клиента, анализ претензий клиентов и принятие мер по их удовлетворению и пресечению впредь, а также мер по сохранению деловых связей);
14. формирование и ведение банка данных о клиентах (клиентской базы), своевременное внесение в него изменений;
15. изучение и анализ политики конкурентов во взаимоотношениях с клиентами;
16. составление отчетов, содержащих результаты своей деятельности и пожелания клиентов, а также внесение предложений руководству по совершенствованию работы предприятия с клиентами.

Приведенные должностные обязанности взяты в качестве основы должностной инструкции менеджера по работе с клиентами в ООО «Неопринт». При разработке инструкции для конкретного менеджера включены обязанности, обусловленные спецификой работы компании.

1.3. Обзор существующих разработок в сфере автоматизации документооборота полиграфического предприятия

В основе современных интегрированных систем автоматизации лежит использование компонентов, которые удовлетворяют таким свойствам, как открытость, стандартизация, масштабируемость, типизация решений, комплексность подхода, тиражируемость.

На российском рынке информационных технологий представлены системы автоматизации, отличающиеся друг от друга по классу и идеологии, построенные на несовместимых друг с другом протоколах, интерфейсах и контроллерах. В настоящее время на рынке компьютерного программного обеспечения существует достаточное количество различных программных продуктов, предназначенных для автоматизации работы типографий различных масштабов. Основные из них будут рассмотрены ниже.

- 1С: Полиграфия 8

- 1С: Оперативная печать 8

- 1С: Широкоформатная печать 8

- ERP SyteLine

- Infor ERP SyteLine+ Data collection

Программные продукты **1С: Полиграфия 8, 1С: Оперативная печать 8 и 1С: Широкоформатная печать 8** - совместные отраслевые решения фирм «1С» и «Армекс». Основное назначение указанных программных продуктов – автоматизация целого ряда участков полиграфического производства. Для каждого участка определяются сроки выполнения работы, необходимые ресурсы. Программы являются полноценными коммерческими продуктами и требуют покупки лицензии на их использование. Это является основным недостатком подобных систем.

**1С: Полиграфия 8.** Флагманский продукт «1С: Полиграфия 8» - входит в линейку отраслевых решений на базе «1С: Предприятие 8. Управление производственным предприятием». Данная система необходима для решения задач общего плана связанных с логистическими вопросами в управлении полиграфическим предприятием. Так с помощью Полиграфии 8 формируется единое информационное пространство, которое доступно всем отделам предприятия. Это дает возможность оперативно реагировать на изменения в заказе. Этот программный продукт ориентирован на достаточно крупные полиграфические предприятия.

**C: Оперативная печать 8. «**1С: Оперативная печать 8» является облегченным вариантом программного продукта «1С: Полиграфия 8» и предназначен в основном для автоматизации небольших типографий, салонов оперативной полиграфии, копи-центров. Данный продукт ориентирован на небольшие типографии, которые не требуют комплексного подхода. Однако, он также является платным, что требует расходов на покупку лицензии на каждый офис. Кроме того необходимо выделение ресурсов на обучении персонала работы со всеми указанными программными комплексами, а также на их установку.

Все бизнес-операции, происходящие в типографии, можно отнести к одному из трех видов - коммерческие операции, производственные операции и управленческие операции. Менеджеры отдела продаж контролируют исполнение своих заказов на всех участках производства, поэтому помимо доступа к коммерческой функциональности они должны иметь возможность контролировать состояние и график прохождения своих заказов через производство, но без возможности прямого вмешательства в этот график.

**ERP SyteLine** от компании «Фронтстеп СНГ». Система Infor ERP SyteLine этот программный продукт предназначен для ведение склада и управления ресурсами. Система ориентирована на производителей упаковочных материалов, тары, а также полиграфической продукции. Система построена из ряда модулей – модуль управления, модуль контроля закупок и др.

Автоматизированная система управления полиграфическим предприятием позволяет выполнять:

* добавление заказов клиентов.
* Проверка платежеспособности.

Модуль конфигурирования включает в себя предоставление таких функций как:

* Формирование оригинал-макетов с учетом выбранных пользователем характеристик на основании шаблонов.
* создание технологического маршрута (препресс - листовая, ролевая, газетная и др. виды печати - отделка и брошюровка - экспедирование) под конкретную заявку клиента.
* Имеет режим создания производственных заданий, технологической карты для каждого заказа клиента через использование "перекрестной ссылки" между заказом клиента и производственным заданием, позволяющей отслеживать на предприятии каждый заказ клиента во временных, количественных, стоимостных и качественных разрезах;

На выходе позволяет получить: технологические карты и производственные задания, соответствующие заказам клиента с параметрами (количество, дата производства, входящие материалы, маршрут производства); стоимость заказа и ориентировочные сроки выполнения.

Планирование и управление закупками - Infor SyteLine APS: оптимизационное планирование производственной и закупочной деятельности с одновременным учетом всех необходимых ресурсов (материальные ресурсы, рабочая сила, мощности оборудования, объем складов и т.д.). Данная функциональность позволяет также оперативно встраивать новые заказы клиентов в существующую производственную программу с учетом различных ограничений производственного процесса; расчет потребностей в сырье и материалах на основании плановых заказов клиентов и складских запасов.

* формирование графика платежей.
* формирование заказов поставщикам;
* акцептование заказов и отслеживание задолженности перед поставщиками.

Управление запасами - Infor ERP SyteLine+ Data collection: отслеживание приходов/расходов от поставщиков, передачи в цех, отгрузки заказов клиентов. Использование штрих-кодирования.

Все рассмотренные программные продукты ориентированы на самостоятельные типографии и почти непригодны для внутренних типографий, в которых часть бизнес-процессов отсутствуют. Приобретение любого из этих программных продуктов для типографии не всегда будет нецелесообразно. Необходимо разрабатывать автоматизированную систему учитывающую специфику работы типографии.

1.4. Постановка задачи проектирования

Целью данного дипломного проекта является разработка АРМ менеджера по работе с клиентами, создание базы данных клиентов и поставщиков, оперативного подсчета стоимости заказа и передачи информации о нем другим подразделениям компании, ведения учета доходов и расходов и хозяйственных операций типографии.

Разработка и внедрение автоматизированных рабочих мест - это процесс, связанный с определенными капиталовложениями, а также текущими затратами при их эксплуатации. Поэтому прежде необходимо проанализировать затраты ресурсов и экономический эффект (результат) от автоматизации рабочего места. Это обосновывает правомерность выделения денежных средств на разработку проекта, выбора экономически наиболее выгодного варианта, определение последовательности перевода комплексных задач на автоматизированное решение и дает возможность оценить влияние внедрения проекта на показатели хозяйственной деятельности предприятия.

Ожидаемыми результатами реализации автоматизации рабочего места менеджера по работе с клиентами в ООО «Неопринт» являются:

* сокращение бумажного документооборота и упорядочение документации;
* повышение производительности труда;
* сокращение времени обработки входных документов и получения конечных результатов;
* наличие базы клиентов, которые пользуются услугами типографии, и базы ее поставщиков;
* получение оперативных данных для принятия управленческих решений;
* уменьшение трудоемкости составления отчетности.

Функциональные основные возможности разработанной программы:

* регистрация клиентов, поставщиков;
* получение оперативной информации о наличии материалов;
* предоставление разнообразных отчётов, в том числе таких как «Список заказов» и «Список продаж»;
* вывод всех необходимых отчётов в установленном формате, либо сразу на печать.

Проект должен обладать такими характеристиками. надёжность, эффективность, простота в работе, сопровождаемость.

При внедрении проекта будут автоматизированы следующие функции управления:

- автоматическая проверка списка клиентов и поставщиков;

- автоматическое ведение баланса, который отображает наименование товара с указанием его цены.

- автоматическая система документооборота, а также выдача необходимой аналитической информации для оперативного и стратегического управления заказами для менеджера;

Назначением реализации подсистемы является решение следующей задачи – автоматизация списка клиентов и поставщиков, автоматизация документооборота, операций подсчета стоимости заказа, контроля и сопровождения по его выполнению.

1.5. Обоснование необходимости использования информационных технологий в полиграфической деятельности

От того, каким образом организовано рабочее место, во многом зависит эффективность работы менеджеров всех категорий независимо от той должности, которую они занимают. Задача рациональной организации рабочих мест менеджеров включает в себя не только оборудование его всем необходимым в соответствии с содержанием и характером выполняемой работы, рациональное его размещение, создание комфортных условий труда, но и его программное обеспечение. Лучший вариант организации рабочего места, как и в других сферах деятельности, может быть определен только при условии детального анализа той работы, которая выполняется менеджером.

Рабочее место менеджера по работе с клиентами в ООО «Неопринт» представляет собой часть общего помещения, в котором происходит трудовая деятельность работников типографии, оборудованном в соответствии с содержанием выполняемых работ необходимыми орудиями труда. Правильная организация рабочего места позволяет менеджеру рационально, с наименьшими затратами труда выполнять свои функции; эффективно общаться с заказчиками, подчиненными и другими работниками; поддерживать высокую работоспособность и рабочее настроение.

АРМ менеджера представляет собой его рабочее место, оснащенное ПК и представляющее собой самостоятельный программно-технический комплекс, позволяющий вести обработку информации, получать все необходимые выходные данные для формирования заказа, расчета его стоимости, взаимодействовать с другими подразделениями компании по дальнейшему выполнению заказа и контролю этого процесса. Оно должно исключать затраты времени, связанные с нецелесообразным перемещением, обеспечивать условия для рационального выполнения менеджером своих функций и экономное использование площади служебного помещения. Эффективность современного АРМ менеджера во многом определяется комплексом программных продуктов по обработке и передаче информации.

Все работы, выполняемые менеджером по работе с клиентами ООО «Неопринт», непосредственным образом связаны с обработкой, анализом и созданием документов, что порождает весьма объемный документооборот, который основан на устаревших принципах делопроизводства. Основными недостатками существующей технологии являются:

1) низкая оперативность доступа к информации; снижающая качество управления объектом, И наличие сбоев при взаимодействии компонентов информационной системы;

2) отсутствие надежной связи между сотрудниками; затрудненность мониторинга работы сотрудников;

3) высокая трудоемкость обработки информации; чрезмерная трата денег на бумагу и расходные материалы.

4) несовершенство процессов сбора, передачи, обработки, хранения и процессов выдачи результатов расчетов конечному пользователю

В ходе написания дипломного проекта, были выявлены явные недостатки, которые создавали неудобства в работе компании ООО «Неопринт» и замедляли скорость обслуживания клиентов:

1) отсутствие базы данных клиентов и базы поставщиков; высокая трудоемкость обработки информации;

2) сложность и низкая оперативность подсчета стоимости заказа;

3) сложность формирования ежедневных и ежемесячных финансовых отчетов работы предприятия.

Это влечет за собой усложнение и увеличение непроизводственных затрат труда сотрудников, что отрицательно сказывается на результатах деятельности компании. Таким образом, автоматизация документооборота для усовершенствования всех его составляющих, а также составление базы поставщиков и клиентов для более оперативной работы менеджера и связи его с другими сотрудниками является актуальной задачей компании.

Внедрение и совершенствование АРМов повышает качество управленческих решений, выполнения заказов, что достигается за счет повышения общего уровня качества, полноты информации, имеющейся на предприятии и предоставляемой пользователям, сокращения срока прохождения информации по полному циклу ее обработки. При реализации АРМ автоматизируются бизнес-процессы и документооборот предприятия, появляется возможность вводить статьи учета, которые в ручном режиме реализовать практически невозможно. Перечень входных документов очень велик и ручной ввод входной информации иногда по существу одних и тех же данных замедляет скорость документооборота и увеличивает трудовые затраты. Так как фирма занимается изготовлением полиграфической продукции, у нее имеется ряд входных документов (бланк заказа; счет, счет-фактура и накладная от поставщиков) и выходных документов (счет на оплату заказа, счет-фактура, накладная для заказчика; прайс на изготовление своей продукции).

Для подсчета итоговой суммы для клиента используется MS Excel, что крайне неудобно, проблематично и занимает много времени. Такие подсчеты могут привести к неточности. Планируется автоматизировать систему подсчета и схему документооборота.

Схема заказа реализуется следующим образом: на фирму звонит заказчик и делает заказ. Из всех этих факторов складывается цена за комплект визиток. Либо если это другая печатная продукция - формат изделия; кол-во изделий/тираж; формат бумаги; кол-во на печатном формате; кол-во листов; цветность; плотность бумаги; вид бумаги; дополнительные работы.

Автоматизация заказа позволит ускорить документооборот, сократить ручные операции и проводить анализ деятельности предприятия.

1.6. Формальная модель системы автоматизации работы с клиентами

Для оценки эффективности внедрения автоматизированной системы выполним построение формальной математической модели. Введем обозначения для основных параметров модели используя терминологию теории систем массового обслуживая [21, с. 134].

