

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерные Технологии

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: База данных "Учет компьютера и комплектующих"

Специальность 5В070400 ВТ и ИТ

Выполнил (а) Талалова А.Е. ВТУ-13-1
(Фамилия и инициалы) группа

Научный руководитель Аманбаев А.А. к.р.м.н., доцент
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурзаева А.М., к.э.н., доцент

(Фамилия и инициалы, ученая степень, звание)

А.М. « 19 » 05 2016 г.
(подпись)

по безопасности жизнедеятельности:

Мадалиев И.Ф., к.т.н., проф.

(Фамилия и инициалы, ученая степень, звание)

И.Ф. « 16 » 05 2016 г.
(подпись)

по применению вычислительной техники:

Аманбаев А.А. к.р.м.н., доцент

(Фамилия и инициалы, ученая степень, звание)

А.А. « 06 » 06 2016 г.
(подпись)

Нормоконтролер: Аманбаев А.А. к.р.м.н., доцент

(Фамилия и инициалы, ученая степень, звание)

« 06 » 06 2016 г.

(подпись)

Рецензент: Заурбек Нурган Сабирович, д.т.н., доцент

(Фамилия и инициалы, ученая степень, звание)

« 1 » 06 2016 г.

(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Автоматических и информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Талапова Айгерим Ержановна
(фамилия, имя, отчество)

Тема проекта База данных и УИТ компьютера и компьютерных

утверждена приказом ректора № 149 от «23» октября 2015 г.
Срок сдачи законченной работы «__» _____ 20__ г.
Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработать базу данных УИТ компьютера и компьютерных
ЕА-диаграммы, УИТ диаграммы и алгоритм работы ИО.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Исследование предметной области
Выбор средств реализации
Реализация проекта
Экономический анализ
Безопасность и надежность



Перечень графического материала (с точным указанием обязательных чертежей)

1. Схема словных модулей программы
2. Диаграмма компонентов
3. Диаграмма прецедентов
4. Диаграмма классов
5. Диаграмма развертки
6. ER-диаграмма
7. План задания
8. Схема расположения элементов

Рекомендуемая основная литература

- Сатимова Е.Г. Проектирование баз данных. Методическое указание к выполнению лабораторных работ, 2002г.
- Ерксиева З.Д., Бекмешева А.И. Методические указания к выполнению проектной части дипломных работ, 2013г.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БД	Мурлов И.Р.	11.03 - 16.05.16	
Эконом. часть	Бекмешева А.И.	02.03 - 19.05.16	
Нормоконтроль	Амандиев А.А.	11.05 - 05.06.16	
Словная часть	Амандиев А.А.	15.03 - 06.06.16	

Аннотация

Тема дипломного проекта: База данных «Учет компьютера и комплектующих».

Целью данного дипломного проекта является разработка базы данных, которая предназначена в колледж для учета компьютеров и комплектующих и создание специализированной СУБД, рассчитанной на упрощение круга задач для преподавателей.

В технико-экономическом обосновании произведен расчет затрат на научно-исследовательскую работу.

По безопасности жизнедеятельности рассмотрены вопросы условий труда, так же выполнен расчет кондиционирования рабочего помещения и определены нормы освещения для данного помещения.

Annotation

The theme of the degree project: The database system «Accounting for computer and accessories».

The purpose of the diploma project is the development of a database, which is intended to college to account for computers and accessories and create a specialized database, designed to simplify the range of tasks for teacher.

In the feasibility study calculated the costs of the research work.

In safety vital functions addressed issues of working conditions, also calculated the conditioning of the working premises and defined standards for the room lighting.

Андатпа

Дипломдық жобаның тақырыбы: «Компьютер және жинақтаушылар» деректер қоры.

Бұл дипломдық жобаның мақсаты деректер қорын құру, колледжге есепке алыну үшін арналған компьютер мен жинақтаушылар және мамандырылған МББЖ құру, мұғалімдер жұмысын жеңілдету үшін.

Ғылыми-зерттеу жұмыстарының құнын есептеу техникалық-экономикалық негіздемесі жасаған.

Бөлше жарықтандыру үшін үй-жайлар мен жиынтығы стандарттарын ауаны есептегін еңбек жағдайларын тіршілік әрекетін қауіпсіздігі.

Содержание

Введение.....	7
1 Общая часть.....	8
1.1 Постановка задачи	8
1.2 Инфологическое проектирование БД системы.....	9
1.3 UML диаграммы.....	10
1.3.1 Диаграммы компонентов.....	10
1.3.2 Диаграммы прецедентов.....	10
1.3.3 Диаграммы классов.....	11
1.3.4 Диаграммы развертывания.....	11
1.4 Обоснование выбора программно-аппаратных средств.....	12
1.4.1 Обоснование выбора СУБД.....	12
1.4.2 Обоснование выбора языка программирования.....	14
2 Специальная часть.....	17
2.1 Создание информационно-логической модели.....	17
2.2 Построение ER-диаграмм.....	19
2.3 Выявление и моделирование сущностей и связей.....	21
2.4 Разработка модели информационных потоков БД.....	21
2.5 Разработка алгоритмического обеспечения.....	22
3 Техническая часть.....	23
3.1 Описание форм системы.....	23
3.2 Руководство пользователя.....	25
3.3 Результаты проекта.....	26
4 Экономическая часть.....	27
4.1 Расчет затрат на разработку информационных технологий.....	27
4.2 Расчет цены программного продукта.....	35
5 Безопасность жизнедеятельности.....	36
5.1 Анализ условий труда.....	36
5.2 Расчет системы кондиционирования кабинета.....	38
5.3 Расчет естественного освещения кабинета.....	42
5.4 Расчет искусственного освещения.....	45
Заключение.....	48
Список литературы.....	50
Приложение А.....	51

Введение

Современное информационное поле состоит из многих событий, объектов и явлений. Она окружает такие объемы, что без точно, действуя с определенной системой, хранение всех этих данных смогло быть хаотическим и неудержимым.

База данных - набор данных, которые сохранены некоторый заказанный метод. Возможно сравнить базу данных с правительственным входом, который документы сохранены. Другими словами, база данных - склад данных. Базы данных само собой не смогут представить интерес, не было ли никаких систем (DMS) управления базой данных.

Система управления базой данных - установленный язык и программное обеспечение, которое обеспечивает доступ к данным, позволяет создавать их, чтобы измениться и удалять, гарантирует безопасность данных, и т.п. В общей системе управления базами данных - это - системное позволение создать базы данных и манипулировать данными из них. И обеспечивает этот доступ к данным системы управления базами данных посредством специального языка - SQL.

Цель базы данных - чтобы проанализировать потоки данных, рассматривая их в течение прогноза, создание статистических сообщений и в системах отчетности.

Отличительная особенность БД от других хранений подобно архивам есть, что информация в этом несомненно подвергнута обработке электронными СМИ с последующей систематизацией, которая субординированная к определенным правилам.

БД очень важны для непрерывного взаимодействия систем, содержащих глобальную информацию. Например, о государственных ресурсах и определенных территориях. У них все данные на структурах, гидрография и вегетация интересных мест находятся в деталях красил.

Профильная программа понуждала, чтобы производить специфические действия над доступной информацией носит имя системы управления базой данных. Любой пользователь имеет возможность самостоятельно установить к программе работа продукции информации в форме, в которой он захотел бы получить это.

Все объекты систематизированы на технических или физических знаках. И информация относительно объектов доступна к пользователям в следующих типах:

- в текстовом;
- в форме таблицы;
- как рисунок.

Все данные, которые имеют структурирующий вид, позволяют облегчать значительно их обработку и анализ, а также специальный выбор, сортируя, запросы пользователей и математические и логические действия.

Системы информации, образованные на платформе информационных баз, состоят из программного обеспечения, СУБД и непосредственно БД.

1 Общая часть

1.1 Постановка задачи

Передо мной была поставлена задача разработки базы данных на тему: «Разработка программы по учету компьютеров и комплектующих».

Понятие административной системы базой данных близко соединено с понятием базы данных. Это - комплекс программного обеспечения, предназначенного для создания структуры новой основы, автозаправочный к его содержанию, редактирование содержимого и визуализации информации.

Базы данных, возможно, содержат различные столы объектов. Современная жизнь невообразима без эффективного управления. Важная категория - системы обработки информации, на которой общая рабочая характеристика любого объекта или организации во многих отношениях зависит. Эта обработка уменьшена к автоматизации.

Автоматизация во многих отношениях упрощает действие. Посредством этого, чтобы быть сохраненным не только время, но и человек и технические ресурсы. Информация более или менее защищается от повреждения и потери. Автоматизированная база данных снабжается механизмами поиска и информационным выбором.

Без баз данных невозможно обеспечить действие большинства, финансового, индустриального, торгового и другой к организации сегодня. Информационные потоки, циркулирующие в мире, который окружает нас, огромны. Вовремя они стремятся возрасти. Нет никаких баз данных, мы задушили бы в информационной лавине давно. Понимайте хранение структурированных данных как базу (D) данных, в тех же часовых данных должен быть последователен, минимум, лишней, и завершаются. Обычно БД создана, ибо хранение и данные имеют доступ, содержа данные на некоторой области определений данных, это некоторая область человеческой деятельности или область реального мира. Любая БД должна представить систему данных на области определений данных. БД, имеющая отношение к той же области определений данных, содержат более или менее детальную информацию относительно нее в различных случаях. Уровень детализации определяет ряд факторов, в первую очередь цель использования информации от базы данных и сложности производства (бизнес) обрабатывает существование в пределах области определений данных в специфическом условии. Каждая таблица БД представляется как совокупность строк и столбцов, где строки соответствуют экземпляру

объекта, конкретному событию или явлению, а столбцы - атрибутам (признакам, характеристикам, параметрам) объекта, события, явления.

1.2 Инфологическое проектирование БД системы

Цель инфологической, проектирующей, - снабжение методов коллекции и подчинения информации, естественнее к особе, к этому который, как предполагается, сохранен в созданной базе данных. Поэтому, модель данных пробуются, чтобы быть строился по аналогии с естественным языком (выдержка не может быть использована в чистой форме из-за трудности компьютерной обработки текстов и двусмысленности любого естественного языка любым путем). Основные структурные элементы моделей - объекты, коммуникации между ними и их имущество.

Схемная БД может не быть успешна, т.е. могут быть избыточность и аномалии (аномалия обновления, аномалия включения, аномалия перемещения). Нормализация данных представляет процедуру, обеспечивающую согласие информации, моделируют к некоторым стандартам. Это подразумевает минимизацию дублирования, гарантируя необходимое гибкости для поддержки различных функциональных требований, и создания условий адекватного показа модели различным проектам БД.

Раскол в столах осуществляется согласно семантическому анализу предметной области, в то же время, как правило, к каждому объекту (сущность) предметной области стол помещается в согласие, к свойствам объекта там передают свойства стола, и к идентификатору объекта передают табличный ключ.

Процесс нормализации, идущий параллельно с дизайном, включает:

- отождествление существенных объектов, информация, относительно которой, подлежащая осмотру или хранению. Этим объектам придется исключить друг друга взаимно;
- дополнение коммуникаций, представляющих названные отношения между объектами;
- для каждой сущности список данных (свойства), которым нужно быть известным об этом, сформирован;
- , чтобы установить, как каждый случай сущности возможен, это уникально для выделения.

Для создания инфологических данных моделируйте удобный инструмент - ERwin (инструмент развития структуры базы данных) использован. Обычно развитие модели базы данных состоит из двух стадий: компиляция логической модели и создания на его основе физической модели. ERwin полностью поддерживает такой процесс, он имеет две идеи модели : логический и физический. Поэтому, разработчик может построить логическую модель базы данных, без думанья о деталях физического

выполнения, т.е., уделяя главное внимание требованиям к информации и бизнес-процессам, которые будет поддерживать будущая база данных.

В результате анализа 6 объектов, которые описывают эту предметную область, было распределено.

В проектируемой модели логико-физическая модель была использована.

1.3 UML диаграммы

Для моделирования статических объектов в объектно-ориентированной парадигме UML есть 4 типа диаграмм:

- а) диаграмма компонентов;
- б) диаграмма прецедентов;
- в) диаграмма классов;
- г) диаграмма развертывания.

1.3.1 Диаграмма компонентов

Диаграмма компонентов показывает набор компонентов и отношения между. Компонент - физически замененная часть системы, которая имеет набор интерфейсов и обеспечивает их выполнение.

В этом случае выполняемый ProgrammObes.mdb модуль обращается для получения или добавления информации KompTechn.mdb база данных. Но обращается не непосредственно, и через Доступ MS Access. Поэтому Доступ MS Access - интерфейс между программой и базой данных, обеспечивая их взаимодействие. Диаграмма компонентов обеспечена в рисунке 1.1.

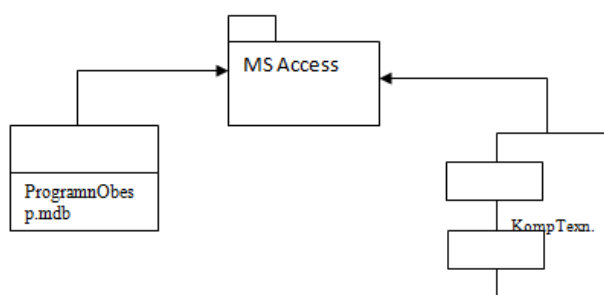


Рисунок 1.1 - Диаграмма компонентов

1.3.2 Диаграмма прецедентов

Диаграмма прецедентов или выбора использования применяется к характеристике взаимодействия пользователей с базами данных. Диаграмма прецедентов обеспечена в рисунке 1.2.

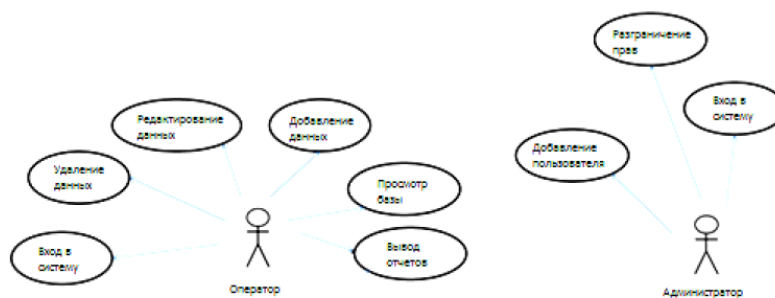


Рисунок 1.2 - Диаграмма прецедентов

1.3.3 Диаграмма классов

Классовая диаграмма в Uml - частный случай диаграммы ERwin. ERwin диаграммы использованы для логического проектирования баз данных. Их главная разница: В ERwin внимания диаграммы есть выплатил структуре данных, и в классовой диаграмме - к поведению классов. Классовая диаграмма обеспечена в фигуре 1.3.

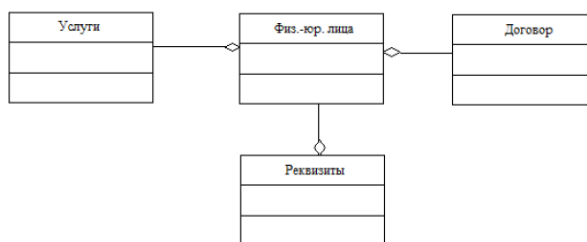


Рисунок 1.3 - Диаграмма классов

1.3.4 Диаграмма развертывания

На диаграмме расширения, или приложения, конфигурация узлов переработки, на которых система осуществляется, показан. Диаграмма расширения представляется в рисунке 1.4.