Пусть, в системе работает три оператора по приему (отгрузке) заказа клиента, и офис одновременно может обслуживать только трех клиентов, тогда работу системы можно рассмотреть как 3-канальную систему массового обслуживания с ожиданиями и ограниченной очередью длины 3. Обозначим:

- абсолютная пропускная способность – А

- относительная пропускная способность Q

- интенсивность обслуживания 

- вероятность отказа 

- среднее время пребывания заявки в системе 

- среднее число обслуживаемых заявок 

- среднее число заявок в системе 

- среднее время пребывания заявки в очереди 

- среднее время обслуживания одной заявки 

Пусть в офис, в среднем поступает n запросов в час на расчет стоимости заказа, тогда интенсивность потока  (задач в час).

Введем аналогичные обозначения со штрихами для показателей системы после внедрения автоматизированной системы.

Время обслуживания одной заявки или  зависит от количества услуг в корзине заказа и может быть представлено в виде

,

где  количество заказов в одной партии,  - время в минутах на выполнение элементарной операции по расчету стоимости услуги.

Возможные состояния системы находятся исходя из длины очереди и количества каналов.

 - заявок нет, все каналы свободны;

 - занят один канал, два канала свободны;

 - все каналы заняты;

 - все каналы заняты, одна заявка ожидает в очереди;

 - все каналы заняты, две заявки в очереди;

 - все каналы заняты, три заявки в очереди;

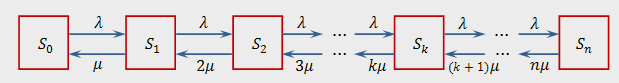


Рис. 1.3. СМО с очередью

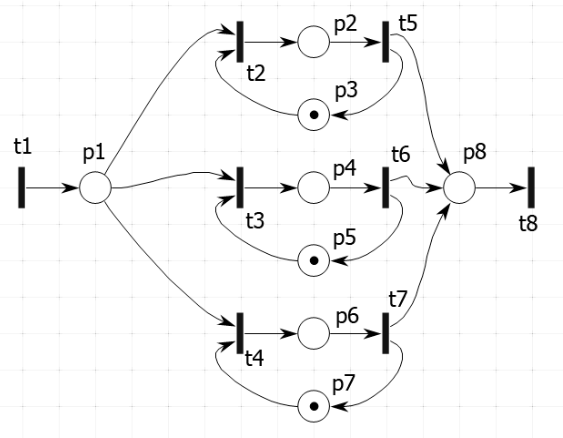


Рис. 1.4 Сеть Петри СМО

В построенной сети Петри - событие поступление заявки к менеджеру,  - завершение операции формирования корзины заказов, - занятие и освобождение рабочих станций менеджеров.  - метки свободного канала.

1.7. Обоснование выбора СУБД

Поскольку наибольшее распространение среди пользователей многих стран имеют операционные системы семейства Windows, то целесообразна разработка компьютерного приложения именно под данную операционную систему.

Операционная система имеет ряд библиотек, которые используют Windows-приложения, и в связи с этим обусловлена специфика их разработки.

Следует выделить несколько видов инструментальных средств, которые используются для разработки приложения по управлению тренировками.

Первый вид инструментальных средств - это средства для разработки и первичного заполнения базы данных.

Выбор СУБД обусловлен в первую очередь способом взаимодействия с базой данных и количеством потенциальных клиентов базы данных. Так как к базе данных будут подключаться только менеджеры по работе с клиентами, т.е. очень небольшое число пользователей, то скорость доступа не будет играть решающего значения.

С другой стороны, необходимо обеспечить возможность легкого редактирования базы данных для сотрудников, это необходимо при внедрении автоматизированной системы управления и минимизировать финансовые затраты.

В целях обеспечения наибольшей мобильности и легкой настройки и установки приложения в качестве СУБД приложения была выбрана СУБД MS Office Access 2007.

Данная СУБД позволяет быстро заполнить базу данных и установить связи между сущностями, а также в случае невозможности запуска приложения получить доступ ко всей информации в виде таблиц и запросов.

1.8. Обоснование выбора среды разработки

Основной средой разработки приложения выбрана среда RAD STUDIO Embarcadero Delphi XE. Данная среда разработки имеет все необходимые драйвера для связи с выбранной СУБД, имеет современный набор визуальных компонент и компонент для обеспечения связи с базой данных. Кроме того, выбор данной среды разработки позволяет построить качественное приложение за минимальное время разработки, что снижает прямые затраты на выполнение разработки.

Таким образом для проектирования и выполнения разработки использованы следующие инструментальные средства:

* ОС Windows 7;
* СУБД MS Access;
* Embarcadero RAD Studio Delphi XE;
* MS Office Excel.

2. Проектная часть

2.1. Информационное обеспечение деятельности полиграфического предприятия

2.1.1. Проектирование концептуальной модели базы данных

Концептуальная модель позволяет выявить основные сущности и их атрибуты, которые должны отражать всю схему взаимодействия между потоками информации. В ней можно выделить несколько ключевых блоков.

1. Взаимодействие клиента и менеджера. Результатом данного взаимодействия является формировние корзины заказов и сопутствующей документации.

2. Взаимодействие директора с базой данных. Директора фирмы формируют ценовую политику путем внесения параметров и цен в базу данных, соответствующие услугам.

3. Взаимодействие между бухгалтерией и приложением АРМ. После оплаты заказа в локальной сети предприятия сохраняются данные с корзиной клиента. Данный документ может быть далее подан для анализа затрат и движения денежных средств.

4. Формируемые документы могут быть использованые и для анализа будущих закупок материалов.

Таким образом, для информационной системы инициирующей информацией служит информация от клиента, которая заполняется в программе менеджеров, после этого осуществляется распространение информации по отделам.

Перед созданием различных категорий моделей и физической структуры базы данных необходимо четко представлять ее взаимосвязи с различными видами моделей (рис. 2.1).

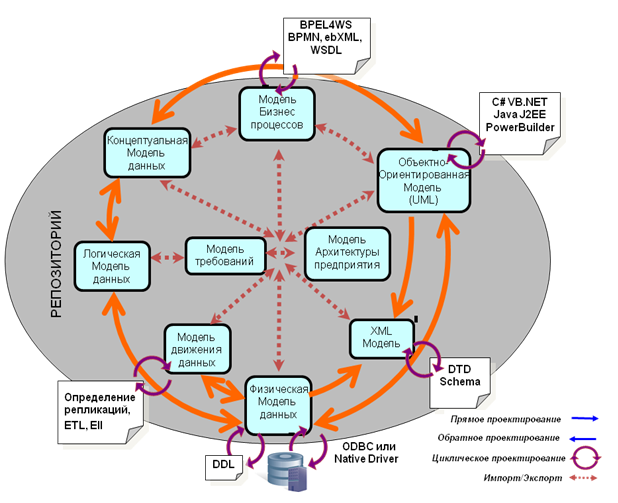


Рис. 2.1. Взаимосвязь между моделями

Анализ предметной области показал, что при построении концептуальной и логической модели будут использованы следующие сущности:

- Сотрудник;

- Клиента;

- Формат;

- Формат пленки;

- Формат чека;

- Вид лакировки;

- Вид бумаги;

- Вид тиснения;

- Цветность;

- Плотность;

- Способ доставки;

- Статус оплаты;

- Тип продукции (услуги);

- Параметры;

- Корзина;

Построим логическую модель в программе Power Designer

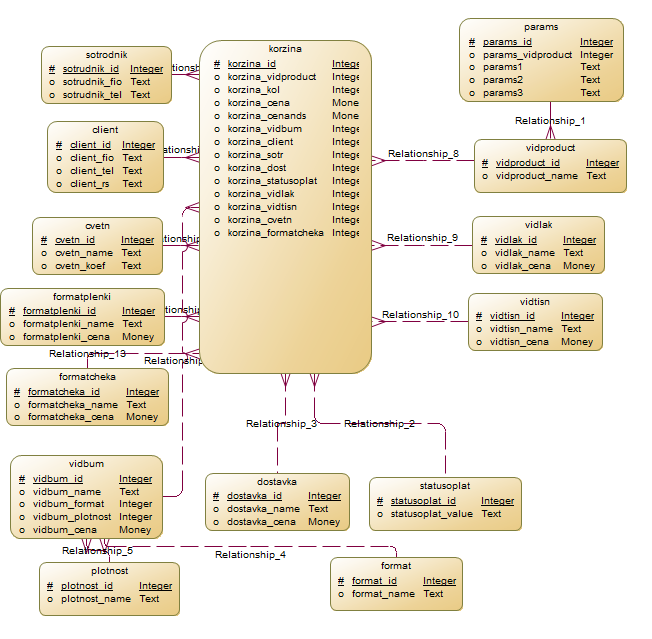


Рис. 2.2. Концептуальная модель базы данных

2.1.2. Проектирование физической модели базы данных

После выполнения проверки модели выполним генерировании PDM-модели (Physical Data Model).

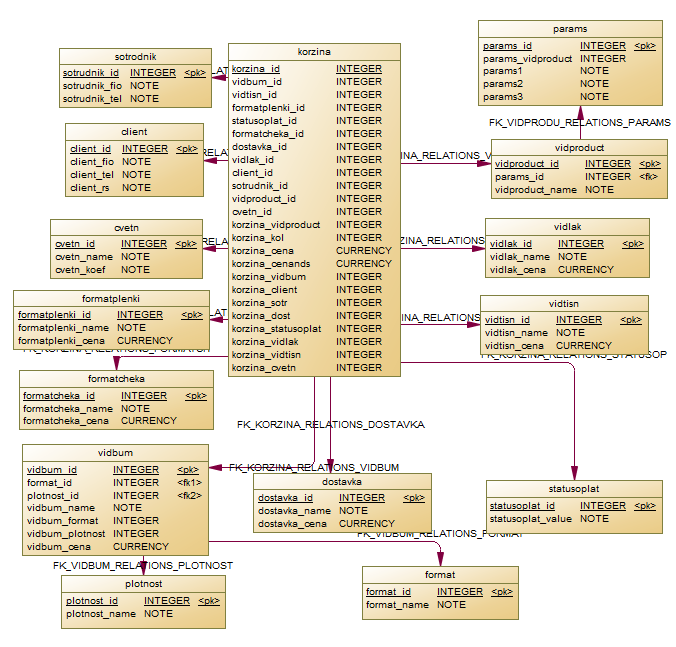


Рис. 2.3. Физическая модель

2.1.3. Описание связей и полей

Разработка физической структуры выполняется на основании физической модели. Каждая из сущностей представлена таблицей

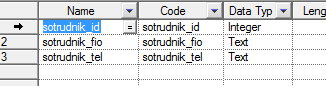


Рис. 2.4. Сотрудники

Таблица сотрудники (рис. 2.4.) имеет поля для хранения ФИО сотрудника, телефона.

Таблица Клиенты имеет аналогичную структуры и содержит еще информацию о расчетном счете клиента.

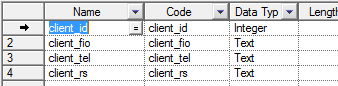


Рис. 2.5. Клиенты

Ряд таблиц содержат информацию для расчета заказа это таблицы цветность, формат пленки, формат чека, формат бумаги.

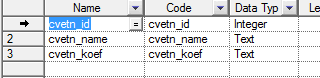


Рис.2.6. Цветность

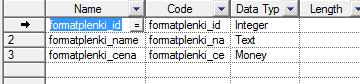


Рис. 2.7. Таблица формат пленки

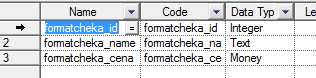


Рис.2.8. Таблица формат чека

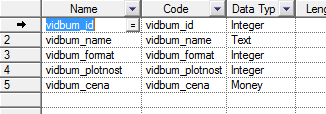


Рис. 2.9. Таблица формат бумаги

Способы доставки сохраняются в таблице Доставка

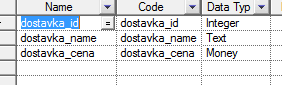


Рис. 2.10. Таблица доставка



Рис. 2.11. Таблица статус оплат

Таблица статус оплат необходима для хранения разных статусов, которые может иметь заказ в корзине (оплачен, ожидается оплата, не оплачен, отказ).

Вид лакировки, вид тиснения используются для расчета стоимости соответствующих услуг.

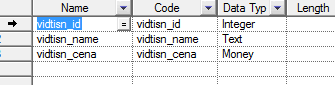


Рис. 2.12. Таблица вид тиснения

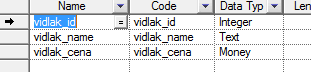


Рис. 2.13. Таблица вид лакировки

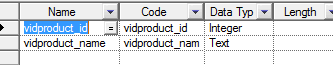


Рис. 2.14. Таблица вид продукции

Вид продукции содержит перечень услуг и видов продукции. Эта информация указывается при печати накладной.

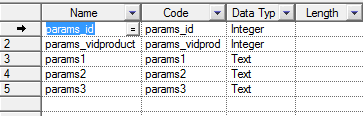


Рис. 2.15. Таблица параметры

Таблица Параметры содержит ряд параметров, которые необходимы для расчетов, например, стоимость реза, запас на брак, наценка и т.п. Количество параметров может быть при необходимости увеличено.

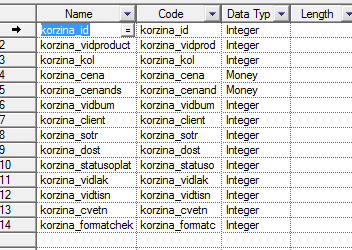


Рис. 2.16. Таблица корзина

Основной таблицей является таблица Корзина, с ней связаны связью один ко многим остальные таблицы. В таблице Корзина указываются все данные о заказе: клиент, сотрудник, вид заказа, статус, все параметры заказа.

2.2. Программная реализация приложения

2.2.1 Проектирование интерфейса

Основной задачей, которую решает разработанная система – расчет стоимости заказа и формирование корзины заказов для клиента.

Для организации АРМ, кроме функции расчета стоимости, система имеет следующие возможности:

- ведение учета заказов, формирование архива заказов в виде отдельных файлов оплаченных корзин и общего файла;

- расчет количества бумаги для выполнения заказа;

- вывод на печать бланков к оплате с реквизитами предприятия и данными заказа;

- редактирование ценовой политики указанием рентабельности;

- отслеживание запасов товара;

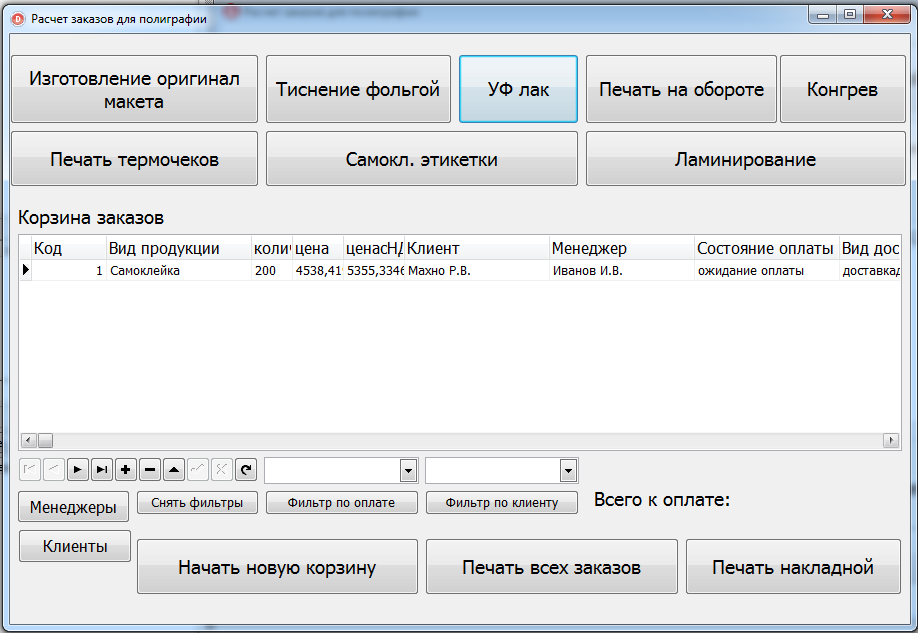


Рис. 2.17. Окно приложения

Основная задача приложения – реализация функций АРМ менеджера.

На главное окно имеет несколько областей. В верхней части окна – производится выбор вкладки, определяющей интерфейс для ввода всех параметров необходимых для расчета цены.

В целях улучшения совместимости, в форме используются только стандартные элементы управления, не требующие подключения дополнительных библиотек, такие как Button, checkbox, combobox, dbtext, label, optionbutton, dbgrid.

В нижней части окна каждой формы расположены функциональные кнопки для выполнения расчета и печати заказов.

После выполнения расчета необходимая информация выводится на экран в корзину заказов главного окна.

Программа имеет эргономический интерфейс и предусматривает различные диалоговые окна для расчета стоимости различных видов полиграфических услуг.

Рассмотрим интерфейс диалоговых окон.

**Форма самоклеящиеся этикетки**

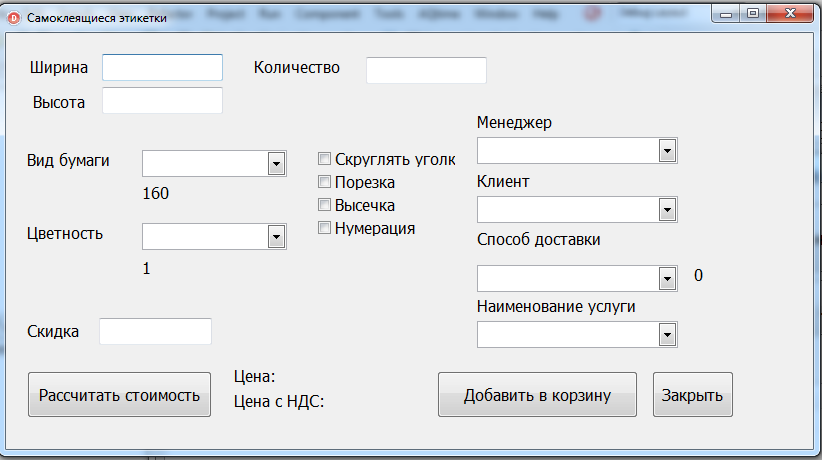


Рис. 2.18. Форма для расчета стоимости этикеток

Небольшие самоклеящиеся этикетки могут быть наклеены не только на товаре, но и в качестве рекламы на улице, они играют роль паспорта продукта, в котором содержатся данные для электронных устройств считывания. При этом этикетки должны привлекать внимание покупателя и содержать всю необходимую информацию.

Менеджер должен учитывать различные пожелания заказчика, а также ориентироваться в ассортименте и наличии бумаги на складе. Для удобного расчета форма имеет несколько полей для ввода параметров.

Для расчета стоимости самоклеящихся этикеток необходимо заполнить все поля – ширина, высота, выбрать вид бумаги, формат листа и цветность. Кроме того можно указать дополнительные операции.

Программа ищет цену на выбранный тип бумаги, также параметры считываются из таблицы Параметры.

В поле Скидка возможен ввод скидки. После этого необходимо нажать кнопку **Рассчитать заказ**. Будет показана стоимость заказа в рублях, и стоимость с НДС.

Если стоимость устраивает клиента, то необходимо нажать кнопку **Добавить в корзину**.

Бланк для оплаты формируется автоматически считыванием заказов из корзины. Печать бланка выполняется нажатием соответствующей кнопки в нижнем функциональном меню.

**Вкладка Конгрев**

Для расчета стоимости конгревного тиснения необходимо заполнить все поля. Цена существенно зависит от вида тиснения.

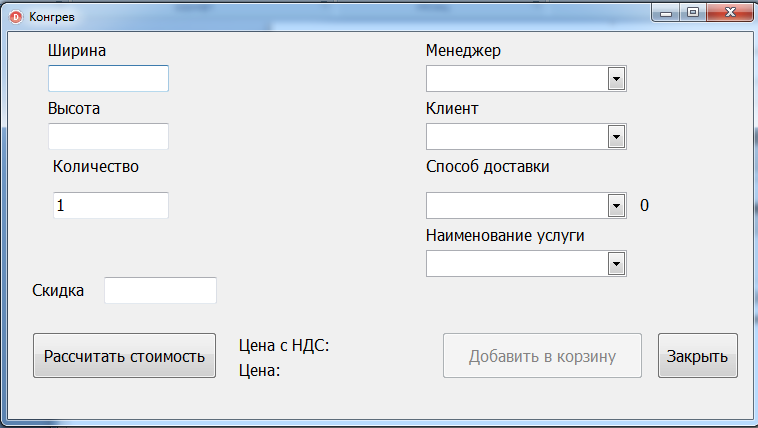


Рис. 2.19. Окно Конгрев

Конгревом можно достичь дополнительных визуальных эффектов в художественном и дизайнерском смысле. Многоуровневый конгрев дает эффект трехмерности, позволяющей создать нечто отличное от плоского оттиска. Но цена на такой конгрев будет дороже.

Конгрев помогает выделить необходимые детали рисунка. Технологический прием конгревной обработки печатной продукции позволяет ощутимо отличить носитель информации от изделий с плоской поверхностью. Конгревное тиснение можно выполнять как при помощи фольги, так и при ее отсутствии. Менеджер должен заполнить все поля для правильного расчета заказа.

**Форма УФ лак**

Для расчета цены на УФ-лакировку аналогично заполняются поля ширина и высота, обязательно необходимо выбрать вид лакировки.

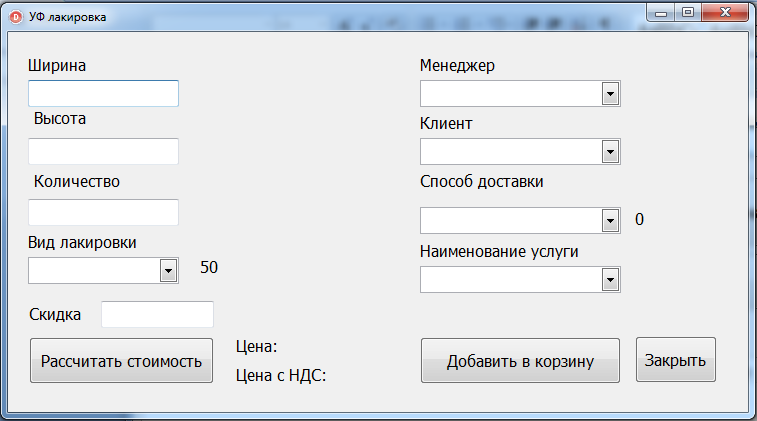


Рис. 2.20. Форма УФ лак

**Форма печать термочеков**

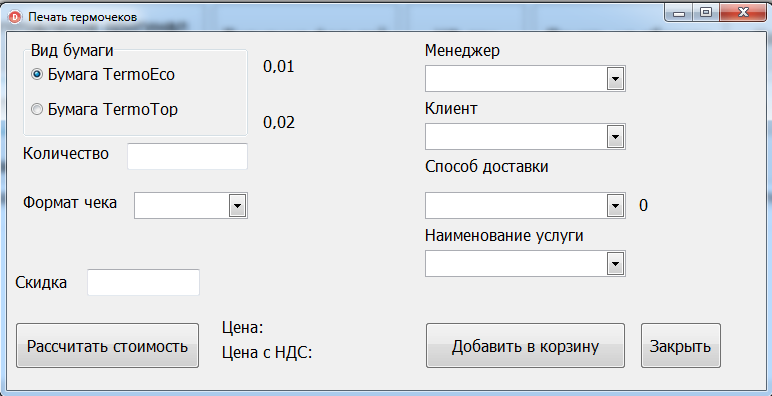


Рис. 2.21. Печать термочеков

**Форма изготовление оригинал-макетов**

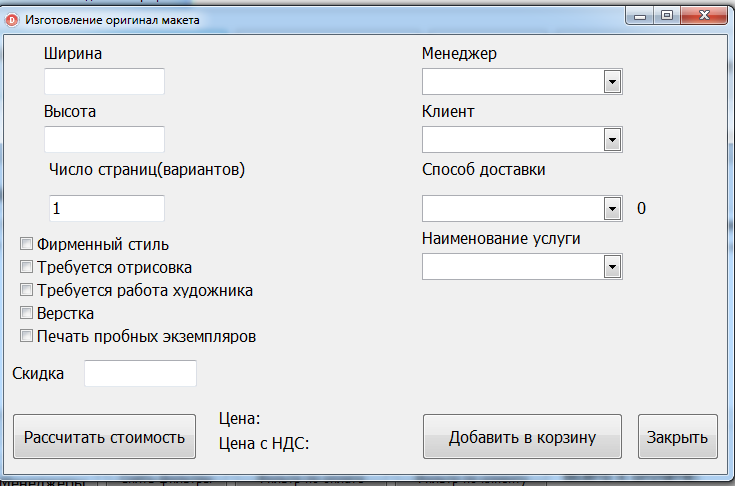


Рис. 2.22. Диалоговое окно для расчета цены оригинал-макета

Оригинал макет представляет собой основу для создания печатного изделия. Дизайнер изготовляет цифровой файл, который затем передают в типографию и по нему изготавливается печатная форма — это важный элемент печатного процесса, очень сильно влияющий на результат печати.

Макеты бывают различной сложности и типа, поэтому для точного оценивания стоимости разработки и трудозатрат дизайнеров менеджер должен как можно более точно узнать у клиента специфические особенности макета и ввести все данные в форму для расчета стоимости.