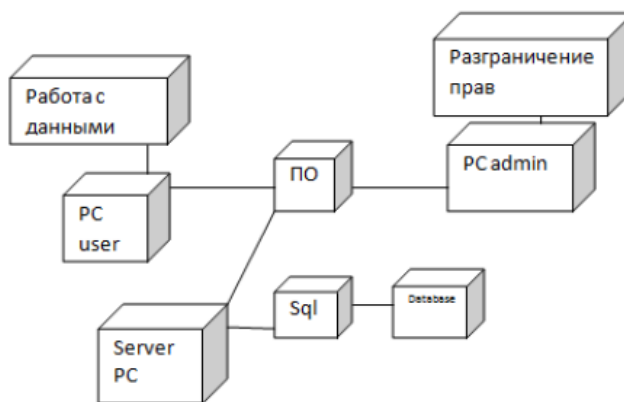


Рисунок 1.4 - Диаграмма развертывания

1.4 Обоснование выбора программно-аппаратных средств

1.4.1 Обоснование выбора СУБД

Одна из самых главных характеристик большинства баз данных - их непрерывное изменение и продолжение. В процессе добавления новых типов данных или в случае появления новых приложений возможность быстрой замены структуры базы данных должна быть обеспечена. Реорганизация базы данных должна осуществляться при любой возможности без копирования прикладных программ и вообще вызывают минимальное количество конверсий. Простота изменения базы данных может напрячь, большие влияют на развитие приложений баз данных в производственном управлении.

База данных - средство коллекции и организация информации. Базы данных могут содержать информацию о людях, продукции, заказывает, и т.п. Много баз данных изначально представляют список в процессоре слова или динамической электронной таблице. Так как список расширяется, в этом чрезмерные и противоречивые данные собирают. В списке формируют эти данные становятся труднее для понимания, и ищут способности или извлечение подмножеств данных для рассмотрения очень ограничены. Когда есть подобные проблемы, полезно переместить информацию в базу данных, созданную посредством системы (DMS) управления базой данных, например Офисный Доступ 2007.

Компьютерная база данных представляет хранение объектов. Одна база данных может содержать несколько столов. Например, система складского счета, в котором три стола использованы, - не три базы данных, но один, содержа три стола. В базе Доступа данных табличного остатка в одном файле вместе с другими объектами, как например формы, сообщения, тасгоес и модули, если бы база данных не предназначалась особенно для использования данных или кода от другого источника. Базы данных создали в формате Access 2007 имеют продолжение имени файла Acc.db, и базы данных создали в более ранних форматах Доступа - продолжение MDB. Приложение Access 2007 может быть использовано для создания файлов в более ранних форматах файла (например, Access 2000 и Access 2002-2003).

Приложение Доступа дает следующие возможности:

- дополнение новых данных в базе (например, новое изделие линии в складском инвентаре) данных;

- изменение существующих данных в базе (например, изменение текущего размещения изделия линии в складе) данных;

- перемещение данных (например, если изделие линии продается или дефектный);

- организация и визуальный отображая данных различными методами;

объединенное использование данных посредством сообщений, e-mail сообщений, внутренней сети или Интернета.

На появлении стол базы данных подобен динамической электронной таблице, в которой данные имеются в линиях и колонках. Поэтому динамические электронные таблицы обычно легко импортированы к столам базы данных. Главное различие между хранилищем данных в динамической электронной таблице и в базе данных - метод структуры данных.

Чтобы обеспечить больше всего гибкости базы данных, необходимо распространять данные на столах таким образом, что, чтобы избежать их избытка. Например, если информация о работниках хранится в основе, необходимо ввести каждый из них однажды к столу, который предназначается только для хранилища данных о работниках. Данные на продукции будут сохранены в отдельном столе, и данных на ветвях - в другом. Эта процедура - называемая нормализация.

Формы иногда - называемые окна ввода данных. Они - интерфейсы, которые использованы для работы с данными, и часто содержат кнопки для выполнения различных команд. База данных может быть создана без помощи форм, только вводя столы это в методе в столе. Однако большинство из пользователей баз данных предпочитает просмотреть, вводят и редактируют данные столов посредством форм.

Формы позволяют работать с данными в удобном формате; кроме, возможно добавить функциональные элементы к ним, например кнопки команд. В программном пути эти кнопки могут назначить выполнение различных задач, как например определение данных, показанных в форме или вводная часть других форм или сообщений. Например, возможно создать форму с именем "Форма Клиента" для работы с данными клиента. В форме клиента может быть кнопка, открывающая бланк заказов, в котором новый заказ для этого клиента создан.

Формы также позволяют устанавливать рабочие состояния других пользователей с информацией, которая содержится в базе данных. Например, возможно создать форму строго ограничил набор показанных полей и позволенных действий. Он помогает защищать данные и гарантировать корректность их вклада.

Сообщает подачу для коллекции и представления данных, содержащего в столах. Обычно сообщение позволяет отвечать на определенный вопрос, например: "Сколько денег были получены от каждого клиента в этом году?", "В том, что города есть клиенты нашей компании?" Каждое сообщение может форматироваться таким образом, что, чтобы подарить данные в самой удобной форме.

Сообщение может быть запущенный в любое время, и это будет всегда отражать текущие данные в базе данных. Обычно сообщает формат для печати, но они могут также быть просмотрел на экране, чтобы экспортировать к другой программе или послать в форме e-mail сообщений.

Запросы являются главным рабочим инструментом базы данных и могут выполнять набор различных функций. Самая широко распространенная функция запросов - извлечение определенных данных из столов. Данные, которые нужно рассматривать, как правило, находятся в нескольких столах; запросы позволяют обеспечивать их в одном столе. Кроме, как обычно не требуется рассматривать все записи непосредственно, посредством запросов это возможно, установив ряд условий, чтобы "фильтровать" только необходимые записи. Часто запросы - рекордный источник для форм и сообщений.

Некоторые запросы обеспечивают обновление: это означает, что данные главным образом столы могут быть изменены через стол запроса. Работая с запросом с обновлением, необходимо помнить, что изменения фактически делаются не только к столу запросов, но и к соответствующим столам базы данных.

Есть два главных вида запросов : запрашивает для выбора и изменяют запросы. Запрос на выбор только берет данные и дает возможность пользоваться ими. Результаты такого запроса могут видаться на экране, чтобы распаковаться или копировать в буфере обмена. Кроме, они могут быть использованы как источник записей для формы или сообщения.

Запрос изменения, поскольку образуется из его имени, выполняет действия с данными. Запросы изменения могут быть использованы для создания новых столов, дополнения данных в существующих столах, обновляя или перемещение данных.

Один из программирования средств в Access - язык макрокоманд. Программы, созданные на этом языке, - называемый макросами и отдельные действия осознали посредством форм, запросов, сообщения позволяют соединяться легко. Макросами события покрова, которые вызывают действия пользователями действием с данными диалога через формы или системными событиями.

Оказывается этот Доступ, обладая всеми линиями системы управления базами данных, дает также дополнительные возможности. Это не только гибкая и простая система управления базами данных в использовании, но и системный для развития приложений, работающих с базами данных.

1.4.2 Обоснование выбора языка программирования

Окружающая среда Delphi - трудный механизм, гарантирующий чрезвычайно эффективное функционирование программиста. Визуально она осознает, что несколько окон, которые есть в то же время, открылись на экране. Windows может пойти дальше экрана, частично или полностью, чтобы блокировать друг друга. Компилятор, который есть, строился в "клиентском сервере" Delphi обеспечивает высокоэффективное

необходимое для прикладного программирования в архитектуре. Этот компилятор - один из самого быстрого в мире сейчас.

Изучив главные термины и БД понятий, возможно перейти к рассмотрению средств, доступный разработчик, создающий приложения БД посредством Delphi. Средство Delphi, предназначенных для работы с БД, может быть делится на 2 категории: инструменты и компоненты. Инструменты включают ряды дополнительные программы, вводящие доставку Delphi, и предназначался для работы с БД. Относительно компонентов, в VCL есть набор специализирован компонент, предназначенный для работы с БД.

Delphi - язык программирования и условия разработки программного обеспечения. Это развивает Borland (раньше известно как Inprise). Язык программирования Delphi, раньше известный, так как Объектный Паскаль (Паскаль с объектно-ориентированными продолжениями) первоначально ориентировал только на Microsoft Windows, но позволяет создавать сейчас частные приложения для Linux и Microsoft. Сетевая структура, и другие (посмотрите ниже). Программирование для начинающих.

Самое популярное использование - развитие настольных прикладных систем и базы данных предприятия и как инструмент язык программирования Delphi может быть использован также для большинства видов развивающихся проектов. Это было одним из первых языков, которые стали известными как инструмент RAD (быстрое прикладное развитие), когда оставлено в 1995. Дельфах 2, которые уходили годом позже, поддерживал 32-разрядная окружающая среда Windows и но только несколькими годами позже были C++, C. В 2001 версия под ОС Linux, известной, так как Kylix (классическая коробка Греческого избирательного бюллетеня) стал доступным. Каждый год было около одной новой версии, в 2002 продукт стал известным как Студия Delphi 7, язык начал быть называемым официально язык программирования Delphi, но не Объектный Паскаль, а также поддержка Linux была добавлена (через Kylix) и сеть (через Предварительный компилятор). Полная поддержка сеть есть планировал для предстоящих Delphi 8. Программирование для заварных чайников.

Главные компоненты Delphi и Kylix : язык Delphi (раньше официально известно как язык Объектного Паскаля), Vcl / Clx (Визуальная Составляющая Библиотека), близкие соединения с базами данных, в комбинации с мощным IDE (Интегральная Среда Разработки) и дополнительными инструментами поддержки.

Замечательные особенности языка Delphi включают:

Прозрачная обработка объектов через связывается или указатели

Имущество как часть языка, вместе с функциями Получения и Агрегирующими, которые являются прозрачной герметизацией доступа к членам к полям

Имущество индекса и имущества по умолчанию которые обеспечивают доступ к коллекции с удобным и прозрачным методом

Делегаты или в методах различного способа указателей безопасного типа которые использованы для приведения в действие события, вызванного компонентами

Делегирование выполнения интерфейса в поле или имуществе класса

Простота процессоров выполнения Windows сообщение, отмечают метод класса о номер / имя окна сообщений для обработки

Большинство функций, перечисленных выше, были введены в Delphi первой и адаптированы на других языках позже.

Главный Андерс Хейлсбергом был архитектором Delphi, и его предшественником Ускоренного Паскаля, до это не превращалось в Microsoft в 1996 посвященный бизнес к программированию сущностей.

Продукт Delphi простирается в различных наборах, каждый из которого предложен большую функциональность в сравнении с другими:

- Личный
- Профессиональный
- Предприятие
- Архитектор

Уважительные причины для использования Delphi:

Очень информативный и полезный для общества новостей

Может компилироваться в одном, выполняемом, упрощая распространение и сокращение вопросов с различным DLL

VCL и трехсторонние компоненты есть, как правило, доступный с полным исходным кодом

Мощный и быстро оптимизирующий компилятор

От одного исходные кодовые превосходные машинные коды для различной ОС оказываются

Поддержка самых последних технологий и стандартов

Без есть способный, чтобы удовлетворить всю необходимость программистов, есть целый набор дополнительных ресурсов, которые вынуждены расширить ряд разрешимых задач посредством Дельф, или это может быть сделано посредством кодовой вставки на других языках, там, где Дельфы и Kylix бессильны.

Эти замены Delphi могут быть использованы в случае невозможности заплатить среду разработки, нужно для бесплатных сред разработки (например, для операционных систем и свободно распространял программное обеспечение или в образовательных целях). В большинстве случаев эти альтернативы использованы для обучения, а также для написания серверных компонентов, для операционных систем, которые широко не распространены (так был до не был Kylix для ОС Linux который использован в массе как ОС для серверов).

База данных может развиваться для пакетной обработки данных, переработка в реальном времени или оперативная обработка (в этом случае,

переработка каждого запроса прибывает в конец к определенному времени, но в то же время в течение периода обработка жесткие ограничения, существующие в системах реального времени, не добавлены). Во многих базах данных предусмотрен набор этих методов обработки, и во многих системах обслуживание терминалов в реальном времени случается с базами данных наряду с пакетной обработкой данных.

Больше всего части диска или библиотек лент, которые существовали перед использованием управлений, базируются данных, содержат большой ряд повторяющейся информации. В случае хранения многих данных записываются по пунктам избыточность как те же данные, отмечающиеся в информационных СМИ для различных целей, позволялось и, кроме, различных выборов модификаций тех же данных был сохранен. База данных предоставляет возможность избавиться существенно от такого избытка. База данных определяется иногда как не дополнительный набор элементов данных. Однако фактически для снижения времени доступа к данным или упрощения способов адресации избыточность в незначительном уровне есть присутствуют на многих базах данных. Некоторые записи повторяются, чтобы обеспечить возможность выздоровления данных в случае их случайной потери. Что база данных не была лишней и соответствовала другим требованиям, необходимо сделать компромисс. В этом случае скажите об управляется, или минимум, избытки или что от хорошей разработанный базы данных свободно. В этом случае говорят об управляемой, или минимальной, избыточности или о том, что хорошо разработанная база данных свободна от излишней избыточности.

Неудержимая избыточность имеет несколько недостатков. Для начала, хранение нескольких экземпляров данных приводит к дополнительным затратам. Во-вторых, обновляя, по меньшей мере, из нескольких чрезмерных копий необходимо осуществлять повторные сделки обновления. Избыточность поэтому затраты значительно дороже, когда при обработке представляют большой ряд информации обновлен или, что есть даже худшие, новые элементы часто вводятся или старый уничтожены. В-третьих, потому что различные экземпляры данных могут соответствовать различным стадиям обновления, информация, выпущенная системой, может быть несовместима.

2 Специальная часть

2.1 Создание информационно-логической модели

Самое главное преимущество понятия баз данных вводит набора стандартных структур, в которых, как в контейнерах, данные помещены. Планирующая работа с данными в специфической предметной области,

после объяснения структур данных главных целей решают проблемы: как группировать данные в столах, что поля и что типы, чтобы обеспечить в каждом столе, как соединить столы друг с другом, и т.п. решение этого комплекса вопросов - называемое создание информационно-логической (инфологической) модели.

Модель инфологический (информационно-логическая модель) - ориентирован на особу и модель предметной области, не зависящий от типа системы управления базами данных, определяя наборы информационных объектов, их свойств и отношений между объектами, динамикой изменений области определений данных, а также природы информационной необходимости пользователей. Инфологическая модель предметной области может описать объект - модель коммуникации, которая разделение краеугольного камня реального мира в отдельных осязательных объектах, которые находятся в определенных коммуникациях друг с другом и обе категории, есть - объект и коммуникация полагаются с первоначальными, неопределенными понятиями.