**Вкладка Печать на обороте**

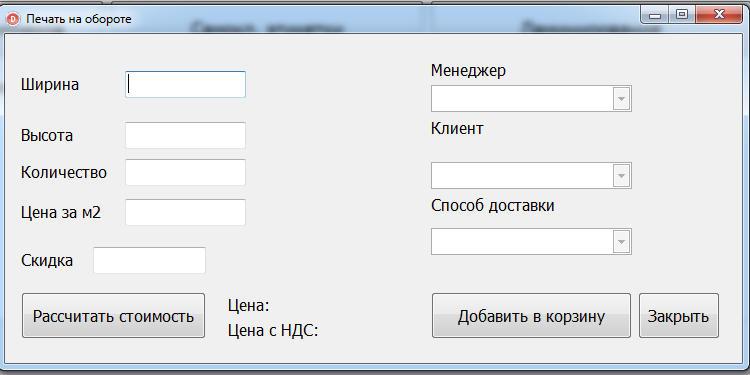


Рис. 2.23. Интерфейс для расчета печати на обороте

**Форма ламинирование**

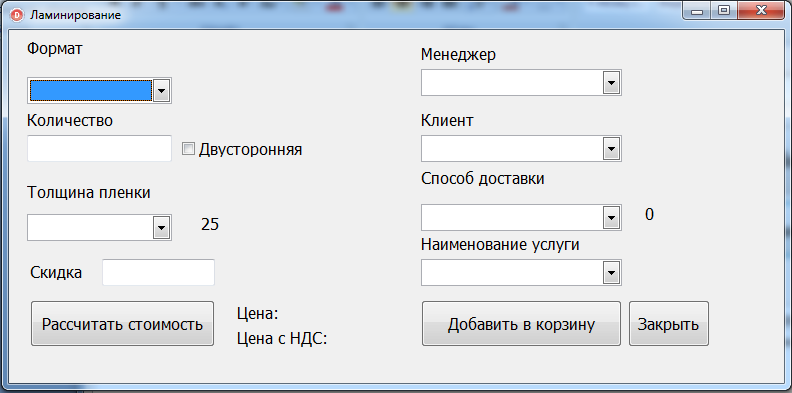


Рис. 2.24. Форма для расчета ламинации

**Форма тиснение фольгой**

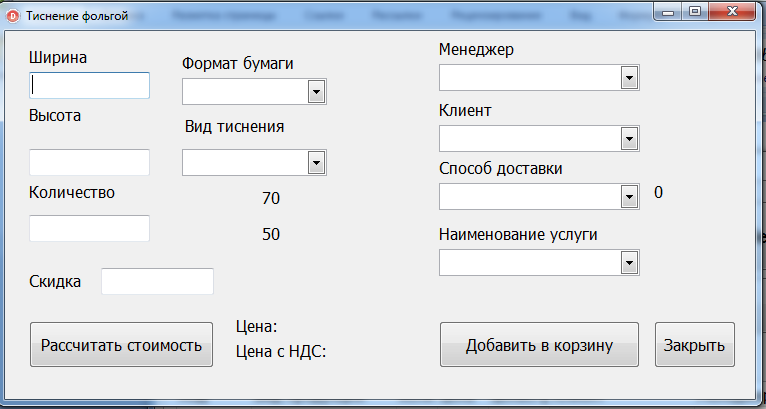


Рис. 2.25. Форма для расчета цены тиснения

Большинство общих параметров для удобства настройки вынесены в базу данных. Это позволяет легко настраивать приложения в случае изменения производственных возможностей.

2.2.2. Взаимодействие с базой данных

Код приложения расположен в модулях экранных форм unit1-unit10, за хранение невизуальных компонент отвечает datamodule2.

Среда разработки Delphi позволяет выполнять построение запросов для обращения к базе данных как с помощью визуального редактора SQL-запросов так и выполняя обращение с помощью программного кода.

База данных в формате MS Access 2007 должна быть расположена в одной папке с приложением. На этапе разработки база данных должна быть расположена как в подпапке \debug\win32\, так и в корневой папке

Для связи с базой данных использованы невизуальные компоненты ADOconnection, ADOTable, ADOQuery, ADODataSorurce.

Данные компоненты осуществляют связь и взаимодействие с базой данных приложения. Компонент ADOconnection необходим для подключения к базе данных, главным параметром подключения является строка инициализации:

Provider=Microsoft.ACE.OLEDB.12.0;User ID=Admin;Data Source=database.accdb;Mode=Share Deny None;Jet OLEDB:System database="";Jet OLEDB:Registry Path="";Jet OLEDB:Database Password="";Jet OLEDB:Engine Type=6;Jet OLEDB:Database Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database Password="";Jet OLEDB:Create System Database=False;Jet OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False;Jet OLEDB:Support Complex Data=False;

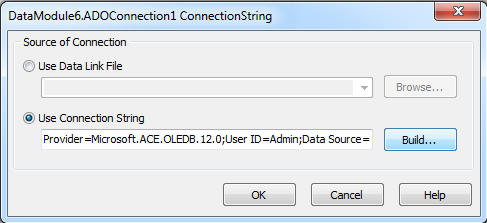


Рис. 2.26. Настройка подключения

В случае переименования базы данных, или расположения базы на сервере необходимо выполнить дополнительную настройку строки инициализации и указать путь к базе данных.

Все компоненты, которые обеспечивают взаимодействие с базой данных, собраны в невизуальном модуле unit2

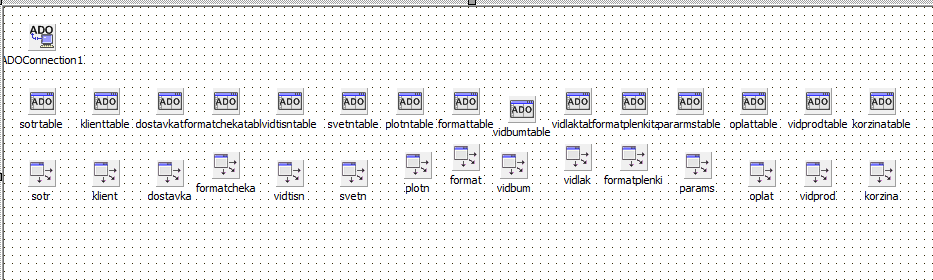


Рис. 2.27. Модуль для хранения компонент

После установки связи с базой данных необходимо выполнить связывание таблиц и компонент. Это может быть выполнено как в визуальной среде разработки, так и с помощью программного кода.

Компонент ADOTable используется для выполнения простого запроса, который отображает все выбранные поля.

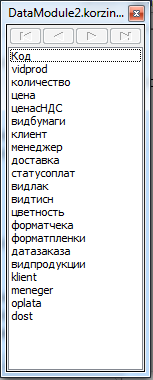


Рис. 2.28. Добавление полей

Для добавления полей также удобно использовать компонент TADOQuery. Данный компонент имеет источник данных DataSource и использует компонент TADOConnection для установки связи с базой данных.

2.2.3. Особенности программной реализации функционала приложения

2.3. Структурная схема приложения



Рис. 2.29. Общая схема алгоритма работы приложения

2.4. Схемы основных алгоритмов



Рис. 2.30. Общая схема алгоритма добавления в корзину



Рис. 2.31 Схема вывода накладных

Для осуществления взаимодействия с приложениями Word, Excel необходимо использование модуля ComObj

И кода вида

WordApp:=CreateOleObject('Word.Application');

appath1:=ExtractFilePath(application.exename ) ;

Wd:=WordApp.Documents.add(appath1+'\tempword.dotx');

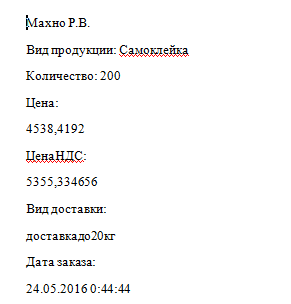


Рис. 2.32. Результат вывода в Word

2.5. Организация процессов обработки информации с использованием приложения

Автоматизация рабочего места менеджера способствует повышению эффективности сбора, обработки и дифференцированного распределения информации, позволяющего каждому сотруднику видеть все процессы, находящиеся в его компетенции и под его контролем, экономии времени и ресурсов.

В следующей таблице детально проведен сравнительный анализ до и после внедрения системы на предприятии. Выгоды от внедрения автоматизации рабочего места менеджера по работе с клиентами в сравнении с возможностями до автоматизации представлены в таблице 3.1.

Таблица 3.1. Сравнение возможностей АРМ менеджера по работе с клиентами типографии

|  |  |
| --- | --- |
| После внедрения | До внедрения |
| 1. Автоматический и быстрый расчет заказов стандартной сложности может происходить интерактивно, можно при непосредственном общении с клиентом. Расчет основывается на внесении первоначальных данных по заказу и после на выборе всех необходимых материалов и технологических операций. | 1. Цена рассчитывается на основании постоянно обновляемого ценника с наиболее востребованными позициями. Нестандартные заказы рассчитываются вручную, привлекаются специалисты, дизайнеры, требуется подпись руководства. |
| 2. Возможность смотреть запасы материалов по складу. В конце дня легко свести все ведомости для оценки потребности в закупке | 2. Наличие материалов на складе узнаётся по факту. Возможны задержки из-за несвоевременной доставки материалов. |
| 3. Автоматический расчет себестоимости заказа и прибыли. В параметрах указывается желаемая рентабельность как наценка. Закладывается система скидок и наценок. | 3. Для расчета рентабельности и параметров требуются отдельные вычисления, возможна потеря прибыли из-за недооценки стоимости выполнения. |
| 4. Автоматическое (на основе расчета) формирование корзины заказов клиента, формирование договоров, счетов и их распечатка. | 4. Составление бланка для оплаты отдельно для каждого клиента вручную. Ручное формирование ведомостей. |
| 5. Автоматическое (на основе расчета) формирование ведомости учета корзин (файлы сохраняются программно). | 5. Выписки нужно сохранять вручную, ручное переименование файлов. |
| 6. Все расчеты сохраняются в едином мести, унификация имен, в конце месяца легко свести отчетные данные. | 6. Типичные заказы считаются быстро, но для нестандартных заказов необходимы дополнительные вычисления, отчетные данные могут теряться. |
| 7. База находится в одном файле на NAS-хранилище, база заказов расположена в сети, есть возможность удаленного входа. | 7. База расположена на локальной машине, нет доступа извне, нет возможности внесения ценовых изменений удаленно. |
| 8. Автоматическое формирование любых отчетов по своей работе за любой период и в любое время. | 8. Самостоятельное ведение статистики по своим заказам и составление отчетов по своей работе. |

Все это в сравнении позволяет менеджеру значительно экономить время, принимать большее количество заказов и обрабатывать гораздо больше информации. Чтобы в этом убедиться, проведем эксперимент расчета времени на ручное формирование заказов и сравним результат с временем автоматического расчета заказа, основанном на внесении первичных данных по заказу и на выборе необходимых материалов и технологических операций.

Для этого рассчитаем время ручного выполнения заказа менеджером со средней производительностью труда, работающего в нормальных условиях производства на принятие и оформление одного заказа для всех видов продукции, предоставляемых типографией, которые более детально были рассмотрены в первом разделе.

Для проведения эксперимента проведем расчет среднего времени ручного заполнения заказов и определения их стоимости по всем видам предоставляемых полиграфических услуг ООО «Неопринт» двух менеджеров по работе с клиентами типографии при обычных условиях труда в течение рабочего дня. Вместе с этим произведем автоматическое заполнение этих же заказов и расчет их стоимости, зафиксировав при этом время выполнения данных операций. Результаты представим в виде таблицы (табл. 3.2).

Таблица 3.2. Расчет времени ручного и автоматизированного заполнения заказа и определения его стоимости.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Вид операции** | | | | | | | | | |
|  | **Самокл. этикетки** | **Конгрев** | **Тиснение фольгой** | **Ламинирование** | **Уф лак** | **Печать термочеков** | **Печать на обороте** | **Изг. оригинал-макета** | **Время на заполнение (секунд)** | **Время на операцию без АРМ** |
| **Количество полей** | 3 | 4 | 3 | 3 | 3 | 1 | 3 | 3 | 5 | 15 |
| **Другие элементы ввода** | 7 | 2 | 3 | 5 | 2 | 2 | 0 | 6 | 2 | 5 |
| **Время ручного расчета** | 80 | 70 | 60 | 70 | 55 | 25 | 45 | 75 |  |  |
| **Время ввода с использованием АРМ** | 29 | 24 | 21 | 25 | 19 | 9 | 15 | 27 |  |  |
| **Разность** | 51 | 46 | 39 | 45 | 36 | 16 | 30 | 48 |  |  |

Как видно из таблицы 3.2. время заполнения заказа и расчета его стоимости с использованием АРМ значительно меньше времени, которое затрачивается на ручное его заполнение и ручной расчет стоимости. Это можно объяснить тем, что на менеджера оказывают влияние многие психофизиологические факторы, которые отсутствуют в работе автоматизированного оборудования.

Также скорость машинной обработки информации может быть значительно выше, чем объем информации, воспринимаемой человеком, т.е. выходить за рамки его физиологических возможностей, и в данном случае автоматизация рабочего места менеджера будет открывать для него дополнительные возможности. Значительная экономия времени будет достигаться и за счет более эффективной координации деятельности всех подразделений и сотрудников полиграфического предприятия. АРМ дает менеджеру по работе с клиентами быстрый доступ к информации, необходимой для принятия заказа, его оформления и определения стоимости, позволяет видеть все процессы, находящиеся в компетенции менеджера или под его контролем.

Данный эксперимент позволяет сделать вывод, что автоматизация расчета позволяет менеджеру рассчитывать стоимость заказа любой сложности даже при непосредственном общении с клиентом с большой экономией времени и ресурсов, и в целом повысит эффективность работы предприятия. Кроме того автоматизация места менеджера по работе с клиентами значительно облегчит работу другим структурным подразделениям (склад, допечатный, печатный и постпечатный отделы, бухгалтерия, логистический отдел) и многим сотрудникам типографии (технологам, дизайнерам, руководителю предприятия, начальнику производства, мастерам печатного цеха, пост-пресса и допечатной подготовки), делая для всех прозрачной текущую ситуацию на производстве.

В целом автоматизация работы менеджера по работе с клиентами позволит достигнуть большей экономии времени в работе других структурных подразделений и сотрудников типографии, а также предоставит новые возможности в их работе (табл.3.3).

Таблица 3.3. Возможности, предоставляемые АРМ менеджера по работе с клиентами другим участникам производственного процесса типографии

|  |  |
| --- | --- |
| Участник производственного процесса | Возможности с АРМ |
| Печатник, резчик, дизайнеры | 1. Сотрудники видят корзину заказов в общей папке и быстро получают файлы с техническим заданием.  2. Возможность самостоятельно планировать изготовление заказов в течении рабочего дня.  3. Загрузка макетов через локальную сеть без необходимости личного присутствия в офисе.  4. Возможность сделать отчет по выполненным заказам и по любым видам работ  5. Доступ к уже выполенным заказам для анализа затрат и эффективного использования остатков материалов (если заказов много, то планирование сборных тиражей, с учетом формата оборудования) |
| Склад | 1. В начале дня по ведомостям оплаты возможно отслеживание запасов.  2. Удобное планирование закупок, экономия на объемах закупок.  3. Удобство взаимодействия с службой доставки. |
| Бухгалтер | 1. Ведомости корзин имеют удобную форму анализа.  2. Быстрое формирование списка проплат  3. Эффективный поиск в базе оплаченных корзин. |
| Директор предприятия | 1. Удобный и понятный расчет зарплаты сотрудников  2. Доступ к базе с ценами и возможность быстрой настройки ценовой политики  (указание допустимых наценок, налогов, рентабельности).  3. Возможность проведения анализа притока и оттока клиентов за месяц, полугодие, год.  4. Легкое разграничение доступа к файлам и папкам с помощью настройки клиента Novell и установки паролей на файлы. |

3. Расчет экономической эффективности проекта

3.1. Выбор методики расчета экономической эффективности

Внедрение информационных технологий и автоматизация технологических процессов очень тесно связаны с капитальными вложениями, как на покупку техники, так и на формирование плана развития предприятия, выполнения подготовительных работ и обучение кадров. Поэтому процессу внедрения должен предшествовать этап экономического обоснования целесообразности внедрения автоматизированной информационной системы. Таким образом, основополагающих факторов при внедрении должен быть фактор оценки эффективности применения автоматизированных информационных технологий в конкретном производстве.

Под эффективностью автоматизированной обработки экономической информации понимают целесообразность использования средств компьютерной и организационной техники, специальных программных продуктов, развертывание сети и сетевых технологий при формировании, передаче и обработке данных.

Принято различать расчетную и фактическую эффективность. Расчетную эффективность определяют на этапе проектирования автоматизации информационных работ, то есть разработки рабочего проекта, фактическую эффективность рассчитывают по результатам внедрения рабочего проекта. Существует следующая зависимость, что, чем больше процессов в управленческих работах автоматизировано, тем выше эффективность использования технического и программного обеспечения.

Экономический эффект от внедрения вычислительной и организационной техники подразделяют на прямой и косвенный.

Под **прямой экономической эффективностью** понимают количественную характеристику в виде экономии материально-трудовых ресурсов и денежных средств, которая получена в результате сокращения численности управляющих структура и менеджмента, общего фонда заработной платы, расходов на основные и вспомогательные материалы вследствие автоматизации конкретных видов планово-учетных и аналитических работ, которые выполняются на предприятии.

Не исключено, что внедрение автоматизированных информационных технологий на первом этапе не приведет к уменьшению числа работников планово-учетных служб. В этом случае учитывают косвенную эффективность, проявляющуюся в конечных результатах хозяйственной деятельности предприятия.

**Косвенный экономический эффект** может быть охарактеризован показателями увеличения прибыли, увеличением общего или относительного числа клиентов, снижением уровня брака при выполнении печатных работ, снижением затрат на пленку и бумагу, расходные материалы, уменьшением простоев, снижением количества отказов и т.д.

Локальными критериями косвенной эффективности может быть:

* + сжатие сроков составления сводок,
  + повышение качества работ по планированию и аналитических работ,
  + сокращение документооборота и экономия на расходах на организацию документооборота,
  + повышение общей культуры труда всех сотрудников,
  + повышение производительности труда и др.
  + оперативная связь с клиентами.

Основным же показателем является повышение качества управления, которое, как и при прямой экономической эффективности, ведет к экономии живого и овеществленного труда. Оба вида рассмотренной экономической эффективности взаимоувязаны.

Определяют экономическую эффективность с помощью трудовых и стоимостных показателей. Основным при расчетах является метод сопоставления данных базисного и отчетного периодов. В качестве базисного периода при переводе отдельных работ на автоматизацию принимают затраты на обработку информации до внедрения автоматизированной информационной технологии (при ручной обработке), а при совершенствовании действующей системы автоматизации экономических работ – затраты на обработку информации при достигнутом уровне автоматизации. При этом пользуются абсолютными и относительными показателями.

Прямой экономический эффект характеризуется снижением трудовых, стоимостных показателей.

К **трудовым показателям** относятся следующие:

1) абсолютное снижение трудовых затрат (ΔТ):

ΔТ = Т0- Т1,

гдеТ0- трудовые затраты на обработку информации по базовому варианту;

Т1- трудовые затраты на обработку информации по предлагаемому варианту;

2) коэффициент относительного снижения трудовых затрат (КТ):

КТ= ΔТ / T0\* 100%;

3) индекс снижения трудовых затрат или повышение производительности труда (YT):

YT= T0/ T1.

К **стоимостным показателям** относятся: абсолютное снижение стоимостных затрат (ΔC), коэффициент относительного снижения стоимостных затрат (КC) индекс снижения стоимостных затрат (YC), рассчитываемые аналогично.

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта машинной обработки информации (Ток):

Ток= КП/ΔC ,

гдеКП- затраты на создание проекта машинной обработки информации (проектирование и внедрение).

Для оценки экономической эффективности внедрения проекта АРМ в ООО «Неопринт» будем использовать методику с применением построения таблиц сравнения временных показателей до и после внедрения АРМ на предприятии. Построение данной таблицы позволит оценить прямой эффект, который заключается в снижении трудозатрат на выполнение расчета заказов и формирование документации.

3.2. Расчет затрат на разработку программного продукта

Рассчитаем основные затраты на разработку и внедрение автоматизированной информационной системы на всех этапах разработки и проектирования, для обоснования ее экономической эффективности. Плановые затраты на разработку включают все расходы, связанные с ее выполнением. Для того, чтобы узнать размер издержек разработки АИС и ее внедрения, необходимо составить калькуляцию ее плановой себестоимости, на основании которой и производится планирование и учет затрат на выполнение.

Смета затрат содержит статьи:

- основная заработная плата разработчиков АИС;

- дополнительная заработная плата разработчиков АИС;

* отчисления на социальные страхования;
* затраты на амортизацию ПК;
* расходы на электроэнергию, затрачиваемую при разработке АИС;

- накладные расходы.

Рассчитаем подробно каждую из перечисленных статей.

***Расчет затрат на основную заработную плату разработчикам АИС.***

Заработная плата – это средства, которые выплачиваются работникам в денежной и натуральной формах за отработанное время, сделанную работу, а также за неотработанное время в установленном законодательством порядке. Основная заработная плата выплачивается исходя из того, какие формы оплаты труда применяются в каждой конкретной организации. Повременная форма оплаты предполагает начисление за отработанное работником время, а сдельная за сделанную им работу.

Для расчета зарплаты специалистов, служащих, рабочих, руководителей применяется повременная форма оплаты труда – зарплата начисляется согласно должностному окладу работника в месяц за отработанное время. Также учитываются доплаты, связанные со сверхурочной работой, и которые рассчитываются в размере 20% тарифной ставки работника.

Рассчитаем затраты на основную зарплату (Зосн.) при повременной форме оплаты труда, применив формулу (3.1):

Зосн. = Омес.\*Траб.\*Кд/Др.мес., (3.1)

где:

Омес. - месячный оклад разработчика программы;

Др.мес. - среднее количество рабочих дней в месяце;

Траб. - фактическое время участия в разработке программы;

Кд - коэффициент, учитывающий доплаты к основной зарплате.

При этом отношение Омес./Др.мес. характеризует среднюю дневную зарплату разработчика.

Пусть для нашей разработки будут произведены следующие затраты:

Омес. руководителя разработки компьютерного приложения = 8000 руб.

Омес. инженера - программиста = 7400 руб.

Омес. оператора ПК = 6000 руб.

Др.мес. = 21 день;

Кд=1,2.

Расчет затрат на основную зарплату разработчиков компьютерного приложения показан в таблице 3.1.

Таблица 3.1.Расчет затрат на основную заработную плату разработчиков компьютерного приложения

|  |  |  |  |
| --- | --- | --- | --- |
| Исполнители | Время работы, кол  дней | Средняя дневная зарплата Омес./Др.мес, руб. | Затраты на зарплату, руб. |
| Руководитель | 6 | 381,0 | 2 286 |
| Инженер-программист | 6 | 352,4 | 2 114,4 |
| Оператор ПК | 6 | 285,7 | 1 714,2 |
| Итого |  |  | 6 114,6 |

***Расчет дополнительной заработной платы разработчиков АИС***

Статья «Дополнительная заработная плата» предполагает планирование и учет выплат, которые предусмотрены в законодательстве о труде или в коллективном договоре за неявочное (непроработанное на производстве) время, например, оплата отпусков или компенсация за неиспользованный отпуск, оплата времени, связанного с выполнением государственных и общественных обязанностей и др. Она рассчитывается в процентном отношении от основной зарплаты.

Здоп. = Кдоп. \* Зосн. (3.2)

где Кдоп. – коэффициент величины дополнительной зарплаты разработчиков.

Примем Кдоп. равным 0,25. По формуле (3.2) получим:

Здоп. = 0,25 \* 6114,6 = 1528,65 руб.

***Расчет отчислений на социальное страхование и обеспечение.***

Исходя из законов РФ о медицинском страховании, о пенсионном обеспечении, о государственном социальном страховании, о занятости населения, работники предприятий подлежат обязательному социальному страхованию и обеспечению. Отчисления на социальное страхование содержат (в % к сумме основной и дополнительной заработной платы):

социальное страхование 3,2

медицинское страхование 2,8

пенсионный фонд 20,0

фонд занятости 10

налог на содержание объектов образования 1,5

транспортный налог (на балансе машина и физические лица - владельцы авто) 1,0 %

ИТОГО: 38,5 %

Так, отчисления на социальное страхование и обеспечение, входящие в состав затрат на производство можно рассчитать по формуле:

Ос.с.о. = Кс.с.о. \* (Зосн. + Здоп.) (3.3)

где Кс.с.о. - коэффициент, учитывающий отчисления в фонд социального страхования, пенсионный фонд, медицинского страхования, государственный фонд занятости. Таким образом, находим:

Ос.с.о. = 0,405\*(6114,6 + 1528,65) = 3095,5 руб.

***Расчет затрат на амортизацию ПК***

Также при составлении сметы расходов на разработку компьютерного приложения необходимо учитывать размер амортизации ПК, которые используются разработчиками.