Система (DMS) управления базой данных - сложный язык и программное обеспечение, предназначенное для создания, редактирование содержимого и информационной визуализации. Информационная визуализация основы понимается как выбор показанных данных в согласии с данным критерием, их упрощением, дизайн и последующая продукция на устройствах вывода или передачах на коммуникации связывается, ведя и объединенное использование БД многими пользователями.

Объекты БД есть:

- Столы делают основу базы данных и предназначаются для хранения информации относительно объектов объектной области.
- Запросы - средство выбора необходимых данных из одного или несколько таблиц БД.

Перед стартовым созданием БД необходимо развить схему данных. Схема данных - структура основы, и, точно, структура или структуры главных столов. Для создания базы данных следующее действие выполняется:

- Полный список полей сформирован.
- Определены самые подходящие данные печатаются для каждого поля.
- Области полного списка согласно основным столам распространяются.
- В каждом из столов планируют ключевое поле или пользуются дополнительным встречным полем.
- Схематично, на отношения крестового стола (оно - схема данных) бумажной линии.

Существует несколько типов связей между таблицами:

- "связь 1:1".
- "связь 1:M".
- "связь M:N".

Перед стартовым созданием БД необходимо развить схему данных.

The depending на этом, сколько копий сущности одного класса соединены, со сколькими копиями сущности другого класса, отличаются следующие виды коммуникаций :

- коммуникация 1: 1. Единственная копия сущности одного класса соединена с единственной копией сущности другого класса.
- коммуникация 1: M. однородная копия сущности одного класса соединена со многими копиями сущности другого класса.
- коммуникация M : N. копии некоторого Количества сущности одного класса соединены с несколькими копиями сущности другого класса.

Схема данных - структура основы, и, точно, структура или структуры главных рабочих столов. Для создания базы данных следующее действие выполняется:

- Полный список полей создан.
- Это определено самые подходящие данные печатается для каждого поля.
- Области полного списка согласно главным рабочим столам простираются.
- В каждом из рабочих столов планируют ключевое поле или использование.

На первой стадии дизайна базы данных необходимо определить цель создания базы данных, его основных функций и информации, которую это должно содержать. Это - то, что необходимо определить главные темы столов базы данных и информации, которую табличные поля будут содержать.

Таким образом, инфологическую модель можно изобразить в виде связи сущностей Компьютерная техника, Программное обеспечение, Комплектующие, Перемещения, Ремонт и Списание.

Здесь используются связи один - ко - многим (1:M), связывающие ключевые поля таблиц Программное обеспечение, Комплектующие, Перемещения, Ремонт и Списание с соответствующими полями таблицы Компьютерная техника.

Схематическое изображение инфологической модели без указания названий полей и вспомогательных таблиц представлено на рисунке 2.1.

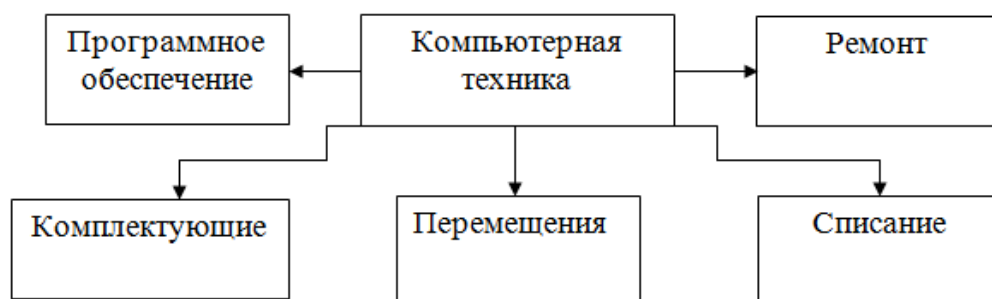


Рисунок 2.1 - Схематическое изображение инфологической модели

Другим основным понятием для описания предметной области является понятие связи. Для представления связей между экземплярами

сущностей могут использоваться атрибуты. В этом случае связь устанавливается путем включения в совокупность атрибутов сущности атрибута, однозначно идентифицирующего экземпляр сущности, находящийся в отношении с исходным экземпляром сущности.

2.2 Построение ER-диаграмм

Чаще всего концептуальная модель есть присутствовал в форме диаграммы объектов - коммуникации (объект - отношения) или диаграммы ER.

Процесс создания диаграммы ER - называемое моделирование ER. В то же время классические обозначения следования использованы.

Класс объектов есть присутствовал в форме четырехугольника. В четырехугольнике уникальное имя класса сущности (большие буквы) и имен свойств записывают строчные буквы.

Коммуникации между объектами обозначают стрелки, возле стрелок имя коммуникации, а также максимальная мощность коммуникации (максимальное число объектов, которые могут участвовать в коммуникации) введена. Чтобы показать, что сущность обязана участвовать в коммуникации (каждой копии придется быть соединенным с копией другого класса), в линии перпендикуляра места линии связи и, чтобы показать, что сущность может (но это не обязательно), чтобы участвовать в коммуникации, в линии связи размещают овал.

В случае описания объектов обычно выбирают специальные наборы свойств - ключи и внешние ключи. Ключ это уникально идентифицирует копию объекта и, вместе с внешним ключом, это использовано для выполнения коммуникаций. На диаграммах свойства, вводящие первичный ключ, сделаны ударение.

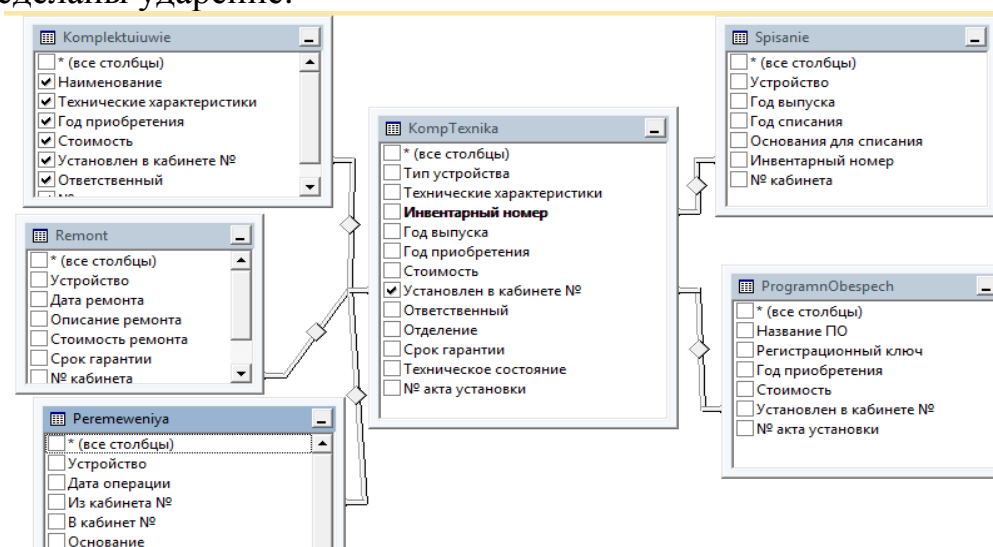


Рисунок 2.2 - ER-диаграмма

Согласно этой диаграмме, независимые объекты изображаются в форме нормальных прямоугольников, зависимый - в форме

прямоугольников с закругляемыми углами. Линия в прямоугольниках отделила ключи объекта (над линией). Конец находится в прикосновении помещают от детского объекта. "Ключ" письма возле пункта означает, что это соответствует одной копии родительского объекта 1 или много копий детского объекта (отсутствие значения исключено). Это вытекает из логики заполнения базы данных. Перечень указанных выше логических структур и характеристик приведены в таблицах 2.1 - 2.6.

Таблица 2.1 - Описание логической структуры таблицы «Компьютерная техника»

Тип устройства	Технические характеристики	Инвентарный номер	Год выпуска	Год приобретения	Стоимость	Установлен в кабинете №	Ответственный	Отделение	Срок гарантии	Техническое состояние	№ акта установки
ПК	Windows XP	23568	02.03.2005	01.05.2006	45000	302	Гайсагалеева Б.М.	ВТ и ПО	1	Работает	123456
Ноутбук	Windows 7	45462	05.08.2010	01.09.2011	60000	306	Шацкая Н.П.	Бухгалтерия	3	Исправно	321654

Таблица 2.2 - Описание логической структуры таблицы «Программное обеспечение»

Название ПО	Регистрационный ключ	Год приобретения	Стоимость	Установлен в кабинете №	№ акта установки
Intel Core i5	546486	01.01.2014	25000	302	56896

Таблица 2.3 - Описание логической структуры таблицы «Комплекующие»

Наименование	Технические характеристики	Год приобретения	Стоимость	Установлен в кабинете №	Ответственный	№ акта установки
Мышка	Laser Jet	01.01.2014	1500	302	Гайсагалеева Б.М.	56875
Принтер	HP	01.01.2014	20000	302	Гайсагалеева Б.М.	87546

Таблица 2.4 - Описание логической структуры таблицы «Перемещения»

Устройство	Дата операции	Из кабинета №	В кабинет №	Основание	ФИО старого ответственного	ФИО нового ответственного	№ акта установки
Принтер	01.01.2015	302	306	По приказу	Гайсагалеева Б.М.	Шацкая Н.П.	45865

Таблица 2.5 - Описание логической структуры таблицы «Ремонт»

Устройство	Дата ремонта	Описание ремонта	Стоимость ремонта	Срок гарантии	№ кабинета	№ акта приемки
Сканер	01.01.2014	Очистка	3500	1	302	46568

Таблица 2.6 - Описание логической структуры таблицы «Списание»

Устройство	Год выпуска	Год списания	Основания для списания	Инвентарный номер	№ кабинета
ПК	01.01.2001	01.01.2007	Не исправно	65524	306

2.3 Выявление и моделирование сущностей и связей

Развивая концептуальную модель, в первую очередь, необходимо определить объекты. Для этой цели необходимо сделать следующее:

- необходимо понимать, какой информации придется быть сохраненным и быть обработанным и является ли это возможным определить это как сущность;

- чтобы присвоить этой сущности имя;

- чтобы выявить свойства сущности и присвоить им имя.

- имея выявил объекты, необходимо определить, какие коммуникации доступны между ними.

- определяя коммуникации (естественно, мы рассматриваем только те коммуникации, которые есть в родстве с разрешимые проблемы обработки данных), необходимо рассматривать следующее:

- как копия одной сущности соединена с копией другой сущности;

- как установлены, что была возможность ответа ко всем запросам пользователей приходится быть установленным (исхождение из их информационных требований).

На стадии начального развития БД больше всего части полезной информации получается в опросе пользователей системы. Поэтому стадию разработки концепций построения системы нужно начать с отождествления достоверных источников информации. Мы сделаем список потенциальных пользователей системы.

Источники информации :

Администраторы базы данных - на основании официальных документов и отчетность формируют структуру БД, изменяя и добавляя ее.

Операторы - предоставляют информацию.

Пользователи - предоставляют информацию.

2.4 Разработка модели информационных потоков БД

Информационный обмен в пределах проектируемой программы контроля базы данных системы представляет коммуникацию нескольких компонентов :

- взаимодействие между системным администратором и информационным компонентом проектировало систему посредством интерфейса администрации;

- информационный обмен между компонентами Delphi 7 и MS Access базой данных;

- информационный обмен между административной системой БД и пользователями этой системы.

Delphi 7 - представляет интегральную крышку разработчика, который набор специализирующихся программ, ответственных за различные стадии создания готового приложения, входит.

БД - база данных, которая содержит информацию относительно пользователей системной и образовательной и методической литературы.

Администратор системы добавляет системе пользователей, распространяя их, а также устанавливая права доступа, имеет права на дополнения услуг.

Оператор имеет доступ к системе, он имеет возможность просмотреть доступную информацию, чтобы удалить или добавить новые данные, чтобы работать согласно договору и заполнять дополнительные договоры.

2.5 Разработка алгоритмического обеспечения

Алгоритм - набор инструкций, описывающих процедуру действий исполнителя для достижения результата решения задачи для конечного ряда действий. Это выполняет конечную последовательность шагов, ведущий от исходных данных к требуемому результату.

Алгоритм может быть вычитан любой инструкцией, если:

- его команды не позволяют различный выбор выполнения;

- инструкции обеспечены для всего возможного выбора последовательности событий.

Интерфейс развиваемой базы данных может нести стандартному интерфейсу Windows.

Для добавления, редактирования и удаления данных применяются формы с расположенными на них элементами управления.

Общий алгоритм работы программы представлен на рисунке 2.3.

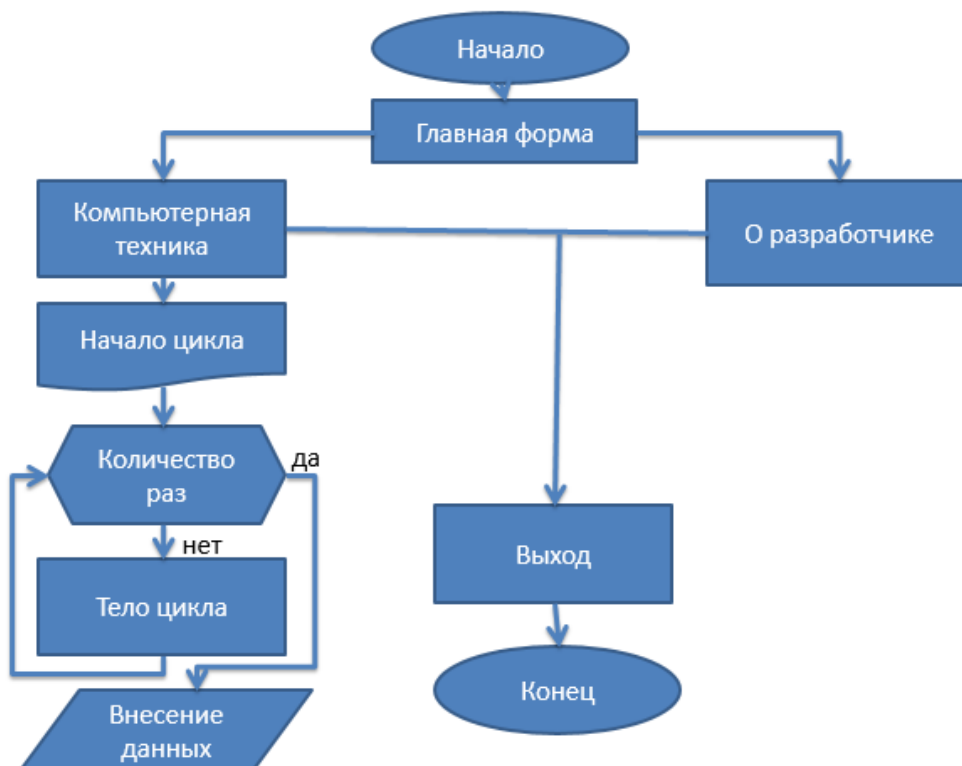


Рисунок 2.3 - Общий алгоритм работы программы

3 Техническая часть

3.1 Описание форм системы

Данный программный продукт разработан с целью выполнения вышеописанных функций и, исходя из специфики функционирования организации, прогнозируется высокая эффективность использования имеющихся вычислительных ресурсов. Начало работы с ПО начинается с того, что открывается главная форма, форма изображена на рисунке 3.1.

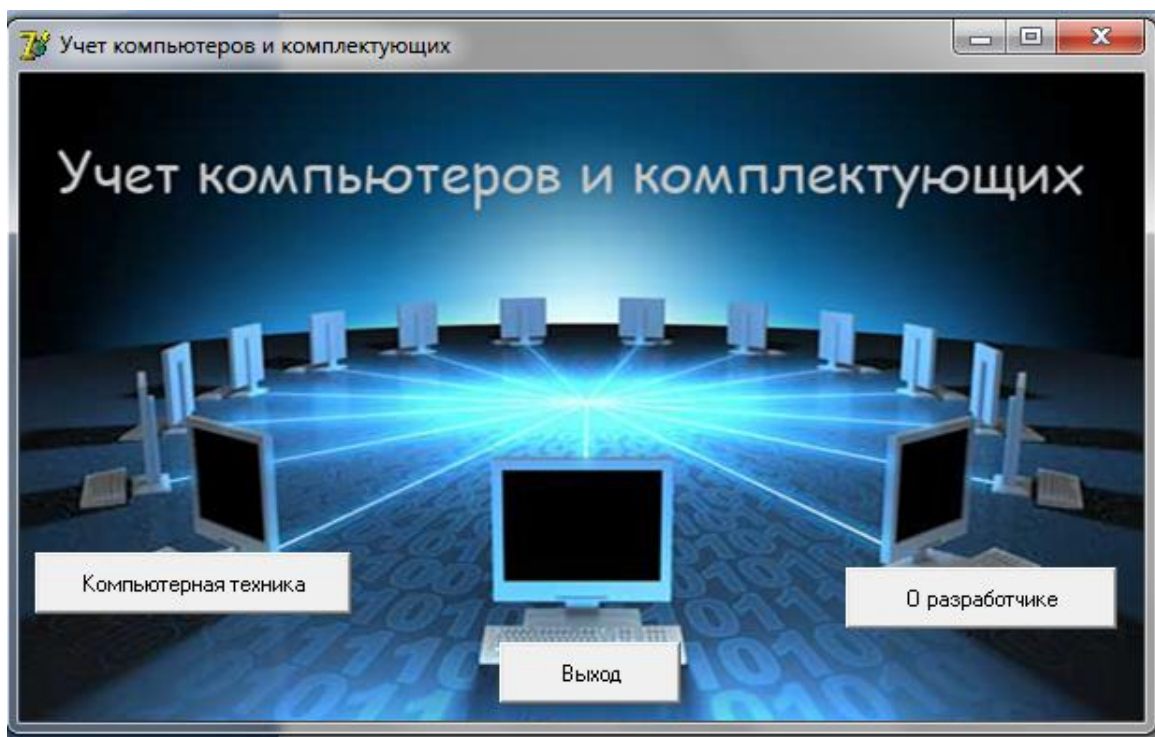


Рисунок 3.1 – Главная форма

При нажатии на «Компьютерная техника» открывается вторая форма которая изображена на рисунке 3.2.

Тип устройства	Технические характеристики	Инвентарный номер	Год выпуска	Год приобретения	Стоимость	Установлен в кабинете №	Ответственный	Отделение	Срок гарантии	Техническое состояние	№ акта установки
ПК	Pentium IV	123063	12.03.2005	23.05.2006	46000	302	Гайсагалиева Б.М.	ВТ и ПО	6	Рабочее	4654
ПК	Core i3	123005	05.06.2009	06.08.2010	47000	301	Жаңадултова Б.М.	ВТ и ПО	6	Рабочее	5644
ПК	Celeron	123018	03.06.2006	21.09.2007	48000	306	Смагулова Н.К.	ВТ и ПО	8	рабочее	5164
ПК	Pentium III	123086	05.09.2008	06.08.2009	45000	502	Тулгенова Э.Н.	ВТ и ПО	5	Рабочее	5164
ЗВМ	Intel Core i5	123589	02.06.2007	03.05.2008	49000	502	Тулгенова Э.Н.	ВТ и ПО	8	рабочее	56415
Латтон	Intel Inside	56897	20.01.2013	01.01.2014	70000	306	Иванова А.Р.	Бухгалтерия	5	Работает	78965
Ноутбуки	Dell	45789	01.01.2014	01.03.2014	85000	302	Петрова Л.И.	Бухгалтерия	5	Исправно	45788
Ноутбуки	Samsung	56895	01.01.2014	01.02.2014	75000	302	Акишев С.Л.	Бухгалтерия	6	Исправно	45789
Ноутбуки	Intel Core	54789	01.01.2013	01.02.2013	75000	302	Килинова А.Ж.	Бухгалтерия	5	Работает	36854

Название ПО	Регистрационный ключ	Год приобретения	Стоимость	Установлен в кабинете №	№ акта установки
Windows XP	1565	01.01.2007	15000	302	541654

Рисунок 3.2 – Вторая форма

На этой форме размещены две таблицы. Первая таблица – главная, а вторая связана с главной таблицей. На панели размещены 5 компонентов RadioGroup, при нажатии на которой меняется вторая таблица. Эти таблицы связаны с главной таблицей. Поиск, фильтрация, отчет и сортировка осуществляется по главной таблице. При добавлении записи, запись попадает во все таблицы. Выходим из этой формы нажатием на кнопку «Назад».

Внесение данных

Компьютерная техника

Тип устройства: Технические характеристики: Инвентарный номер: Год выпуска:
 Год приобретения: Стоимость: Установлен в кабинете №: Ответственный:
 Отделение: Срок гарантии: Техническое состояние: № акта установки:

Програмное обеспечение

Название ПО: Регистрационный ключ: Год приобретения:
 Стоимость: Установлен в кабинете №: № акта установки:

Комплектующие

Наименование: Технические характеристики: Год приобретения: Стоимость:
 Установлен в кабинете №: Ответственный: № акта установки:

Наименование	Технические характеристики	Год приобретения	Стоимость	Установлен в кабин
Сканер	Samsung	01.01.2014	12000	

Перемещения

Устройство: Дата операции: Из кабинета №: В кабинет №:
 Основание: ФИО старого ответственного: ФИО нового ответственного: № акта установки:

Ремонт

Устройство: Дата ремонта: Описание ремонта: Стоимость ремонта:
 Срок гарантии: № кабинета: № акта приемки:

Данные о списании

Устройство: Год выпуска: Год списания:
 Основания для списания: Инвентарный номер: № кабинета:

Перемещение по записям:

Рисунок 3.3 - Заполнение и изменение таблиц

Книжка1 - Microsoft Excel

Тип устройства	Технические характеристики	Инвентарный номер	Год выпуска	Год приобретения	Стоимость	Установлен в кабинете №	Ответственный	Отделение	Срок гарантии	Техническое состояние	№ акта уст
ПК	Pentium IV	123063	12.03.2005	23.05.2006	46 000,00р.	302	Гайсагалиева Б.М.	ВТ и ПО	6	Рабочее	
ПК	Core i3	123005	05.06.2009	06.08.2010	47 000,00р.	301	Жандаулетова Б.М.	ВТ и ПО	6	Рабочее	
ПК	Celeron	123018	03.06.2006	21.09.2007	48 000,00р.	306	Смагулова Н.К.	ВТ и ПО	8	рабочее	
ПК	Pentium III	123086	05.09.2008	06.08.2009	45 000,00р.	502	Тулгенова Э.Н.	ВТ и ПО	5	Рабочее	
ЭВМ	Intel Core i5	123589	02.06.2007	03.05.2008	49 000,00р.	502	Тулгенова Э.Н.	ВТ и ПО	8	рабочее	
Латтоп	Intel Inside	56897	20.01.2013	01.01.2014	70 000,00р.	306	Иванова А.Р.	Бухгалтерия	5	Работает	
Ноутбук	Dell	45789	01.01.2014	01.03.2014	85 000,00р.	302	Петрова Л.И.	Бухгалтерия	5	Исправно	
Ноутбук	Samsung	58965	01.01.2014	01.02.2014	75 000,00р.	302	Акишев С.Л.	Бухгалтерия	6	Исправно	
Ноутбук	Intel Core	54789	01.01.2013	01.02.2013	75 000,00р.	302	Килимова А.Ж.	Бухгалтерия	5	Работает	

Рисунок 3.4 - Формирование отчета в Excel

Просмотрев все выше показанные рисунки ПО можно увидеть, что в системе есть поиск, он осуществляется по информации, указанных в полях заполнения.

3.2 Руководство пользователя

Функциональное назначение

Развиваемый проект предназначается для выполнения требуемого круга задач, которые работники колледжа на бухгалтерском учете компьютеров и аксессуаров в течение целого периода учебного лица.

Эксплуатационное назначение

Программе придется быть управляемым в колледже, образование на приготвлении и предоставляя сертифицированным специалистам.

Персоналу этой организации придется быть заканчивают пользователей программы.

Состав функций

Эта система должна обеспечить следующие главные функции:

- к рекордной пригодности компьютеров на обслуживании и дополнительные аксессуары;
- чтобы управлять выполнениями;
- чтобы проводить прослеживание своевременности оплаты приобретения компьютеров;
- чтобы обеспечить администрацию клиентской основы;
- чтобы записать документы, которые прибыли на обработку;
- чтобы создать отчеты о запросах пользователя;
- чтобы дифференцировать права доступа в системе.

Климатические условия эксплуатации

Климатические условия действия, в котором установленным характеристикам придется быть обеспеченным, приходится навстречу требования, наложенные к техническим средствам относительно условий их действия.

Минимальный состав технических средств

Разрабатываемая система предназначена для использования на персональных компьютерах типа IBM с центральным процессором Intel Pentium III 1000ЕВ или мощнее (рекомендовано Intel Pentium IV), ОЗУ объемом не менее 256 Мб, свободным дисковым пространством не менее 100Мб. Обязательным условием является наличие манипулятора типа «мышь», SVGA-адаптера и монитора с разрешением не менее 1024x768. Дополнительно необходимо наличие принтера для возможности распечатки отчетов.

Минимальный состав программных средств

Приложение разработано с применением среды разработки Delphi 7 и СУБД MS Access 2007. На компьютере должна быть установлена среда визуального программирования Borland Delphi 7.0 и пакет MS Office 2007.

Приложение предназначено для работы в операционных системах семейства MS Windows 7/8/XP. Отчеты, создаваемые приложением, должны быть совместимы с MS Word и MS Excel.

3.3 Результаты проекта

В случае проектного выполнения на информатизации главная цель развития программы - высококачественное усовершенствование работы вообще, а именно сокращая трудовые затраты в случае выполнения тех же сделок.

Высококачественные выплаты:

- легкость в установке и действии;
- развиваемая программа может быть использована на любом числе компьютеров в одной организации;
- программа позволяет записывать компьютеры и общепринятые аксессуары, это значительно сокращает вероятность ошибок, соединенных с "человеческим фактором";
- низкие требования к аппаратной и высокой эффективности.

4 Экономическая часть

4.1 Расчет затрат на разработку информационных технологий

Вычисление полных затрат развития на проектное решение в форме информационных технологий ($C_{\pi i}$) осуществляется по формуле

$$C_{\pi i} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + A + \Pi_{\text{зи}} + P_{\text{ни}}, \quad (4.1)$$

где

$Z_{\text{фот}}$ – основной фонд компенсации разработчиков, тенге;

$Z_{\text{сзи}}$ – назначения на социальном налоге, тенге;

M_i – затрат на материал, тенге;

A – амортизаций;

$P_{\text{ми}}$ – затраты, связанные с эксплуатацией техники, тенге;

$\Pi_{\text{зи}}$ – прочие затраты, тенге;

$P_{\text{ни}}$ – накладные расходы, тенге.

Размер фонда заработной платы разработчиков ($Z_{\text{фот}}$) рассчитывается по формуле

$$Z_{\text{фот}} = Z_{\text{oi}} + Z_{\text{ди}}, \quad (4.2)$$

где

Z_{oi} – основные заработные платы, тенге;

$Z_{\text{ди}}$ – дополнительные заработные платы, тенге.

Трудовые затраты зависят от количества и трудового вклада развития программного обеспечения.

Итог равняется (V_0) программный продукт определен, исходя из количества и количества функции, осознанной программой

$$V_0 = \sum_{j=1}^n V_i, \quad (4.3)$$

где

V_i – объем разных функции ПО;

n – всего чисел функций.

Расчет уточненного объема ПО представлен в таблице 4.1.

$$V_0 \approx 10000$$

Таблица 4.1 - Перечень и объем функций программного модуля

№ функции	Наименование (содержание)	Объем функции (LOC)	
		по каталогу V_i	уточненный V_{yi}
1	2	3	4
201	Генераций структур БД	4 350	1 000
203	Формирование БД	2 180	850
204	Обработка наборов и записей баз данных	2 650	565
206	Обслуживание баз данных в интерактивном режиме	6 950	600
207	Манипулирование данными	9 550	3 895
208	Организаций поисков и поиск в БД	5 580	1 260
707	Графический вывод результатов	480	1 295
	Итого:	31 740	9 465

Общий трудовой вклад маленьких проектов выплачивает полностью на формуле

$$T_0 = T_n \cdot K_c \cdot K_r \cdot K_n, \quad (4.4)$$

где

T_n – нормативная трудоемкость;

K_c – коэффициент, учитывающий сложность ПО;

K_r – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

K_n – коэффициент, учитывающий степень новизны ПО.

Посредством коэффициента сложности, показанного в таблице 4.2, это рассматривается, рабочие затраты соединились со сложностью программного обеспечения [4].

В развиваемом проекте K_c , из-за присутствия в программном модуле в то же время две характеристики:

- рабочий режим в реальном времени;
 - управление удаленными объектами;
- Принимаем $K_c = 0,17$ из таблицы 4.2.

Таблица 4.2 - Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения K_c
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,07
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,08
4. Наличие у ПО одновременно нескольких характеристик	
4.1. 2 характеристики	0,13
4.2. 3 характеристики	0,17
4.3. Свыше 3-х характеристик	0,25

Содействующая рассматриваемая протяженность использования в случае разработки программного обеспечения стандартных модулей (K_T). Протяженность использования в развигается НА стандартных модулях определяется на их удельный вес во всего проектируемого продукта. В этом проектном градусе возможности реализуемых функционирует развивается на стандартных модулях, стандартные программы от 20% - 40%, поэтому $k = 0,8$.

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО (K_H) определяется на основе данных представленных в таблице 4.3 и составляет 1,0.

Таблица 4.3 - Поправочные коэффициенты, учитывающие новизну ПО(K_H).

Категория новизны	Степень новизны	Использование		Значение K_H
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,85
		-	+	1,65
		+	-	1,12
		-	-	1,0
Б	ПО, являющиеся развитием определенного	+	+	1,0
		-	-	0,8

	параметрического ряда ПО	+	-	0,8
В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации	-	-	0,9

Нормативная трудоемкость ПО (T_H) определяется на основании принятого во внимание V_y и категории сложности, которая указана принимая во внимание сложность и новизны проекта и степени использования стандартных модулей при разработке.