Амортизация представляет процесс постепенного изнашивания основных средств ипереноса их стоимости на произведенный продукт по установленным нормам. При начислении амортизационных отчислений следует руководствоваться ПБУ 6/01 «Учет основных средств». Нормы амортизационных отчислений установлены в процентах к балансовой (первоначальной) стоимости основных средств. Амортизация начисляется ежемесячно. Норма амортизации рассчитывается по формуле при условии Спер=100%:

На=Сперв/срок полезного использования (%)

Расчет затрат на амортизацию оборудования производится следующим образом:

Зам.=Сперв.\*(На/100) \* m \* (tраб/Фд.о.) (3.4)

где Сперв.- первоначальная стоимость ПК, используемой при разработке программы;

На - норма амортизационных отчислений;

m - количество используемых ПК;

tpaб. - время работы ПК;

Фд.о. - действительный годовой фонд времени работы ПК.

Пусть Сперв. = 125000 руб., На =12,5%, m = 3 шт., tpaб. = 6 дней \* 8 ч. = 48 ч,

Фд.о. = Кол.раб.дн. \* Кол.смен \* Продолж.смены = 252 дня\* 1 смена\* 8 ч. = 2016 ч.

На основе формулы (3.4) определяем:

Зам.= 125000 \*(12,5/100)\*3\*(48/2016) = 1116,07 руб.

Результаты расчета затрат на амортизацию ПК, используемые при разработке программы, представлены в таблице 3.2.

Таблица 3.2*.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование  оборудования | Количество  единиц  оборудования т, шт | Время  работы  оборудования, tpaб., ч | Норма  амортизационных  отчислении, **%** | Затраты на  амортизацию, руб. |
| IBM PC  Pentium 166  Pro | 3 | 48 | 12,5 | 1116,07 |

***Расчет затрат на электроэнергию, используемую ПК в процессе разработки компьютерного приложения.***

Затраты на электроэнергию (Зэл.эн.) рассчитываются по формуле:

Зэл.эн.=Цэ. \* Р \* m \* tp (3.5)

где:

Р - мощность ПК, используемого при разработке программы;

tp - время работы ПК, используемого при разработке программы;

m - количество используемых ПК;

Цэ. - цена 1 кВт\*ч электроэнергии.

Пусть Р = 250 Вт; tp = 48 ч; m = 3; Цэ. = 4 руб.

На основе формулы (3.5) определяем Зэл.эн.:

Зэл.эн. = 3\*150\*3\*48 = 64800 руб.

Результаты расчета затрат на электроэнергию, используемую в процессе разработки программы, представлены в таблице 3.3.

Таблица 3.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование  оборудования | Количество единиц оборудования т, шт | Время работы  оборудования tp.,ч | Мощность оборудования, кВт | Затраты на электроэнергию,  руб. |
| IBM PC  Pentium 166  Pro | 3 | 48 | 150 | 64800 |

***Расчет накладных расходов***

Встатью "Накладные расходы" включаются расходы на управление и хозяйственное обслуживание. По этой статье учитывается заработная плата аппарата управления и общехозяйственных служб, затраты на содержание и текущий ремонт зданий, сооружений, оборудования и инвентаря, амортизационные отчисления на их полное восстановление и капитальный ремонт, расходы по охране труда, научно-технической информации, изобретательству и рационализации. Величина накладных расходов определяется в процентах от основной и дополнительной заработной платы.

Накладные расходы (Рнакл.) рассчитываются по формуле:

Рнакл =Кн \* (Зосн.+Здоп.) (3.6)

где:

Кн - коэффициент накладных расходов.

Примем Кн равным 1,1. На основе формулы (3.6) определяем:

Рнакл. = 1,1\* (6114,6 + 1528,65) = 8407,58 руб.

Результаты расчета затрат на разработку компьютерного приложения сведем в таблицу 3.4.

Таблица 3.4. Смета затрат на разработку компьютерного приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Статьи затрат | Затраты, руб. | % к итогу |
| 1 | Основная заработная плата разработчиков | 6114,6 | 7,19 |
| 2 | Дополнительная заработная плата разработчиков | 1528,65 | 1,8 |
| 3 | Отчисления на социальное страхование. | 3095,5 | 3,64 |
| 4 | Амортизационные отчисления | 1116,07 | 1,31 |
| 5 | Расходы на электроэнергию | 64800 | 76,18 |
| 6 | Накладные расходы | 8407,58 | 9,9 |
|  | Итого: | 85062,4 | 100% |
|  | Затраты на разработку Сполн. | 85062,4 |  |

3.3. Оценка экономической эффективности внедрения автоматизации дипломной документации

Экономическая эффективность от автоматизации рабочего места менеджера по работе с клиентами ООО «Неопринт» будет иметь как прямой, так и косвенный эффект.

Прямой эффект.

Экономическую эффективность от внедрения данного проекта будем рассчитывать с помощью трудовых и стоимостных показателей прямого эффекта, таких как: абсолютное снижение трудовых затрат (Т), коэффициент относительного снижения трудовых затрат (*КТ*), индекс снижения трудовых затрат или повышение производительности труда (*YT*); абсолютное снижение стоимостных затрат (C), коэффициент относительного снижения стоимостных затрат (*КC),* индекс снижения стоимостных затрат (*YC*). Данные по расчетам вышеуказанных показателей представлены в таблице 3.5.

Таблица 3.5. Показатели эффективности от внедрения системы автоматизации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Затраты | | Абсолютное изменение затрат | Коэффициент изменения затрат | Индекс изменения затрат |
| Базовый вариант | Проектный вариант |
| Трудоемкость | T0 (час) | T1 (час) | Т=Т0–Т1 (час) | КТ=Т/T0100% | YT=T0/T1 |
|  | 30 | 5 | 25 | 83,3% | 6 |
| Стоимость | C0 (руб.) | C1 (руб.) | C=C0–C1 (руб.) | КC=C/C0100% | YC=C0/C1 |
|  | 6000 | 1000 | 5000 | 83,3% | 6 |

Внедрение данного компьютерного приложения в работе менеджера предполагает автоматизацию его рабочего места. До внедрения компьютерного приложения менеджер тратил на обработку заказа, заполнение бланков и базы данных, расчет стоимости заказа порядка четырех рабочих дней (это около 30 рабочих часов - в таблице 3.5 – Т0).

Теперь же, после внедрения данного приложения менеджеру потребуется для выполнения той же работы порядка пяти часов (в таблице 3.5 – Т1). На основе этих данных рассчитываются: абсолютное изменение трудовых затрат (Т), коэффициент изменения трудовых затрат (КТ), индекс изменения трудовых затрат (YT).

Оплата труда менеджера до внедрения приложения составляла 1500 рублей в день, то есть это порядка 6000 рублей (C0), а оплата труда после внедрения проекта составит около 1000 рублей за всю выполненную работу (C1).

Сравнив получившиеся показатели *YT*(индекс изменения трудовых затрат) *и YC*(индекс изменения стоимостных затрат), можно сделать следующий вывод: трудоемкость выполнения работы уменьшиться примерно в шесть раз, и соответственно, стоимостные затраты на оплату труда уменьшатся в шесть раз. Следовательно, внедрение данного проекта принесет положительный экономический эффект для всего предприятия.

К тому же высвобождение времени дает возможность менеджеру больше времени уделять творческой составляющей в работе, поиску и привлечению новых клиентов. Тогда экономический эффект для типографии будет измеряться стоимостью высвободившегося дополнительного времени менеджера, которое он теперь будет тратить на другие виды работ (т.е. составит С = 5000 руб.).

**Косвенный эффект.**

При внедрении автоматизации в работе менеджера косвенный экономический эффект получается за счет:

1. Сокращения документооборота и доступности информации – вся необходимая информация и необходимые отчеты будут храниться в базе данных компьютера и в любой момент будут доступны для сотрудников типографии, а также могут быть выданы при обращении клиентов.

2. Наглядности, унификации и быстроте получения информации – заказ будет оформляться по единому образцу, будет производиться быстрый расчет его стоимости, клиенту будет предоставлена возможность получения оперативной информации относительно стоимости его заказа, сопоставления стоимости различных вариантов заказа для выбора оптимального.

3. Повышения качества организационной и аналитической работы для менеджера, у которого высвободится время для непосредственной работы с клиентом над вариантами заказа, что дает возможность к повышению качества и результативности работы.

4. Снижение риска допуска ошибок, связанных с человеческим фактором.

5. Ускорение процесса обмена информацией между структурными подразделениями типографии, что повышает эффективность ее работы в целом.

3.4. Оценка периода окупаемости проекта

Срок окупаемости (величина, обратная коэффициенту эффективности) – показатель эффективности использования капиталовложений – представляет собой период времени, в течение которого произведенные затраты на программные изделия окупаются полученным эффектом.

*Ток**= КП /*C*,*

где КП - затраты на создание проекта машинной обработки информации (проектирование и внедрение).

Затратами на создание проекта машинной обработки информации (проектирование и внедрение) как правило выступает заработная плата программиста (не обязательно это может быть программист в штате сотрудников) и дополнительная оплата сотрудникам, формирующим техническое задание и тестирующим программу. Немаловажным фактором является стоимость сопровождения программного обеспечения. Она заведомо ниже, чем заработная плата программиста (а тем более группы программистов) в штате.

Затраты на разработку и внедрение компьютерного приложения для автоматизации дипломной документации были рассчитаны в п. 3.2 и составили 85062,4 руб.

Таким образом, срок окупаемости проекта составит

85062,4/5000 = 17 дней.

Заключение

Использование новых информационных технологий на предприятиях на современном этапе становится жизненно важным для сохранения качества обслуживания и клиентов в условиях жесткой рыночной конкуренции.

Одним из способов повышения эффективности работы всего предприятия в целом является создание автоматического рабочего места менеджера по работе с клиентами. Что дает не только прямую экономию трудозатрат, но и косвенный эффект в виде повышения прибыли и увеличения числа клиентов.

Все это говорит о том, что тематика дипломной работы является чрезвычайно актуально для малого и среднего бизнеса.

Дипломная работа посвящена разработке информационной системы для организации работы с клиентами и расчета заказов полиграфического предприятия ООО «Неопринт».

Сложность разработки информационной системы для бизнеса заключается в учете специфических требования предприятия. В данном случае это учет особенностей расчета заказов и делопроизводства в сфере предоставления широкого спектра полиграфических услуг.

Кроме того, необходимо учесть выполнение таких требований времени как мобильность, скорость расчета, эргономика интерфейса, совместимость, дешевизна внедрения.

Для решения всех поставленных задач по созданию АРМ был выбран язык программирования Delphi как наиболее подходящий для реализации с учетом всех факторов.

Использование Delphi позволило минимизировать затраты на разработку и внедрение информационной системы на ООО «Неопринт», а также достичь высоких показателей эффективности. Низкие затраты достигают за счет использования только стандартных офисных приложений, которые имеют встроенную систему программирования.

Внедрение проекта АРМ заключается в развертывании локальной сети Novell, установке клиентов, разграничении доступа к файлам и копировании файлов на сетевое хранилище. Таким образом, наибольшие показатели эффективности внедрения могут быть достигнуты, если предприятие уже имеет локальную сеть и беспроводной доступ. В этом случае затраты состоят только из затрат на разработку.

В работе также детально были рассмотрены теоретические и технические аспекты внедрения АРМ, проведен эксперимент по оценке трудозатрат. В последней главе изучены условия труда менеджера, а также представлены рекомендации по улучшению условий труда менеджера по работе с клиентами в сфере полиграфии.