В соответствии с этим, согласно укрупненным нормам времени на разработку ПО (T_H) в зависимости от уточненного объема ПО (V_0) и группы сложности: объем ПО (строки исходного кода, LOC) 9465, категория сложности ПО 2-я $T_H = 273$, категория сложности ПО 40.

Следовательно T_0 будет равно

$$T_0 = 273 \cdot 0.17 \cdot 0.8 \cdot 1 = 37,13 \text{ (чел./дн.)}$$

Численность исполнителей проекта ($Ч_p$) рассчитывается по формуле

$$Ч_p = \frac{T_0}{T_p \cdot \Phi_{эф}} \quad (4.5)$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

T_0 – общая трудоемкость разработки проекта (чел./дн.);

T_p – срок разработки проекта (лет).

Срок разработки проекта (T_p) определяется по формуле

$$T_p = \frac{T_0}{Ч_p \cdot \Phi_{эф}} \quad (4.6)$$

где

$Ч_p$ – плановое число разработчиков.

Эффективный фонд времени работы одного работника ($\Phi_{эф}$) рассчитывается по формуле

$$\Phi_{эф} = D_r - D_{п} - D_{в} - D_{о}, \quad (4.7)$$

где D_r - количество дней в году;
 $D_{п}$ - количество праздничных дней в году;
 $D_{в}$ - количество выходных дней в году;
 D_o - количество дней отпуска.

В соответствии с производственным календарем на 2016 год.:

$$D_r = 366; D_{п} = 17; D_{в} = 104; D_o = 14,$$

$$\Phi_{эф} = 366 - 17 - 104 - 14 = 231 \text{ дней}$$

Плановое число разработчиков $Ч_p = 1$, следовательно по формуле (4.6)

$$T_p = 39,31 / (1 \cdot 231) = 0,17 \text{ года} = 60 \text{ дней.}$$

Таким образом, согласно приведенным расчетам и в соответствии с формулой (4.5)

$$Ч = 37,13 / (0,17 \cdot 231) = 1 \text{ чел.}$$

Основные заработные платы рабочих на конкретное ПО рассчитывается по формуле

$$З_{oi} = \sum_{i=1}^n T_{чи} \times T_{ч} \times \Phi_{п} \times K \quad (4.8)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$T_{чи}$ – часовая тарифная ставка i -го исполнителя (тенге);

$\Phi_{п}$ – плановый фонд рабочего времени i -го исполнителя (дней), 60 дня;

$T_{ч}$ – количество часов работы в день (час), 8 часов;

K – коэффициент премирования, составляет 1,2.

Данными о специфических особенностях и сложность выполняемых функционирует штатной список группы специалистов - подрядчики, участвующие в разработке программного обеспечения с определением образные особенности, квалификация и позиция в таблице 4.4 назначена.

Таблица 4.4 - Сведения по работникам, задействованным в проекте

Специалист – Исполнитель	Количество, человек	Заработная плата в месяц, тенге
--------------------------	---------------------	---------------------------------

Программист	1	65 000
Итого		65 000

Тарифная ставка часа выплачивается полностью разделением ежемесячной тарифной ставки на установленном в 40 - к норме недели часа рабочего времени (Пятница), рассматривающего норму ежемесячника средней величины поселения рабочего времени в часах, что через месяц 21 рабочий день

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_{\text{р}}}, \quad (4.9)$$

где $T_{\text{ч}}$ – часовая тарифная ставка (тенге);

$T_{\text{м}}$ – месячная тарифная ставка (тенге).

Таким образом

$$\Phi_{\text{р}} = 8 \cdot 21 = 168 \text{ часов}$$

$$T_{\text{ч}} = \frac{65\,000}{21} = 3\,095,23 \text{ тенге/день}$$

$$T_{\text{м}} = \frac{65\,000}{21} \cdot 60 = 185\,713,8 \text{ тенге/мес}$$

По формуле (4.8) основная заработная плата исполнителей составляет

$$З_{\text{oi}} = 185\,713,8 \cdot 1,2 = 222\,856,56 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$З_{\text{сзи}} = (\text{ФОТ} - \text{ПО}) \cdot 11\% \quad (4.11)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются

$$\text{ПО} = \text{ФОТ} \cdot 10\% \quad (4.12)$$

Таким образом

$$ПО = 222\,856,56 \cdot 0,1 = 22\,285,65 \text{ тенге}$$

$$З_{сзи} = (222\,856,56 - 41\,142,85) \cdot 0,11 = 19\,988,5 \text{ тенге}$$

Затрат на материалы определяются по формуле

$$M_i = (З_{осн} \cdot Н_{мз}) / 100\% \quad (4.13)$$

где $Н_{мз}$ – норма расхода материалов от основной заработной платы (3–5%). Применяем 5%.

$$M_i = 222\,856,56 \cdot 0,05 = 11\,142,82 \text{ тенге}$$

Амортизационные начисления делаются на установленных нормах амортизации, выразил, как процент к балансовой стоимости оборудования и вычисляются на формуле

$$A = \frac{C_{обор} \cdot H_a \cdot N}{366 \cdot 100} \quad (4.14)$$

где H_a – норма амортизации (25 %);

$C_{обор}$ – первоначальная стоимость оборудования;

N – фактический срок эксплуатации оборудования, 60 дней;

Данные по стоимости оборудования представлены в таблице 4.5.

Таблица 4.5 - Стоимость оборудования одного ПК с периферией

Наименование	Цена без НДС, тенге
Монитор	30 470
Материнская плата	15 250
Процессор	44 400
Видеокарта	25 350
HDD	10 560
DVD–RW	6 720
CPU Cooler	5 840
Оперативная память	9 040
Клавиатура	2 000
Мышь	1 500
Принтер	15 040

ПО	35 500
Итого:	201 670

$$C_{обор.} = 201\,670 \text{ тенге}$$

Тогда, согласно формуле (4.14)

$$A = \frac{201\,670 \cdot 25 \cdot 60}{366 \cdot 100} = 8\,265,16 \text{ тенге}$$

Расходы по статье «Прочие затраты» (P_{zi}) на специфике ПО включают закупочные затраты и приготовление специальной научной и технической информации и специальной литературы. Определяются стандартной нормой развилась вообще на организации, как процент к основе оплачивают

$$P_{zi} = Z_{oi} \cdot H_{пз} / 100, \quad (4.16)$$

где $H_{пз}$ – норматив прочих затрат в целом по организации (20 %),
Согласно формуле (4.16)

$$P_{zi} = 222\,856,56 \cdot 0,2 = 44\,571,31 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), соединились с необходимостью содержания персонала управления, вспомогательных ферм и экспериментальных (экспериментальный) производств, а также с расходами на общей необходимости экономики (P_{ni}), принадлежат специфике ПО согласно стандартной норме ($H_{рн}$) как процент основы оплачивают подрядчиков. Стандартная норма устанавливается вообще на организации

$$P_{ni} = Z_{oi} \cdot H_{рн} / 100\%, \quad (4.17)$$

где P_{ni} – накладные расходы на конкретную ПО (тенге);
 $H_{рн}$ – норматив накладных расходов по организации (70 %)
Согласно формуле (4.17)

$$P_{ni} = 222\,856,56 \cdot 0,7 = 155\,999,59 \text{ тенге}$$

В соответствии с формулой (4.1) расчет полных затрат на разработку прикладного приложения составит

$$C_{\text{пi}} = 222\,856,56 + 19\,988,5 + 11\,142,28 + 8\,265,16 + 44\,751,31 + 155\,999,91 = 463\,003,72 \text{ тенге}$$

Сводные результаты расчета затрат на разработку ПО и их структура представлены в таблице 4.6 и на рисунке 4.1.

Таблица 4.6 - Затраты на разработку базы данных учет компьютеров и комплектующих

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{ФОТ}}$	222 856,56	48,13
Социальный налог	$Z_{\text{сзи}}$	19 988,5	4,32
Материалы	M_i	11 142,28	2,41
Амортизация	A	8 265,16	1,78
Прочие затраты	$P_{\text{зи}}$	44 751,31	9,66
Накладные расходы	$P_{\text{ни}}$	155 999,91	33,7
Итого:		463 003,72	100,00



Рисунок 4.1 - Структура затрат на разработку базы данных учета компьютеров и комплектующих

4.2 Расчет цены программного продукта

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$C_{ПП} = Z_{РПР} + П_{п} + НДС \quad (4.18)$$

где $C_{ПП}$ – цена программного продукта, тенге;

$Z_{РПР}$ – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_{п}$ – планируемая прибыль, тенге;

$НДС$ – налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле

$$П_{п} = Z_{РПР} \cdot R_{НПП} \quad (4.19)$$

где $R_{НПП}$ – нормативная рентабельность ПП, определяемая организацией (20%).

НДС, начисленный на ПП, определяется следующим образом

$$НДС = (Z_{РПР} + П_{п}) \cdot k_{НДС} \quad (4.20)$$

где $k_{НДС}$ – ставка налога на добавленную стоимость.

Подставляем все значения в формулы (4.18) – (4.20) и получаем

$$П_{п} = 463\,003,72 \cdot 0,2 = 92\,600,74 \text{ тенге}$$

Подставив данные в формуле (4.20) получаем

$$НДС = (463\,003,72 + 92\,600,74) \cdot 0,12 = 66\,672,53 \text{ тенге}$$

Подставив данные в формуле (4.18) получаем

$$C_{ПП} = 463\,003,72 + 92\,600,74 + 66\,672,53 = 622\,276,99 \text{ тенге}$$

5 Безопасность жизнедеятельности

5.1 Анализ условий труда

Для рабочих удобных рабочих состояний комнаты, которые поддерживают высокую эффективность, не разрушительное здоровье особы требуются. В рабочей комнате согласно деятельности, есть много специализирующегося вычислительного оборудования. В этой комнате два рабочих места, которые соответствуют плану, конкретизировали в фигуре 5.1.

В течение длинной работы на компьютерной относительной влажности воздушный воздух уменьшений и температура возрастает, содержимое в воздухе углекислоты и органических субстанций возрастает. Содержимое в воздухе указанных субстанций может чрезвычайно превышать несколько раз допустимый размер. Поэтому необходимо поддерживать следующие оптимальные параметры микроклимата : воздушный температурный воздух придется вешение с 18 до 21 °С; относительная влажность воздуха соответственно 55 - 62%; скорость воздушного движения менее чем 0,1 м/с. Это необходимо к воздушным комнатам, в зависимости от погодных условий, продолжительности придется не быть меньше, чем 10 мин. Электрическая техника один из источников возникновения пожара. Ноутбуки и другие электрические приборы практически не производят шума и не препятствуют работе.

Длинное действие над записной книжкой, отрицательно влияет на здоровье особы с тех пор, как излучает электромагнитные волны.

Источник искусственной системы использований освещения с покрывающими освещенными с освещенными LED однородно разместил на потолке. Рабочие столы размещены таким образом, что, чтобы не осветить на мониторе проектирования естественного освещения, а именно солнечные лучи.

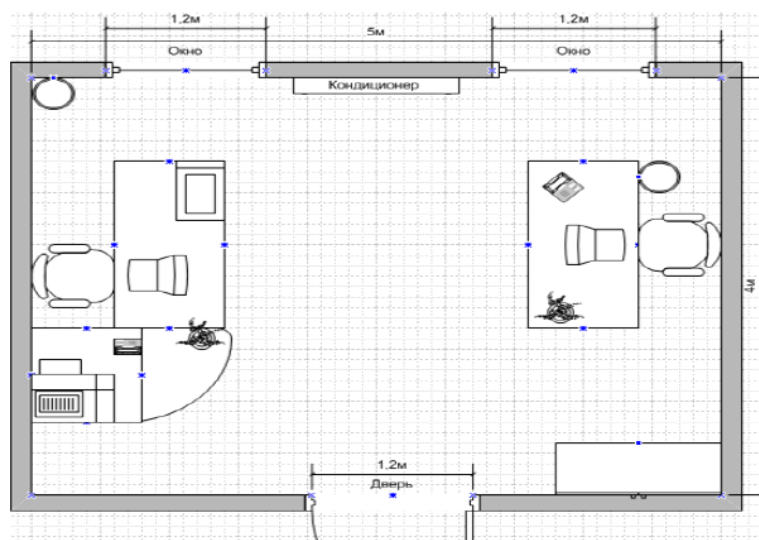


Рисунок 5.1 - План помещения

Согласно ГОСТ 12.1.005–88 «ССБТ. Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ», работа людей в

помещении относится к работе лёгкой тяжести (1а), так как управление оборудованием осуществляется дистанционно с помощью компьютеров.

С целью создания нормальных условий труда оператора установлены нормы производственного микроклимата. При работе с ПК должны соблюдаться следующие условия:

В холодный период:

- оптимальная температура 22 - 24 °С , допустимая – 18 - 26 °С;
- относительная влажность воздуха 40 - 60 %, допустимая 75 %.

В теплый период:

- оптимальная температура 23 - 25 °С, допустимая 20 - 30 °С;
- относительная влажность воздуха 40 - 60 %, допустимая влажность 55 %.

Помещение имеет следующие параметры:

- помещение расположено на втором этаже пятиэтажного здания;
- размеры помещения: длина 5 м, ширина 4 м, высота 4 м;
- площадь: 20 м²;
- светопропускающий материал: стекло листовое, двойное;
- солнцезащитные устройства: убирающиеся регулируемые жалюзи;
- два окна размером 1,5 × 1,2;
- внутренняя отделка стен - светлая;
- помещение по зрительным условиям работы относится к категории легких работ (легкая физическая, категория Ia, работа производится сидя и не требует физического напряжения);
- искусственное освещение – 2 светильника с двумя люминесцентными лампами.

Характеристики используемого в работе оборудования:

- Intel Core i3-380M, 2,5 GHz, 3 Gb RAM / HDD 512 Gb;
- диагональ (дюймы): 15.6;
- электропитание: переменное напряжение 220 В, частотой 50 Гц. Мощность 400 Вт, мощность светильника 2 × 28 Вт;
- 2 светильника, 2 люминесцентные лампы.

5.2 Расчет системы кондиционирования кабинета

Оптимальные правила параметров микроклимата принимая во внимание период года согласно для легкой физической работы даются в таблице 5.1. Оборудование установило в рабочей комнате - не источник размещения жары (очень незначительное размещение жары оборудование никоим образом не напрягает воздействие на микроклимате рабочей комнаты).

Климатические условия действия оборудования полностью соответствуют климатическим условиям, нормализованным для рабочего персонала. Работа людей в помещении относится к работе легкой тяжести (1а), так как управление оборудованием осуществляется дистанционно с помощью компьютеров.

С целью создания нормальных условий, ибо работники правил агентств телекоммуникаций производственного микроклимата установлены. В комнатах в течение работы с ПК климатические условия следования (таблица 5.1) должны быть уступлены.

Таблица 5.1 - Оптимальные нормы температуры, относительной влажности и скорости движения воздуха в обслуживаемой зоне населенного, государственные и административные и семейные должности

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22–24	40–60	0,1
Теплый	Легкая – 1а	23–25	40–60	0,1

Для вентиляции офисной комнаты каналы естественной вентиляции лежали в случае конструкции здания и открывают окна в лето использованы. В течение теплого периода года в случае достижения температуры в офисе над правилами, данными в таблице 5.2 для обслуживания оптимального микроклимата кондиционер использован. Нормальный микроклимат в офисе обеспечивает хорошее здоровье работников во все времена года, и соответственно производительность работы возрастает. Поэтому, для обслуживания условий микроклимата в помещении, рассудительно оборудовать это центральным воздухом.

Вычисление центрального воздуха дается в рабочей комнате ниже. Создание условий обеспечит согласие климата в рабочей комнате к стандартным нормам. Количество шьющегося воздуха, L_{np} (пр) м³/ч, который мы определяем формулой

$$L_{np} = \frac{Q_{изб}}{c \times \rho_{np} \times (t_{вып} - t_{np})}, \quad (5.1)$$

где $Q_{изб}$ - избыточное выделение явной теплоты, кДж/ч;
 c - удельная теплоемкость воздуха при постоянном давлении, равная $c=1$ кДж/кг × оС;
 ρ_{np} - плотность поступающего в помещение воздуха, равная 1,2 кг/м³;

$t_{вып}$ - температура удаляемого из помещения воздуха за пределы рабочей или обслуживаемой зоны, оС;

t_{np} - температура приточного воздуха, оС.

Температура удаляемого из помещения воздуха $t_{вып}$, оС, определяется по формуле

$$t_{\text{выт}} = t_{\text{рз}} + \Delta t \times (h_{\text{вп}} - z), \quad (5.2)$$

где $t_{\text{рз}}$ - температура в рабочей зоне, которая не должна превышать допустимую по нормам ($t_{\text{рз}} \leq t_{\text{доп}}$), оС;

$h_{\text{вп}}$ - расстояние от пола до центра вытяжных проемов (кондиционера), м;

H - высота рабочей зоны, м.

Поскольку расчет производится для теплого периода года, то примем $t_{\text{рз}} = 23$ оС. Внутренняя часть кондиционера расположена на высоте $h_{\text{вп}} = 2.6$ м.

Температура приточного воздуха $t_{\text{пр}}$ при наличии избытка явной теплоты должна быть на 5 - 7 оС ниже температуры воздуха в рабочей зоне $t_{\text{пр}} = 23 - 7 = 16$ оС

Величину избыточного выделения явной теплоты $Q_{\text{изб}}$ находят на основании баланса теплоты в помещении по формуле

$$Q_{\text{изб}} = \sum Q - \sum Q_{\text{ух}}, \quad (5.3)$$

где $\sum Q$ - суммарное количество поступающей в помещение явной теплоты;

$\sum Q_{\text{ух}}$ - суммарное количество уходящей из помещения теплоты (за счет теплопотерь ограждениями, нагрева поступающего в помещение воздуха и т. п.).

Главные источники дополнительной жары - освещения, люди, и т.п., необходимо рассматривать тепlopоступления от солнечной радиации. В этой комнате тепловая эмиссия электронной аппаратуры может быть заброшена. Поэтому мы рассматриваем тепловые эмиссии от искусственного освещения, от людей, количество жары, прибывающей в комнату через окна от солнечной радиации.

Тепловые эмиссии от искусственного освещения Q_2 , граф, предполагая, что практически вся проводимая энергия, в конечном счете, будет преобразована для нагревания, на формуле

$$Q_2 = 1000 \times N, \quad (5.4)$$

где N - расходуемая мощность светильников кВт;

$$Q_2 = 1000 \times 0,28 \times 2 = 560 \text{ кВт}$$

Тепловыделения от людей Q_3 определяют по формуле

$$Q_3 = n \times qч, \quad (5.5)$$

где n - число работающих;

$qч$ - количество тепла, выделяемое одним человеком, представлено в таблице 5.2.

Таблица 5.2 - Количество тепла, выделяемое одним человеком в зависимости от категории работ и температуры окружающей среды

Категория работ	Тепло, Вт			
	Полное		Явное	
	при 100 °С	при 350 °С	при 100 °С	при 350 °С
Легкая	180 °С	145 °С	150 °С	5 °С

$$Q_3 = 2 \times 145 = 290 \text{ кВт}$$

Количество тепла, поступающего в помещение от солнечной радиации $Q_{\text{ост.рад}}$, определяют по формуле

$$Q_{\text{ост.рад}} = F_{\text{ост}} \times q_{\text{ост}} \times A_{\text{ост}} \quad (5.6)$$

для покрытий

$$Q_{\text{п.рад}} = F_{\text{п}} \times q_{\text{п}} \times k_{\text{п}}, \quad (5.7)$$

где $F_{\text{ост}}$ и $F_{\text{п}}$ – площадь поверхности и покрытия, м² ;

$q_{\text{ост}}$ и $q_{\text{п}}$ – теплопоступления через 1 м² поверхности остекления и поверхности покрытия, при коэффициенте теплопередачи, равном Вт/м² × оС;

$A_{\text{ост}}$ - коэффициент остекления;

$k_{\text{п}}$ - коэффициент теплопередачи покрытия, 1 Вт/м² × оС.

Значение $q_{\text{ост}}$ в зависимости от географической ориентации поверхности и характеристики окон или фонарей принимается в пределах 70 - 210, а коэффициент $A_{\text{ост}}$ в зависимости от вида остекления и его солнцезащитных свойств – в пределах 0,25 – 1,25 средние значения теплопоступления от солнечной радиации через покрытие в зависимости от географической широты и вида покрытия принимают в пределах 6 – 24

$$F_{\text{ост}} = 1,5 \times 1,2 \times 2 = 3,6 \text{ м}^2$$

Окно рабочего помещения направлено на север, поэтому примем значение $q_{\text{ост}}$ равным 140 Вт/м² × оС. Примем $A_{\text{ост}} = 0,35$

$$Q_{\text{ост.рад}} = 3,6 \times 140 \times 0,35 = 176,4 \text{ Вт}$$

Среднее значение теплопоступления для покрытия с учетом географической широты примем равным $Q_{п.рад.} = 18 \text{ Вт}$.

Потери тепла из помещения $Q_{ух}$, кВт через стены двери, окна оценивают ориентировочно по формуле

$$Q_{ух} = \frac{\lambda \times S \times (t_{вн.г} - t_{пр.})}{\delta}, \quad (5.8)$$

где λ - теплопроводность стен, Вт/м \times оС ;

S - площадь, м²;

δ - толщина стен, м.

Стены рабочего помещения изготовлены из тяжелого бетона М600, теплопроводность которого равна 12 Вт/(м \times °С). Толщина стен $\delta=0,5$

$$Q_{ух} = \frac{1,2 \times 15 \times (21,4 - 15)}{0,5} = 230,4 \text{ Вт}$$

Вычислим суммарное количество поступающей в помещение явной теплоты

$$\Sigma Q = Q_2 + Q_3 + Q_{ост.рад} + Q_{п.рад} \quad (5.9)$$

$$\Sigma Q = 560000 + 290000 + 176,4 + 18 = 850,2 \text{ кВт.}$$

Так как расчет производится для летнего периода, величина избыточного выделения явной теплоты равна $Q_{изб} = 850,2 \text{ кВт}$

Вычислим количество приточного воздуха

$$L_{пр} = \frac{850,2}{1 \times 1,2 \times (21,4 - 15)} = 110,7 \text{ м}^3/\text{ч}$$

Чтобы обеспечивать расход воздуха $L = 110,7 \text{ м}^3/\text{ч}$, можно использовать 1 кондиционер фирмы LG серии G 07 LHT с рекомендуемой площадью помещения 20 м².

Краткие характеристики модели LG G 07 LHT:

- конструкция: сплит-система;
- основной режим: обогрев, охлаждение, рециркуляция;
- рекомендуемая площадь помещения: 20 м²;
- потребляемая мощность: 0,68 кВт;
- мощность в режиме охлаждения: 2,05 кВт;
- мощность в режиме обогрева: 2,14 кВт;
- расход воздуха (min-max), м³/ч – 100 – 165;
- уровень шума внутреннего блока: 31 дБ;

- уровень шума наружного блока: 46 дБ;
- вес внутреннего блока: 6,2 кг;
- вес внешнего блока: 25 кг;
- габариты внутреннего блока: 26,2 × 80,2 × 16,5 см;
- габариты внешнего блока: 54 × 57,5 × 26,2 см.

Данные характеристики является достаточными для обеспечения комфортного микроклимата в рабочем помещении.

5.3 Расчет естественного освещения кабинета

Естественное освещение не обеспечивает в течение всего рабочего времени необходимого освещения, должен к изменению первого часа дня, или изменения погоды поэтому система освещения состоит из освещений с люминесцентными освещением обеспечена. Стандарты на легких источниках обеспечены в таблице 5.3.

Необходимое освещения для нормального выполнения работы в этой комнате: 400 lx. Для этой цели 4 люминесцентных освещения белого цвета в помещении использованы. Этот выбор вызывает факт, что люминесцентные освещение более экономические, чем обычные лампы накаливания. Так как освещение будут использованы в рабочей комнате, люминесцентные освещение белого цвета Ld- 40 были выбраны.

Здание принадлежит я градус огнестойкости. (Здания с отношением и защита проектирует от естественных или искусственных материалов, конкретный или армированный бетон с использованием листовых невоспламеняющихся материалов). Рабочая комната по поводу безопасности огня принадлежит к классовому "D". Согласно стандартным административным зданиям правил безопасности огня и определенным комнатам, и технологические установки обеспечены с аварийно-спасательным firefighting оборудованием согласно стандартам.

Таблица 5.3 - Рекомендуемые источники света при системе общего освещения

Характеристика зрительной работы по требованию к цветоразличию	Характеристика зрительной работы по требованию к цветоразличию	Характеристика зрительной работы по требованию к цветоразличию	Характеристика зрительной работы по требованию к цветоразличию
Различие	300, 400	3500 – 5500	ЛД, ЛДЦ, ЛБ,

цветных объектов при невысоких требованиях к цветоразличию	150, 200	3000 – 4500	ЛБ, (ЛХБ), НЛВД+МТЛ, ДРЛ
--	----------	-------------	--------------------------

Рассчитаем площадь боковых световых проемов помещения, необходимую для создания нормируемой освещенности на рабочем месте.

Помещение имеет размеры: длина $L = 5$ м, ширина $B = 3$ м, высота $H = 4$ м. Высота рабочей поверхности над уровнем пола h_{rp} , $h_{rp} = 0,725$ м, окна начинается с высоты $h_{но}$, $h_{но} = 0,8$ м, высота окна h_o , $h_o = 1,5$ м. Рабочее помещение находится в IV часовом поясе – в г. Алматы (пояс светового климата – IV 50 оС северной широты и южнее (Алматы, Караганда)).

Рабочее место расположено в $1_{рт}, 1_{рт} = 0,5$ м от наружной стены помещения, где проектируем оконные проемы. Минимальная освещенность будет в точке, отстоящей на расстояние 4 м от оконного проема.

Общую площадь окон S_0 , м², определим по формуле

$$100 \times \frac{S_0}{S_n} = \frac{e_n \times \eta_0}{\tau_0 \times r_1} \times k_{зд} \times k_з, \quad (5.10)$$

$$S_0 = \frac{S_n \times e_n \times \eta_0}{100 \times \tau_0 \times r_1} \times k_{зд} \times k_з \quad (5.11)$$

где S_n – площадь помещения м²;

$S_n = 20$ м²;

e_n – нормированное значение КЕО, выбираемое по таблице 5.3;

mN – коэффициент светового климата.

Учитывая заданный световой пояс, приняв ориентацию световых проемов на Север, определим $mN = 0,9$

$$e_n^{IV} = e_n \cdot m \cdot c, \quad (5.12)$$

где $m = 0,7$;

$c = 0,7$ (в наружных стенах зданий);

e_n - для работ высокой точности III разряда зрительной работы

$$e_n^{IV} = 1,2 \times 0,7 \times 0,75 = 0,63$$

Учитывая тип помещения, найдем коэффициент запаса, если $k_з = 1,2$ (учебные помещения, лаборатории, конструкторские бюро);

τ_0 – общий коэффициент светопропускания равный; $\tau = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4$

$\tau_1 = 0,5$ (пустотелые стеклянные блоки);

$\tau_2 = 0,6$ (деревянные двойные отдельные переплеты);

$\tau_3 = 0,8$ (железобетонные фермы и арки);

$\tau = 1$ (убирающиеся регулируемые жалюзи и шторы);

$$\tau_0 = 0,5 \times 0,6 \times 0,8 \times 1 = 0,24$$

η_0 - световая характеристика окон.

Определяем η_0

$$l = 4 - 1 = 3 \text{ м}$$

Рассчитываем расчетную высоту по формуле

$$h_{\text{расч}} = h_{\text{но}} + h_0 - h_{\text{рп}}, \quad (5.13)$$

$$h_{\text{расч}} = 0,8 + 1,5 - 0,72 = 1,57,$$

$$\frac{B}{h_{\text{расч}}} = \frac{4}{2,3} = 1,7$$

r_1 - коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию.

Средний коэффициент отражения в помещении, принимаем одностороннее боковое освещение $\rho_{\text{ср}} = 0,5$

$$\frac{1_{\text{рт}}}{B} = \frac{0,5}{4} = 0,125$$

где $r_1 = 1,05$

$k_{\text{зд}}$ - коэффициент, учитывающий затенение окон противостоящими зданиями.

Поскольку затеняющих зданий поблизости нет, то $k_{\text{зд}} = 1$.

Вычислим общую площадь окон

$$S_0 = \frac{20 \times 1,35 \times 10 \times 1 \times 0,63}{100 \times 0,24 \times 1,05} = 6,75 \text{ м}^2$$

Площадь световых проемов равна $S_{\text{сп}} = 6,75 \text{ м}^2$.

Таким образом, площадь световых проемов ($1,5 \times 1,2 \times 2 = 3,6 < 6,75$) не обеспечивает необходимых условий труда на рабочем месте. С целью создания наиболее благоприятных условий труда в помещении с параметрами $6 \times 4 \times 3$ для обеспечения нормированного значения КЕО, еN

= 0,84 при III характеристике зрительных работ, совместно с естественным освещением используется искусственное освещение.

5.4 Расчёт искусственного освещения

В течение всего рабочего дневного естественного освещения, должен к изменению погодных условий и первому часу дня, не может обеспечить рабочее место необходимыми условиями, есть потребность в искусственной системе освещения состоит из освещений с светодиодного освещения.

Преимущества светодиодного освещения:

- к призматическому диффузору ослепительный эффект исключен;
- универсальный кожух освещающего обеспечивает различный выбор установки;
- высокоэффективный;
- защита от сверхтоков, к напряжению, температура;
- встроенный корректор мощности;
- пульсация светлого течения менее чем 1%;
- защита против высоковольтного пульса на сети.

Разряд зрительной работы – V. Нормируемая освещенность 2 – 400 лк.

В качестве светильника возьмем ЛСП64–2. Длина светильника 1540 мм, ширина 276 мм (таблица 5.4).