Список использованной литературы

1. Авдошин С. М., Песоцкая Е. Ю. Информатизация бизнеса. Управление риска-ми. – М.: ДМК Пресс, 2011. – 356 с.
2. Автоматизированные информационные технологии в экономике: Учебник /Под ред. проф. Г. А. Титоренко. - М.: Компьютер, ЮНИТИ, 2003. – 247 с.
3. Антонова Г. М., Байков А. Ю. Современные средства ЭВМ и телекоммуникаций: учеб. пособие для вузов. - М.: Академия, 2010. - 144 с.
4. Безручко В. Т. Компьютерный практикум по курсу "Информатика": работа в Windows XP, Word 2003, Excel 2003, PowerPoint 2003, Outlook 2003, PROMT Family 7.0, Интернет: учеб. пособие для вузов.- 3-е изд., перераб. и доп. - М. : Форум, 2009. - 368 с.
5. Венделева М. А., Вертакова Ю. В. Информационные технологии управления: учебное пособие. - Москва: Юрайт, 2011. – 462 с.
6. Вендров А. М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2008. – 238 с.
7. Гапоненко А. Л., Панкрухин А. П. Стратегическое управление. – СПб.: Омега-Л. Серия: Высшая школа менеджмента, 2010. – 502 с.
8. Граничин О. Н., Кияев В. И. Информационные технологии в управлении. – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний. Серия: Основы информационных технологий. - 2011.
9. Заложнев А. Ю., Шуремов Е. Л. Информационные технологии маркетинга. Управление взаимоотношениями с клиентами. – М.: Бухгалтерия и Банки, 2009. – 361 с.
10. Ивасенко А. Г., Гридасов А. Ю., Павленко В. А. Информационные технологии в экономике и управлении. – М.: КноРус, 2010. – 289с.
11. Информатика: базовый курс: учеб. пособие для студентов втузов / под ред. С. В. Симоновича.- 2-е изд. - СПб. : Питер, 2010. - 640 с.
12. Информатика: учеб. для вузов / под ред. Н. В. Макаровой.- 3-е изд., перераб. - М.: Финансы и статистика, 2009. - 768 с.
13. Информатика в экономике: учеб. пособие / под ред. Б. Е. Одинцова, А. Н. Романова. - М.: Вузовский учебник, 2010. - 478 с.
14. Информатика и информационные технологии: учеб. пособие / под ред. Ю. Д. Романовой.- 4-е изд., перераб. и доп. - М.: Эксмо, 2010. - 688 с.
15. Информатика и программирование. Компьютерный практикум: учеб. пособие / А. Н. Гуда и др. - М.: Дашков и К, 2009. - 238 с.
16. Информационные системы и технологии в экономике и управлении / Под ред. В. Трофимова. – М.: Юрайт. Серия: Бакалавр, 2011. – 462 с.
17. Информационные системы и технологии управления / Под ред. Г. А. Титоренко. – М.: Юнити-Дана. Серия: Золотой фонд российских учебников, 2010. – 392 с.
18. Кандрашина Е. А. Финансовый менеджмент.–М.:Дашков и Ко,2012.–340с.
19. Каймин В. А. Информатика: учебник. - М.: Проспект, 2009. - 272 с.
20. Косарев В. П., Мамонтова Е. А. Информатика: практикум для экономистов: учеб. пособие. - Финансовая акад. при Правительстве Рос. Федерации. - М.: Финансы и статистика: ИНФРА-М, 2009. - 544 с.
21. Логинов В. Н. Информационные технологии управления: учебное пособие по специальности «Государственное и муниципальное управление». - Москва: КноРус, 2008. – 238 с.
22. Мельников В. П., Клейменов С. А., Петраков А. М. Информационная безопасность и защита информации: учеб. пособие для вузов.- 4-е изд., стер. - М. : Академия, 2009. - 332 с.
23. Мишенин А. И. Теория экономических информационных систем. – М.: Финансы и статистика, 2008. – 292 с.
24. Саак А. Э., Пахомов Е. В. Информационные технологии управления: учебник по специальности «Государственное и муниципальное управление». - Санкт-Петербург: Питер, 2009. - 320 с.
25. Смирнова Г. Н., Сорокин А. А., Тельнов Ю. Ф. Проектирование экономических информационных систем. Учебник. – М.: Финансы и статистика, 2001. – 607 с.
26. Управление персоналом / Под ред. А. Литвинюка. – М.: Юрайт. Серия: Бакалавр, 2012. – 198 с.
27. Фатхутдинов Р. А. Производственный менеджмент. – СПб.: Питер. Серия: Учебник для вузов, 2011. – 348 с.
28. Шуремов Е. Л. Информационные технологии оптовой торговли [Электронный ресурс] / Режим доступа: www.booksgid.com/business/583-.html

Приложение. Код программы

unit Unit1;

interface

uses

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

Dialogs, StdCtrls, Grids, DBGrids, Menus, unit2, DBCtrls, ExtCtrls, comobj;

type

TMyDBGrid = class(TDBGrid);

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

DBGrid1: TDBGrid;

Label1: TLabel;

Button10: TButton;

Button11: TButton;

Button12: TButton;

Button9: TButton;

DBNavigator1: TDBNavigator;

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox2: TDBLookupComboBox;

Button13: TButton;

Button14: TButton;

Button15: TButton;

Button16: TButton;

Label2: TLabel;

Label3: TLabel;

procedure Button6Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure Button13Click(Sender: TObject);

procedure Button14Click(Sender: TObject);

procedure Button15Click(Sender: TObject);

procedure Button16Click(Sender: TObject);

procedure Button12Click(Sender: TObject);

procedure Button11Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

korzina:double;

implementation

uses Unit3, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9, Unit10, Unit11, Unit12;

{$R \*.dfm}

procedure TForm1.Button10Click(Sender: TObject);

begin

form11.show;

end;

procedure TForm1.Button11Click(Sender: TObject);

var

a,b: Integer;

WordApp, Wd: Variant;

appath1:string;

j,k, t:integer;

sss:string;

begin

WordApp:=CreateOleObject('Word.Application');

appath1:=ExtractFilePath(application.exename ) ;

Wd:=WordApp.Documents.add(appath1+'\tempword.dotx');

k:=dbgrid1.DataSource.DataSet.RecordCount;

with TMyDBGrid(DBGrid1).DataLink do

begin

t:=activerecord;

// dbgrid1.DataSource.DataSet.MoveBy(t);

WordApp.ActiveDocument.Range.InsertAfter(Fields[5].AsString);

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('Вид продукции: '+Fields[1].AsString);

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('Количество: '+Fields[2].AsString);

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('Цена:');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

if (Fields[3].AsString<>'') then WordApp.ActiveDocument.Range.InsertAfter(strtofloat(Fields[3].AsString)) else WordApp.ActiveDocument.Range.InsertAfter('');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('ЦенаНДС:');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

if (Fields[4].AsString<>'') then WordApp.ActiveDocument.Range.InsertAfter(strtofloat(Fields[4].AsString)) else WordApp.ActiveDocument.Range.InsertAfter('');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('Вид доставки:');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter(Fields[8].AsString);

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter('Дата заказа:');

WordApp.ActiveDocument.Range.InsertAfter(chr(13)+chr(10));

WordApp.ActiveDocument.Range.InsertAfter(Fields[14].AsString);

end;

WordApp.Visible:=true;

end;

procedure TForm1.Button12Click(Sender: TObject);

var

a,b: Integer;

ExApp, WB, WS: Variant;

appath1:string;

uu:array[1..10] of integer;

j,k, t:integer;

sss:string;

begin

ExApp:=CreateOleObject('Excel.Application');

appath1:=ExtractFilePath(application.exename ) ;

WB:=ExApp.WorkBooks.open(appath1+'\temp.xltx');

WS := ExApp.Workbooks[1].WorkSheets[1];

k:=dbgrid1.DataSource.DataSet.RecordCount;

with TMyDBGrid(DBGrid1).DataLink do

begin

for t := 1 to k do

begin

for j := 0 to 14 do

begin

if (j=3) or (j=4) then

begin

if Fields[j].AsString<>'' then

WS.Cells[t+1, j+1].value:=strtofloat(Fields[j].AsString)

else WS.Cells[t+1, j+1].value:=''

end

else

WS.Cells[t+1, j+1].value:=Fields[j].Asstring;

end;

dbgrid1.DataSource.DataSet.MoveBy(1);

end;

end;

ExApp.Visible:=true;

end;

procedure TForm1.Button13Click(Sender: TObject);

var s:string;

begin

s:=form1.DBLookupComboBox1.KeyValue;

datamodule2.korzinatable.Filtered:=True;

datamodule2.korzinatable.Filter:='статусоплат='+s;

end;

procedure TForm1.Button14Click(Sender: TObject);

var s:string;

begin

s:=form1.DBLookupComboBox2.KeyValue;

datamodule2.korzinatable.Filtered:=True;

datamodule2.korzinatable.Filter:='клиент='+s;

end;

procedure TForm1.Button15Click(Sender: TObject);

begin

datamodule2.korzinatable.Filtered:=false;

end;

procedure TForm1.Button16Click(Sender: TObject);

begin

korzina:=0;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

form7.show;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

form8.show;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

form10.show;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

form6.show;

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

form4.show;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

form3.show;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

form5.show;

end;

procedure TForm1.Button8Click(Sender: TObject);

begin

form9.show;

end;

procedure TForm1.Button9Click(Sender: TObject);

begin

form12.show;

end;

end.

unit Unit2;

interface

uses

SysUtils, Classes, DB, ADODB;

type

TDataModule2 = class(TDataModule)

ADOConnection1: TADOConnection;

sotrtable: TADOTable;

klienttable: TADOTable;

dostavkatable: TADOTable;

formatchekatable: TADOTable;

vidtisntable: TADOTable;

svetntable: TADOTable;

plotntable: TADOTable;

formattable: TADOTable;

vidbumtable: TADOTable;

vidlaktable: TADOTable;

formatplenkitable: TADOTable;

pararmstable: TADOTable;

oplattable: TADOTable;

vidprodtable: TADOTable;

korzinatable: TADOTable;

sotr: TDataSource;

sotrtableКод: TAutoIncField;

sotrtableфио: TWideStringField;

sotrtableтелефон: TIntegerField;

klient: TDataSource;

klienttableКод: TAutoIncField;

klienttableфио: TWideStringField;

klienttableтелефон: TWideStringField;

klienttableрс: TWideStringField;

dostavka: TDataSource;

formatcheka: TDataSource;

vidtisn: TDataSource;

svetn: TDataSource;

plotn: TDataSource;

format: TDataSource;

vidbum: TDataSource;

vidlak: TDataSource;

formatplenki: TDataSource;

params: TDataSource;

oplat: TDataSource;

vidprod: TDataSource;

korzina: TDataSource;

dostavkatableКод: TAutoIncField;

dostavkatableназвание: TWideStringField;

formatchekatableКод: TAutoIncField;

formatchekatableформатчека: TWideStringField;

formatchekatableценатермоэко: TWideStringField;

formatchekatableценатермотоп: TWideStringField;

vidtisntableКод: TAutoIncField;

vidtisntableвидтиснения: TWideStringField;

vidtisntableценаформатА4: TWideStringField;

vidtisntableценаформатА3: TWideStringField;

svetntableКод: TAutoIncField;

svetntableчислоцветов: TWideStringField;

svetntableкоэфцены: TWideStringField;

plotntableКод: TAutoIncField;

plotntableплотность: TWideStringField;

formattableКод: TAutoIncField;

formattableформат: TWideStringField;

vidlaktableКод: TAutoIncField;

vidlaktableвидлака: TWideStringField;

vidlaktableцена: TWideStringField;

formatplenkitableКод: TAutoIncField;

formatplenkitableтолщина: TWideStringField;

formatplenkitableцена: TWideStringField;

pararmstableКод: TAutoIncField;

pararmstableназваниепараметра: TWideStringField;

pararmstableзначение: TWideStringField;

dostavkatableцена: TWideStringField;

vidprodtableКод: TAutoIncField;

vidprodtableнаименование: TWideStringField;

korzinatableКод: TAutoIncField;

korzinatableколичество: TWideStringField;

korzinatableцена: TWideStringField;

korzinatableценасНДС: TWideStringField;

korzinatableвидбумаги: TIntegerField;

korzinatableклиент: TIntegerField;

korzinatableменеджер: TIntegerField;

korzinatableдоставка: TIntegerField;

korzinatableстатусоплат: TIntegerField;

korzinatableвидлак: TIntegerField;

korzinatableвидтисн: TIntegerField;

korzinatableцветность: TIntegerField;

korzinatableформатчека: TIntegerField;

korzinatableформатпленки: TIntegerField;

korzinatableдатазаказа: TDateTimeField;

korzinatablevidprod: TStringField;

korzinatableвидпродукции: TIntegerField;

korzinatableklient: TStringField;

korzinatablemeneger: TStringField;

oplattableКод: TAutoIncField;

oplattableстатус: TWideStringField;

korzinatableoplata: TStringField;

korzinatabledost: TStringField;

vidbumtableКод: TAutoIncField;

vidbumtableвид: TWideStringField;

vidbumtableплотность: TIntegerField;

vidbumtableформат: TIntegerField;

vidbumtableцена: TWideStringField;

private

{ Private declarations }

public

{ Public declarations }

end;

var

DataModule2: TDataModule2;

implementation

{$R \*.dfm}

end.

unit Unit3;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, unit2, unit1;

type

TForm3 = class(TForm)

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

CheckBox4: TCheckBox;

CheckBox6: TCheckBox;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label4: TLabel;

Label5: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Label6: TLabel;

Edit4: TEdit;

Label7: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox3: TDBLookupComboBox;

Label10: TLabel;

DBLookupComboBox4: TDBLookupComboBox;

DBText1: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

var

fs,otris,hud,ver,prob:double;

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr := 200;

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

If (checkbox1.Checked) Then fs:=400 Else fs:=0;

If (checkbox2.Checked) Then otris:=600 Else otris:=0;

If (checkbox3.Checked) Then hud:=800 Else hud:=0;

If (checkbox4.Checked) Then ver:=1000 Else ver:=0;

If (checkbox6.Checked) Then prob:=200 Else prob:=0;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr+kol\*(fs+otris+hud+ver+prob);

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext1.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label5.Caption :=floattostr(cena);

label4.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

if (label4.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

form3.Close;

end;

end.

unit Unit4;

interface

uses

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

Dialogs, DBCtrls, StdCtrls, ExtCtrls, unit2, unit1;

type

TForm4 = class(TForm)

RadioGroup1: TRadioGroup;

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

Edit1: TEdit;

Label1: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label4: TLabel;

Label5: TLabel;

Edit4: TEdit;

Label7: TLabel;

Label2: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Label6: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox3: TDBLookupComboBox;

DBLookupComboBox4: TDBLookupComboBox;

DBText1: TDBText;

DBText2: TDBText;

Label10: TLabel;

DBLookupComboBox5: TDBLookupComboBox;

DBText3: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm4.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

If (edit1.text <> '') Then kol := strtoint(edit1.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

if form4.RadioButton1.Checked then cenametr:=strtofloat(dbtext2.Field.Value);

if form4.RadioButton2.Checked then cenametr:=strtofloat(dbtext1.Field.Value);

cena:=kol\*cenametr;

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext3.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label5.Caption :=floattostr(cena);

label4.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm4.Button2Click(Sender: TObject);

begin

if (label4.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox5.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('форматчека').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm4.Button3Click(Sender: TObject);

begin

form4.Close;

end;

end.

unit Unit5;

interface

uses

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

Dialogs, DBCtrls, StdCtrls, unit2, unit1;

type

TForm5 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox3: TDBLookupComboBox;

Label4: TLabel;

Label6: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label10: TLabel;

Label11: TLabel;

Edit4: TEdit;

Label12: TLabel;

Label7: TLabel;

DBLookupComboBox4: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox5: TDBLookupComboBox;

DBLookupComboBox6: TDBLookupComboBox;

DBText1: TDBText;

DBText2: TDBText;

Label5: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

DBText3: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm5.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr := 40;

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

if (dblookupcombobox1.KeyValue=2) or (dblookupcombobox1.keyValue=1) then cenametr:=dbtext1.Field.Value;

if dblookupcombobox1.keyValue=3 then cenametr:=dbtext2.Field.Value;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr;

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext3.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label11.Caption :=floattostr(cena);

label10.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm5.Button2Click(Sender: TObject);

begin

if (label10.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)and (DBLookupComboBox5.KeyValue<>Null)and (DBLookupComboBox6.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox6.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox5.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('видтисн').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('видбумаги').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm5.Button3Click(Sender: TObject);

begin

form5.Close;

end;

end.

unit Unit6;

interface

uses

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

Dialogs, DBCtrls, StdCtrls, unit2, unit1;

type

TForm6 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Edit3: TEdit;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

CheckBox4: TCheckBox;

DBLookupComboBox2: TDBLookupComboBox;

DBLookupComboBox4: TDBLookupComboBox;

Label5: TLabel;

Label7: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label11: TLabel;

Label12: TLabel;

Edit4: TEdit;

Label13: TLabel;

Label8: TLabel;

DBLookupComboBox5: TDBLookupComboBox;

Label9: TLabel;

Label10: TLabel;

DBLookupComboBox6: TDBLookupComboBox;

DBLookupComboBox7: TDBLookupComboBox;

DBText1: TDBText;

DBText2: TDBText;

DBText3: TDBText;

Label4: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm6.Button1Click(Sender: TObject);

var svetn, num, vis, porez, scrugl:double;

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr := strtofloat(dbtext1.Field.Value);

svetn:= strtofloat(dbtext3.Field.Value);

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr\*svetn;

if checkbox1.Checked then scrugl:=0.05\*kol else scrugl:=0;

if checkbox2.Checked then porez:=0.02\*kol else porez:=0;

if checkbox3.Checked then vis:=0.03\*kol else vis:=0;

if checkbox4.Checked then num:=0.02\*kol else num:=0;

cena:=cena+scrugl+porez+vis+num;

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext2.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label12.Caption :=floattostr(cena);

label11.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm6.Button2Click(Sender: TObject);

begin

if (label11.Caption<>'0') and (DBLookupComboBox7.KeyValue<>Null) and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox6.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)and (DBLookupComboBox5.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox7.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox6.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox5.KeyValue);

datamodule2.korzinatable.FieldByName('цветность').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('видбумаги').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm6.Button3Click(Sender: TObject);

begin

form6.Close;

end;

end.

unit Unit7;

interface

uses

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

Dialogs, DBCtrls, StdCtrls, unit2, unit1;

type

TForm7 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Label4: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label11: TLabel;

Label12: TLabel;

Edit4: TEdit;

Label13: TLabel;

Label6: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox3: TDBLookupComboBox;

DBLookupComboBox4: TDBLookupComboBox;

DBText1: TDBText;

Label10: TLabel;

DBLookupComboBox5: TDBLookupComboBox;

DBText2: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm7.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr :=strtofloat(dbtext2.Field.Value);

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr;

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext2.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label2.Caption :=floattostr(cena);

label1.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm7.Button2Click(Sender: TObject);

begin

if (label1.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null) and (DBLookupComboBox5.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox5.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('видлак').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm7.Button3Click(Sender: TObject);

begin

form7.Close;

end;

end.

unit Unit8;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, unit2, unit1;

type

TForm8 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label10: TLabel;

Label11: TLabel;

Edit5: TEdit;

Label12: TLabel;

Label6: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

DBLookupComboBox3: TDBLookupComboBox;

Label5: TLabel;

DBLookupComboBox4: TDBLookupComboBox;

DBText2: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form8: TForm8;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm8.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit1.text <> '') Then cenametr := strtoint(edit4.text) Else cenametr := 0;

If (edit5.text <> '') Then skidka := strtofloat(edit5.text)\*0.01 Else skidka := 0;

cena := zw \* 0.001 \* kol \* zh \* 0.001 \* cenametr;

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext2.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label5.Caption :=floattostr(cena);

label4.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm8.Button2Click(Sender: TObject);

begin

if (label10.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm8.Button3Click(Sender: TObject);

begin

form8.Close;

end;

end.

unit Unit9;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, unit2, unit1;

type

TForm9 = class(TForm)

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox2: TDBLookupComboBox;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Label3: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label11: TLabel;

Label12: TLabel;

Edit4: TEdit;

Label13: TLabel;

Label6: TLabel;

DBLookupComboBox3: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox4: TDBLookupComboBox;

DBLookupComboBox5: TDBLookupComboBox;

DBText1: TDBText;

Label10: TLabel;

DBLookupComboBox6: TDBLookupComboBox;

CheckBox1: TCheckBox;

DBText2: TDBText;

procedure Button3Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form9: TForm9;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm9.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr := strtofloat(dbtext1.Field.Value);

If (edit1.text <> '') Then kol := strtoint(edit1.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

if (DBLookupComboBox1.KeyValue='A4') then

begin

zw:=210; zh:=297;

end;

if (DBLookupComboBox1.KeyValue='A3') then

begin

zw:=297; zh:=420;

end;

if zw=0 then

begin

zw:=210; zh:=297;

end;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr;

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext2.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label11.Caption :=floattostr(cena);

label12.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm9.Button2Click(Sender: TObject);

begin

if (label11.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null) and (DBLookupComboBox5.KeyValue<>Null)and (DBLookupComboBox6.KeyValue<>Null) then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox6.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox5.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('форматпленки').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm9.Button3Click(Sender: TObject);

begin

form9.Close;

end;

end.

unit Unit10;

interface

uses

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

Dialogs, StdCtrls, DBCtrls, unit2, unit1;

type

TForm10 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Label4: TLabel;

Label5: TLabel;

Edit4: TEdit;

Label7: TLabel;

Label6: TLabel;

DBLookupComboBox1: TDBLookupComboBox;

Label8: TLabel;

Label9: TLabel;

DBLookupComboBox2: TDBLookupComboBox;

DBLookupComboBox3: TDBLookupComboBox;

DBText1: TDBText;

Label10: TLabel;

DBLookupComboBox4: TDBLookupComboBox;

procedure Button3Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form10: TForm10;

rez1,rw, rh, zap: integer;

zw, zh, kol, cv :integer;

cenarez,rentab: double;

formatpl:string;

i:integer;

cena, cenabeznds, cenametr, skidka:double;

nds, ss:double;

implementation

{$R \*.dfm}

procedure TForm10.Button1Click(Sender: TObject);

begin

rez1:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

zap:=strtoint(datamodule2.pararmstable.FieldByName('значение').AsString);

datamodule2.pararmstable.MoveBy(1);

rentab:=strtofloat(datamodule2.pararmstable.FieldByName('значение').AsString)\*0.01;

nds:=0.18;

cenametr := 40;

If (edit1.text <> '') Then zw := strtoint(edit1.text) Else zw := 0;

If (edit2.text <> '') Then zh := strtoint(edit2.text) Else zh := 0;

If (edit3.text <> '') Then kol := strtoint(edit3.text) Else kol := 0;

If (edit4.text <> '') Then skidka := strtofloat(edit4.text)\*0.01 Else skidka := 0;

ss := zw \* 0.001 \* zh \* 0.001 \* kol;

cena := ss \* cenametr;

cena := cena \* (1 + 20 \* 0.01);

cena := cena \* (1 + rentab);

cena := cena+strtofloat(dbtext1.Field.Value);

cenabeznds := cena - cena \* skidka;

cena := cena - cena \* skidka;

cena := cena \* (1 + nds);

label5.Caption :=floattostr(cena);

label4.Caption :=floattostr(cenabeznds);

button2.Enabled:=true;

end;

procedure TForm10.Button2Click(Sender: TObject);

begin

if (label4.Caption<>'0') and (DBLookupComboBox1.KeyValue<>Null) and (DBLookupComboBox2.KeyValue<>Null) and (DBLookupComboBox3.KeyValue<>Null) and (DBLookupComboBox4.KeyValue<>Null)then

begin

datamodule2.korzinatable.Insert;

datamodule2.korzinatable.FieldByName('видпродукции').AsInteger:=strtoint(DBLookupComboBox4.KeyValue);

datamodule2.korzinatable.FieldByName('доставка').AsInteger:=strtoint(DBLookupComboBox3.KeyValue);

datamodule2.korzinatable.FieldByName('клиент').AsInteger:=strtoint(DBLookupComboBox2.KeyValue);

datamodule2.korzinatable.FieldByName('менеджер').AsInteger:=strtoint(DBLookupComboBox1.KeyValue);

datamodule2.korzinatable.FieldByName('количество').AsString:=inttostr(kol);

datamodule2.korzinatable.FieldByName('цена').AsString:=floattostr(cenabeznds);

datamodule2.korzinatable.FieldByName('ценасНДС').AsString:=floattostr(cena);

datamodule2.korzinatable.FieldByName('датазаказа').value:=now();

datamodule2.korzinatable.FieldByName('статусоплат').AsInteger:=1;

korzina:=korzina+cena;

form1.Label3.Caption:=floattostr(korzina);

datamodule2.korzinatable.Post;

end

else showmessage('Заполните все поля!');

button2.Enabled:=false;

end;

procedure TForm10.Button3Click(Sender: TObject);

begin

form10.Close;

end;

end.

unit Unit11;

interface

uses

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

Dialogs, Grids, DBGrids, StdCtrls, ExtCtrls, DBCtrls, unit2;

type

TForm11 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Button1: TButton;

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form11: TForm11;

implementation

{$R \*.dfm}

procedure TForm11.Button1Click(Sender: TObject);

begin

if (edit1.Text<>'') and (edit2.Text<>'') then

begin

datamodule2.sotrtable.Insert;

datamodule2.sotrtable.FieldByName('фио').Value:=edit1.Text;

datamodule2.sotrtable.FieldByName('телефон').Value:=edit2.Text;

datamodule2.sotrtable.Post;

end

else showmessage('Заполните все поля!');

end;

end.

unit Unit12;

interface

uses

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

Dialogs, unit2, ExtCtrls, DBCtrls, Grids, DBGrids, StdCtrls;

type

TForm12 = class(TForm)

Edit1: TEdit;

Edit2: TEdit;

Button1: TButton;

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

Edit3: TEdit;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form12: TForm12;

implementation

{$R \*.dfm}

procedure TForm12.Button1Click(Sender: TObject);

begin

if (edit1.Text<>'') and (edit2.Text<>'') then

begin

datamodule2.klienttable.Insert;

datamodule2.klienttable.FieldByName('фио').Value:=edit1.Text;

datamodule2.klienttable.FieldByName('телефон').Value:=edit2.Text;

datamodule2.klienttable.Post;

end

else showmessage('Заполните все поля!');

end;

end.

**Последний лист выпускной квалификационной работы**

(является обязательным)

Выпускная квалификационная работа выполнена мной самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Фамилия. И.О.       \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
                                                   (подпись)  
"\_\_" \_\_\_\_\_\_\_\_ 200\_г.