Таблица 5.4 - Технические характеристики газоразрядных ламп ЛБ

Номинальная мощность, Вт	Номинальный световой поток ламп типа ЛБ, лм	Размеры ламп, мм	
		Диаметр	Длина по штырькам
65	3570	40	1514,2

Коэффициенты отражения от потолка стен и пола соответственно равны:

- $r_{пот} = 70 \%$;
- $r_{ст} = 50 \%$;
- $r_{пол} = 30 \%$.

Вычислим высоту подвеса светильника над рабочей поверхностью

$$H = h - h_p - h_c, \quad (5.16)$$

где h_c - расстояние от светильника до перекрытия, $h_c = 0,05$ м;
 h_p - высота рабочей поверхности над полом, $h_p = 0,7$ м;
 h - высота помещения, $h = 4$ м.

$$H = 4 - 0,7 - 0,05 = 3,25 \text{ м.}$$

Лучшее расстояние от окна до светильника определяется по формуле

$$L = \lambda \times H, \quad (5.17)$$

где $\lambda = 1,2 \div 1,4$.

$$L = 1,25 \times 3,25 = 4,0625 \text{ м}$$

Расстояние от стены до ближайшего светильника, когда работа у стены не проводится, определяем по формуле

$$l_1 = (0,4 \div 0,5) \times L, \quad (5.18)$$

$$l_1 \frac{0,4}{0,5} \times 4,0625 = 3,25$$

Определяем индекс помещения по формуле

$$i = \frac{1 \times s}{H \times (1 + S)}, \quad (5.19)$$

$$i = \frac{4 \times 2}{3,25 \times (4 + 2)} = 0,41$$

Коэффициент использования в данном случае равен $\eta = 65 \%$, коэффициент запаса равен $kz = 1,2$

Определим количество люминесцентных ламп по формуле

$$N = \frac{E \times kz \times S_{oc} \times Z}{n \times \Phi_{л} \times \eta}, \quad (5.20)$$

где S_{oc} – площадь помещения;

kz – коэффициент запаса;

E – заданная минимальная освещённость, $E = 400$ лк.;

Z – коэффициент неравномерности освещения, $Z = 1,1$;

n – количество ламп в светильнике;

$\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л} = 3570$ лм;

η – коэффициент использования, $\eta = 65 \%$.

$$N = \frac{400 \times 1,2 \times 20 \times 1,1}{2 \times 3570 \times 0,65} = 2$$

Всего для создания нормируемой освещенности 400 лк необходимо 2 люминесцентных лампы серии ЛД, мощность каждой лампы должна быть не меньше 65 Вт, что соответствует действительности, а значит имеющегося в наличии освещения достаточно для соответствия санитарным нормам, представленные на рисунке 5.2.

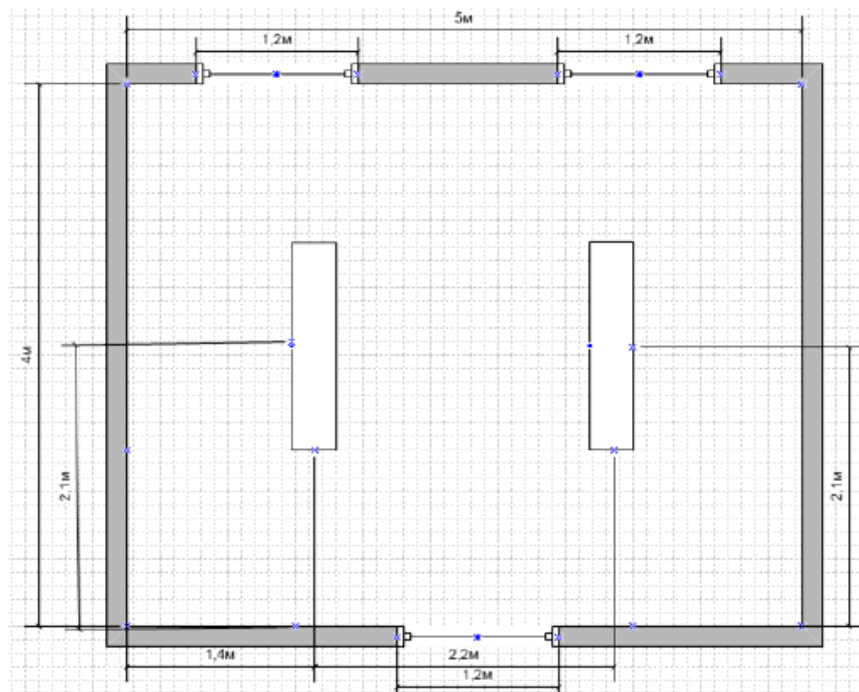


Рисунок 5.2 - Схема расположения светильников

ЗАКЛЮЧЕНИЕ

Главный результат проведенной работы действия - создание действующей системы управления базами данных, которая осуществляет требуемый круг задач, которые работники колледжа на руководстве бухгалтерской дачи компьютеров и аксессуаров отчет о целом периоде учебного лица.

Выполнение этого проекта осуществлялось без аттракциона мощных инструментов действия с базами данных, которые очень громоздки, поскольку имеют универсальный характер и кроме требуют необходимую основу знания согласно теории базы данных.

Использование мощных инструментов Delphi 7.0 на прикладном создании приложений баз данных, работающих в Windows, действующий с системой и в частности, позволил создать программный продукт, который наиболее ориентируется на конечного пользователя, кто не соблазняется в вопросах теории базы данных.

Сейчас, программируя очень развивается, как с точки зрения расширения круга задач, решенных близко это, и с точки зрения существенного осложнения использовал в программировании технологий И особенно необходимо отметить, что значительные размеры - развиваемые программные продукты. Все это требует максимального упрощения и ускорения процесса прикладного программирования и использования раньше осознанных программных фрагментов.

Этот градус развивается на Delphi. Delphi - объектно-ориентированная окружающая среда для визуального дизайна приложений Windows с развиваемыми механизмами многократного использования программного кода. Главный конкурент Delphi - среда разработки Microsoft визуального C++, имеющий преимущества и недостатки, однако, будучи популярнее, в общем по причине факта, что это развивает Microsoft линия Сущности фирмы Delphi, составляющая модель развития программных продуктов есть. Сущность модели заключается в подтверждение системой постоянно расширенного набора объектных компонентов, которых, программа есть строилась. Компоненты в Delphi простые для использования и развития как результат укрывательства значительной части этой структуры программы, которая есть близко к, взаимодействие к операционной системе.

Поэтому, для создания в Delphi простых программных продуктов не обязательно понимать внутреннюю структуру Windows вообще - приложение получило после развития в Delphi. Это скорее просто, чтобы быть способным работать с некоторыми компонентами доставил вместе с кругом разработчика. В то же время возможно начать работу со среды практически без дружественных отношений, и писавши первого приложения не будет требовать углубления в частности системы. Это продвигается частично удобным интерфейсом круга разработчика, который не перегружен с чрезмерными вопросами к разработчику.

Однако такой подход абсолютно неприемлем для серьезного программирования, и, рано или поздно, необходимо освоить также основы программирования под ОС Windows и серьезно изучать среду разработки Delphi, а также возможности, которые это дает. Кроме, это заключительно, создание качественных программных продуктов требует глубоко понимания составляющей модели.

Все необходимое действие над выполнением методов информационного доступа, сохраненных в базе данных, его модификации, обслуживание базы данных в интеграле смотрят скрыт и пользователь не имеет надобности знать об этом успешно, чтобы решить весь круг происходящих проблем, соединенных с информацией, используют к базе данных. Более того, программный интерфейс как можно больше облегчает действие согласно адресу с базой (вплоть до выбора от предложенного числа выбора) данных. Даже адресация база данных с передовыми запросами осуществляется в таком, смотрят, что структура возвращенных данных видима даже перед его выполнением. Система управления базами данных самостоятельно проверяет записи, которые находятся в базе данных, и делает принуждение базы данных к целочисленному статусу, исключая возможные ошибки. Все шаблонные действия этого сортируют предпринимаются машиной, которая несомненно сохраняет усилия и время конечного пользователя.

Круг квалификации стандартов весьма широкий. Он окружает весь ряд задач, начинаясь от первых компьютеров и образования групп на определенном критерии, перед списанным имуществом, реставрация и ведя архива компьютеров после конца периода обучения. Они решаются в пределах этой системы управления базами данных, с максимальной простотой, удобством и скоростью.

Все функции выполняемая система управления базами данных были тщательно проверил и проверил в процессе разработки и их действие гарантировано.

Этот продукт несомненно может соревноваться с базой данных, существующей в моменте и средстве его контроля.

Список литературы

- 1 Баженова И.Ю. Самоучитель программиста Delphi 7. – М.: КУДИЦ-образ, 2003. – 447 с.
- 2 Рудикова Л.В. Базы данных: разработка приложений, – СПб.: ВНУ Санкт-Петербург, 2007.
- 3 Гусева Т.И. Проектирование баз данных в примерах и задачах. – М.: Просвещение, 1992. – 289 с.
- 4 Сатимова Е.Г. Проектирование баз данных: Методические указания к выполнению лабораторных работ. – Алматы: АИЭС, 2009.
- 5 Еркешева З.Д., Боканова Г.Ш. Техничко-экономическое обоснование: Методические указания к выполнению экономической части дипломных работ. – Алматы: АУЭС, 2013.
- 6 Зубов А. Программирование на Delphi. Трюки. Эффекты. – СПб.: Питер, 2004. – 396 с.
- 7 Тимоти Б. ООП в действии. – СПб., 1997. – 325 с.
- 8 Лодон Дж., Додон К. Управление информационными системами, 7-е издание, – Питер, 2005.
- 9 Плю Р., Стефенс Р., Райан К. Освой самостоятельно программу за 24 часа. – М.: Издательский дом «Вильямс», 2000.
- 10 Роб П., Корнел К. Системы БД: проектирование, реализация. Управление. 5-е издание, – СПб.: ВНУ Санкт-Петербург, 2004.
- 11 Питер Роб, Карлос Корнел. Системы БД: проектирование и реализация. – СПб.: ВНУ Санкт-Петербург, 2004.
- 12 Фатрелл Р., Шафер Д., Шафер Л. Управление программными проектами. Достижение оптимальных качеств при минимальных затратах. - М.: Издательский дом «Вильямс», 2004. – 1136 с.

Приложение А

```
procedure TForm2.Edit2Change(Sender: TObject);
begin
//áñèè â ñèå Edit1 áñòù õìòù íáíà áóéâà,
if Edit2.Text <> " " then begin
DataModule3.TabKT.Filtered := False; //ìòèëþ÷ààì òèèüòð
ed := Edit2.Text; //íàðååååì â DM íââé òàèñò
DataModule3.TabKT.Filtered := True; //âèëþ÷ààì òèèüòð
end
//áñèè áóéâ íàò, òèèüòðàòèëþ ìòèëþ÷ààì:
else DataModule3.TabKT.Filtered := False;

end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Form1.Show;
end;

procedure TForm2.BitBtn2Click(Sender: TObject);
var
XL, Cell1, Cell2, ArrayData: Variant;
    RowCount: integer;
    BeginCol, BeginRow,i,ColCount:byte;
begin
// Êñòðåéàòù èåííâ ãðððíââ òåè àéåñòè, èóåå áóååì âùâñèèòù ääííù
    BeginCol := 1;
    BeginRow := 3;
    RowCount := DataModule3.TabKT.RecordCount; //Êñèè÷åñòòî ÷àèèñå â
òåèèèèè, èìòððð áóååì íàðåååååòù â Excel
    ColCount := 12; //Êñèè÷åñòòî ñòèèéåíâ
    XL := CreateOleObject('Excel.Application'); //ñèçååå Excel
    XL.Application.EnableEvents := false; // Ìòèëþ÷ààì ðååèèè Excel íà ñíâùòèè
(íèçâèèèò òñèèèèèè âùâñè èìòðððèèèè)
    XL.WorkBooks.Add; // Æíââèèèè èèèè
    XL.Rows.Rows[2].Font.Bold:=true; //Æåååå ìèèèèè ðèèèè â èèèèè òåèèèè
    XL.Rows.Rows[2].Font.Size:=10; //Óñòàíåèèèèèè ðàçèèèè ðèèèèè
    XL.Rows.Rows[2].WrapText := True; //Æèëþ÷ààì íàðâñèè òåèèèè â ÿ÷åèèè
    XL.Rows.Rows[2].HorizontalAlignment := 3; //Óñòàíåèèèèèè ãèèèèèèèèèèè
âùðåèèèèèèèè ñèèèèè
    XL.Rows.Rows[2].VerticalAlignment := 2; //Óñòàíåèèèèèèèè ãððèèèèèèèèèèè
âùðåèèèèèèèè ñèèèèè
    XL.Cells[2,1]:= 'Òèè òñòðèèèèèè'; //Óìèèèèèèèèèè èèèèèè òåèèèèèè
```

```

XL.Cells[2,2]:='Òåðíè÷åñêèå ðàðàèòåððèñòèèè';
XL.Cells[2,3]:='Èíåííòàðííúé íííåð';
XL.Cells[2,4]:='Ãîå âúíóñèà';
XL.Cells[2,5]:='Ãîå ïðèíáððòáíèý';
XL.Cells[2,6]:='Ñòèèíñòü';
XL.Cells[2,7]:='Óñòáííåèá â èàáèíàòà ' ';
XL.Cells[2,8]:='Îòååòñòåíííúé';
XL.Cells[2,9]:='Îòååèèèè';
XL.Cells[2,10]:='Ñðèè ãàðáíèèè';
XL.Cells[2,11]:='Òåðíèè÷åñêèå ññòòèýíèè';
XL.Cells[2,12]:=' ' àèòà òñòáííèè';

```

```

ArrayData := VarArrayCreate([1, RowCount, 1, ColCount], varVariant); //
çàåååí ààðèèòííèè ìåññèå, èíòèðíèè áóååí çàññèýòü àáíííèè èç ADOTable1

```

```

DataModule3.TabKT.First;//çàññèýòü ìåññèå çíà÷åíèè èç ADOTable1

```

```

for i := 1 to RowCount do

```

```

begin

```

```

    ArrayData[i,1] := DataModule3.TabKT['Òèè_ óñòðèèñòåå'];

```

```

    ArrayData[i,2] := DataModule3.TabKT['Òåðíèè÷åñêèå ðàðàèòåððèñòèèè'];

```

```

    ArrayData[i,3] := DataModule3.TabKT['Èíåííòàðííúé íííåð'];

```

```

    ArrayData[i,4] := DataModule3.TabKT['Ãîå âúíóñèà'];

```

```

    ArrayData[i,5] := DataModule3.TabKT['Ãîå ïðèíáððòåííèý'];

```

```

    ArrayData[i,6] := DataModule3.TabKT['Ñòèèíñòü'];

```

```

    ArrayData[i,7] := DataModule3.TabKT['Óñòáííåèá â èàáèíàòà ''];

```

```

    ArrayData[i,8] := DataModule3.TabKT['Îòååòñòåíííúé'];

```

```

    ArrayData[i,9] := DataModule3.TabKT['Îòååèèèè'];

```

```

    ArrayData[i,10] := DataModule3.TabKT['Ñðèè ãàðáíèèè'];

```

```

    ArrayData[i,11] := DataModule3.TabKT['Òåðíèè÷åñêèå ññòòèýíèè'];

```

```

    ArrayData[i,12] := DataModule3.TabKT[' ' àèòà òñòáííèè'];

```

```

    DataModule3.TabKT.Next;

```

```

end;

```

```

Cell1 := XL.Cells[BeginRow-1, BeginCol]; // èåååý ååðííýý ÿ÷åèè
íåèåñòè â èíòèðíèè áóååí åúåíåèòü àáíííèè

```

```

Cell2 := XL.Cells[BeginRow+RowCount-1, BeginCol + ColCount-1];//íðåååý
íèèííèè ÿ÷åèè íåèåñòè èóåå áóååí åúåíåèòü àáíííèè

```

```

XL.Range[XL.Cells[BeginRow, BeginCol], Cell2].Value := ArrayData;
//çàññèýòü èíèèè íåèèè àáíííèè

```

```

XL.Range[Cell1, Cell2].Borders.LineStyle := 1;

```

```

XL.Visible := true;//ååèèè Excel àèèèèè

```

```

end;

```

```

procedure TForm2.RadioGroup1Click(Sender: TObject);
begin
case RadioGroup1.ItemIndex of
0: DataModule3.TabKT.Sort:='Òèì_óñòðíéñòàà ASC';
1: DataModule3.TabKT.Sort:='Ãîä_âüíóñèà ASC';
2: DataModule3.TabKT.Sort:='Ãîä_ïðèíáððòáíèÿ ASC';
3: DataModule3.TabKT.Sort:='Ñòìèíñòü ASC';
4: DataModule3.TabKT.Sort:='Îòääëáíèà ASC';
end;
end;
procedure TForm2.BitBtn3Click(Sender: TObject);
begin
if RadioButton7.Checked then
DataModule3.TabKT.Filter:='Ãîä_ïðèíáððòáíèÿ >="
+DateToStr(DateTimePicker1.Date)+" and DataStU <="
+DateToStr(DateTimePicker2.Date)+"";
end;

end.
unit Unit3;

interface

uses
SysUtils, Classes, DB, ADODB;

type
TDataModule3 = class(TDataModule)
ADOConnection1: TADOConnection;
TabKT: TADOTable;
TabPO: TADOTable;
TabKomplekt: TADOTable;
TabPeremeshenia: TADOTable;
TabRemont: TADOTable;
TabSpisania: TADOTable;
DSKT: TDataSource;
DSPO: TDataSource;
DSKomplekt: TDataSource;
DSPeremeshenia: TDataSource;
DSRemont: TDataSource;
DSSpisania: TDataSource;
TabPODSDesigner: TIntegerField;
TabPODSDesigner2: TWideStringField;
TabPODSDesigner3: TIntegerField;

```

TabPODSDesigner4: TDateTimeField;
TabPODSDesigner5: TBCDField;
TabPODSDesigner6: TIntegerField;
TabPODSDesigner7: TIntegerField;
TabKomplektDSDesigner: TIntegerField;
TabKomplektDSDesigner2: TWideStringField;
TabKomplektDSDesigner3: TWideStringField;
TabKomplektDSDesigner4: TDateTimeField;
TabKomplektDSDesigner5: TBCDField;
TabKomplektDSDesigner6: TIntegerField;
TabKomplektDSDesigner7: TWideStringField;
TabKomplektDSDesigner8: TIntegerField;
TabPeremesheniaDSDesigner: TIntegerField;
TabPeremesheniaDSDesigner2: TWideStringField;
TabPeremesheniaDSDesigner3: TDateTimeField;
TabPeremesheniaDSDesigner4: TIntegerField;
TabPeremesheniaDSDesigner5: TIntegerField;
TabPeremesheniaDSDesigner6: TWideStringField;
TabPeremesheniaDSDesigner7: TWideStringField;
TabPeremesheniaDSDesigner8: TWideStringField;
TabPeremesheniaDSDesigner9: TIntegerField;
TabRemontDSDesigner: TIntegerField;
TabRemontDSDesigner2: TWideStringField;
TabRemontDSDesigner3: TDateTimeField;
TabRemontDSDesigner4: TWideStringField;
TabRemontDSDesigner5: TBCDField;
TabRemontDSDesigner6: TIntegerField;
TabRemontDSDesigner7: TIntegerField;
TabRemontDSDesigner8: TIntegerField;
TabSpisaniaDSDesigner: TIntegerField;
TabSpisaniaDSDesigner2: TWideStringField;
TabSpisaniaDSDesigner3: TDateTimeField;
TabSpisaniaDSDesigner4: TDateTimeField;
TabSpisaniaDSDesigner5: TWideStringField;
TabSpisaniaDSDesigner6: TIntegerField;
TabSpisaniaDSDesigner7: TIntegerField;
TabKTDSDesigner: TAutoIncField;
TabKT_: TWideStringField;
TabKTDSDesigner2: TWideStringField;
TabKTDSDesigner3: TIntegerField;
TabKT_2: TDateTimeField;
TabKTDSDesigner4: TBCDField;
TabKTDSDesigner5: TIntegerField;
TabKTDSDesigner6: TWideStringField;


```

TabKTDSDesigner7: TWideStringField;
TabKTDSDesigner8: TIntegerField;
TabKTDSDesigner9: TWideStringField;
TabKTDSDesigner10: TIntegerField;
TabKT_3: TDateTimeField;
procedure TabKTFilterRecord(DataSet: TDataSet; var Accept: Boolean);
private
  { Private declarations }
public
  { Public declarations }
  procedure MyLocate(s: string);

end;

var
  DataModule3: TDataModule3;
  ed: String; //òâñò èç Edit2

implementation

{$R *.dfm}
procedure TDataModule3.MyLocate(s: String);
begin
  TabKT.Locate('Òèì_ónòðîéñòâà', s, [loPartialKey]);

end;

procedure TDataModule3.TabKTFilterRecord(DataSet: TDataSet;
  var Accept: Boolean);
  var
    s : string; //äÿ çíà÷áíèÿ ïÿÿ

begin
  //ïëó÷ààì ñòíèüèì íà÷àèüíüð áóéâ èç ïÿÿ Òèì óñòðîéñòâà,
  //ñèíèüèì áóéâ èìáàðñÿ â ïàðâàíáííé ed:
  s := Copy(DataSet['Òáðíé÷áñèèà ðàðàèòàððèñòèè'], 1, Length(ed));
  //ääèààì ïðíàððèó íà ñíàðâàáíèà çíà÷áíèé:
  Accept := s = ed;
end;

end.

procedure TForm4.GroupBox1Exit(Sender: TObject);
begin

```

```
if DataModule3.TabKT.Modified then  
DataModule3.TabKT.Post;  
end;
```

```
procedure TForm4.GroupBox2Exit(Sender: TObject);  
begin  
if DataModule3.TabPO.Modified then  
DataModule3.TabPO.Post;  
end;
```

```
procedure TForm4.GroupBox3Exit(Sender: TObject);  
begin  
if DataModule3.TabKomplekt.Modified then  
DataModule3.TabKomplekt.Post;  
end;
```

```
procedure TForm4.GroupBox4Exit(Sender: TObject);  
begin  
if DataModule3.TabPeremeshenia.Modified then  
DataModule3.TabPeremeshenia.Post;  
end;
```

```
procedure TForm4.GroupBox5Exit(Sender: TObject);  
begin  
if DataModule3.TabRemont.Modified then  
DataModule3.TabRemont.Post;  
end;
```

```
procedure TForm4.GroupBox6Exit(Sender: TObject);  
begin  
if DataModule3.TabSpisania.Modified then  
DataModule3.TabSpisania.Post;  
end;
```

```
procedure TForm4.BitBtn1Click(Sender: TObject);  
begin  
if DataModule3.TabKT.Modified then  
DataModule3.TabKT.Post;  
if DataModule3.TabPO.Modified then  
DataModule3.TabPO.Post;  
if DataModule3.TabKomplekt.Modified then  
DataModule3.TabKomplekt.Post;  
if DataModule3.TabPeremeshenia.Modified then  
DataModule3.TabPeremeshenia.Post;
```

```
if DataModule3.TabRemont.Modified then
DataModule3.TabRemont.Post;
if DataModule3.TabSpisania.Modified then
DataModule3.TabSpisania.Post;
close;
end;
```

```
procedure TForm4.BitBtn2Click(Sender: TObject);
begin
DataModule3.TabKT.Append;
DataModule3.TabPO.Append;
DataModule3.TabKomplekt.Append;
DataModule3.TabPeremeshenia.Append;
DataModule3.TabRemont.Append;
DataModule3.TabSpisania.Append;
DBEdit1.SetFocus;
end;
```

```
procedure TForm4.BitBtn3Click(Sender: TObject);
begin
if DataModule3.TabKomplekt.Modified then
DataModule3.TabKomplekt.Post;
DataModule3.TabKomplekt.Append;
DBEdit19.SetFocus;
end;
```

```
end.
unit Unit5;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, Buttons, ComCtrls, ComObj,
ExtCtrls,
OleServer, ExcelXP;
```

```
type
```

```
TForm5 = class(TForm)
ADOConnection1: TADOConnection;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
```

```

BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
StatusBar1: TStatusBar;
GroupBox1: TGroupBox;
Label1: TLabel;
RadioGroup1: TRadioGroup;
RadioButton1: TRadioButton;
Edit1: TEdit;
BitBtn6: TBitBtn;
ExcelApplication1: TExcelApplication;
ADOQuery1DSDesigner: TAutoIncField;
ADOQuery1_: TWideStringField;
ADOQuery1_2: TWideStringField;
ADOQuery1_3: TIntegerField;
ADOQuery1_4: TDateTimeField;
ADOQuery1DSDesigner2: TBCDField;
ADOQuery1DSDesigner3: TIntegerField;
ADOQuery1DSDesigner4: TWideStringField;
ADOQuery1DSDesigner5: TWideStringField;
ADOQuery1DSDesigner6: TIntegerField;
DBGrid1: TDBGrid;
RadioButton2: TRadioButton;
DateTimePicker1: TDateTimePicker;
DateTimePicker2: TDateTimePicker;
Label2: TLabel;
Label3: TLabel;
BitBtn5: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);
procedure ADOQuery1FilterRecord(DataSet: TDataSet;
  var Accept: Boolean);
procedure Edit1Change(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure RadioGroup1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }

end;

```

```

var
  Form5: TForm5;
  fs:string;
  ed: String; //ðâêñò èç Edit1
implementation

uses Unit6, Unit1;

{$R *.dfm}

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
  Form6.Show;
  AdoQuery1.Insert
end;

procedure TForm5.BitBtn3Click(Sender: TObject);
begin
  AdoQuery1.Delete;
end;

procedure TForm5.BitBtn2Click(Sender: TObject);
begin
  Form6.Show;
end;

procedure TForm5.BitBtn4Click(Sender: TObject);
begin
  Form5.Hide;
  Form1.Show;
end;

procedure TForm5.DBGrid1KeyPress(Sender: TObject; var Key: Char);
begin
  fs:=fs+key;
  DBGrid1.DataSource.DataSet.Locate('Òèï_óñòðîéñòââ', fs, [loPartialKey]);
  StatusBar1.Panels.Items[0].Text:='Èùài: '+fs;
end;

procedure TForm5.ADOQuery1FilterRecord(DataSet: TDataSet;
  var Accept: Boolean);

```

```

var
s : String; //äëÿ çíà÷áíëÿ îëÿ
begin
//îëó÷ääî ñòîëüêî íà÷äëüíüð áóêâ èç îëÿ Ôàìèèèÿ,
//ñêîëüêî áóêâ èìààòñÿ â ïäðàíáííé ed:
s := Copy(DataSet['Òèì_óñòðîéñòàà'], 1, Length(ed));
//ääèääì ïðíàáðéó íà ñíàíàääáíèà çíà÷áíèé:
Accept := s = ed;
end;

```

```

procedure TForm5.Edit1Change(Sender: TObject);
begin
//ññèè â ïñèá Edit1 ãñòü ðíòü íáíà áóêâà,
if Edit1.Text <> " then begin
ADOQuery1.Filtered := False; //îèèþ÷ääì òèèüòð
ed := Edit1.Text; //íäðàääääì â DM íâúé òàêñò
ADOQuery1.Filtered := True; //âèèþ÷ääì òèèüòð
end
//ññèè áóêâ íàð, òèèüòðàòèþ îèèþ÷ääì:
else ADOQuery1.Filtered := False;
end;

```

```

procedure TForm5.BitBtn6Click(Sender: TObject);
var
XL, Cell1, Cell2, ArrayData: Variant;
RowCount: integer;
BeginCol, BeginRow,i,ColCount:byte;
begin
// Òíðàèíàòü èääíâí àáððíáâí óàèà íáèàñòè, èóàà áóääì àúâíàèòü àáííúâ
BeginCol := 1;
BeginRow := 3;
RowCount := ADOQuery1.RecordCount; //Ëîèè÷ãñòâí çàèèñáé â òàáèèòà,
èîððòþ áóääì ïäðàääääòü â Excel
ColCount := 9; //Ëîèè÷ãñòâí ñòèèáóíâ
XL := CreateOleObject('Excel.Application'); //ñíçääääì Excel
XL.Application.EnableEvents := false; // Ìòèèþ÷ääì ðààèòèþ Excel íà ñíáúòèÿ
(ííçâíèÿò òñèíðèòü àúâíâ èíîððàòèè)
XL.WorkBooks.Add; // Äíáààèÿâí èíèáó
XL.Rows.Rows[2].Font.Bold:=true; //Äãèääì æèðíúé ððèòò â òàèèà òàáèèòü
XL.Rows.Rows[2].Font.Size:=10; //Óñòàíàèèèèääì ðàçíàð ððèòòà
XL.Rows.Rows[2].WrapText := True; //Âèèþ÷ääì ïäðàíí òàêñòà â ÿ÷áèèáð
XL.Rows.Rows[2].HorizontalAlignment := 3; //Óñòàíàèèèèääì ãðèçîâàèííâ
àúðàâíèèèèè ïí òáíòðó

```



```
XL.Range[Cell1, Cell2].Borders.LineStyle := 1;  
XL.Visible := true; //ääää Excel äèèù
```

```
end;
```

```
procedure TForm5.BitBtn5Click(Sender: TObject);
```

```
begin
```

```
if RadioButton2.Checked then
```

```
ADOQuery1.Filter:='Ãä_ïðèáðòòáèÿ
```

```
+DateToStr(DateTimePicker1.Date)+'
```

```
and
```

```
DataStU
```

```
>=""
```

```
<=""
```

```
+DateToStr(DateTimePicker2.Date)+'";
```

```
end;
```

```
procedure TForm5.RadioGroup1Click(Sender: TObject);
```

```
begin
```

```
case RadioGroup1.ItemIndex of
```

```
0: ADOQuery1.Sort:='Òèï_ónòðíéñòää ASC';
```

```
1: ADOQuery1.Sort:='Èíääíòàðíúé_íîäð ASC';
```

```
2: ADOQuery1.Sort:='Ãä_ïðèáðòòáèÿ ASC';
```

```
3: ADOQuery1.Sort:='Ñòèèñòü ASC';
```

```
end;
```

```
end;
```

```
end.
```