Министерство образования и науки РФ

**ГОУ ВПО Череповецкий государственный университет**

Институт инноваций и непрерывного образования

Кафедра Программное обеспечение ЭВМ

Дисциплина Проектирование информационных систем

**КУРСОВАЯ РАБОТА**

Проектирование информационной системы

«Справочник радиолюбителя»

Расчетно-пояснительная записка к курсовой работе

Листов 38

Выполнил студент:

Холодов К. И.

Группа: 6СПО32

Принял преподаватель:

Селяничев О. Л.

Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Череповец

2007г

### Аннотация

Данная курсовая работа предназначена для приобретения навыков проектирования информационных систем. Научиться принципам построения функциональных и информационных моделей систем, проведению анализа полученных результатов, применению инструментальных средств поддержки проектирования информационных систем.

Количество листов……………………………………………………..……38

Количество таблиц…………………………………………………………..14

Количество приложений……………………………………………………..2

**Содержание**

Введение……………………………………………….……………………….4

1. Описание предметной области……………………………………….….....5

2.Выбор ЖЦ ИС………………………………………………………………..6

 2.1 Методология RAD…………………………………………………….....10

3. Этапы ЖЦ…………………………………………………………………...11

 3.1 Фаза анализа и планирования требований……………………………..11

 3.2 Фаза проектирования…………………………...…………………….....11

 3.3 Фаза построения……………………………………………………..…..17

 3.4 Фаза внедрения…………………………………………………………..18

4. Выбор CASE средства……………………………….………………….....19

5. Описание программы………………………………………………………24

Заключение…………………………..……………………..............................25

Литература…………………………………..…………………………….......26

Приложение 1. Руководство пользователя………………………..………...27

Приложение 2. Листинг программы…………………………………………33

**Введение**

В ходе данной работы необходимо спроектировать и реализовать информационную систему “Справочник радиолюбителя”, а также приложение, которое позволит работать с ней.

База данных предназначена для людей работающих в сфере электроники. База должна включать в себя данные о радиодеталях с различными критериями, а так же о их производителе.

В современном мире сложно найти такую сферу деятельности, где бы не использовались базы данных, которые позволяют эффективный и удобный сбор, хранение, обработку и представление информации различного рода. БД «Справочник радиолюбителя» необходима для удобной и быстрой работы с базой радиодеталей, которая бы обладала простой реализацией и структурой. Этими свойствами как раз и обладает реляционная модель базы данных.

В приложении 1 необходимый материал для пользователя программы

В приложении 2 приводится листинг программы.

**1.Описание предметной области**

Данная ИС проектируется как пособие для тех людей, которые увлекаются радиоэлектроникой. Так как база данных включает в себя полный спектр свойств по радиодетали, радиолюбитель всегда сможет узнать интересующую его информацию, к примеру такую как: производитель, материал исполнения, подкласс, мощность, и т.д. И на основе, всего вышеизложенного сделать какие – либо выводы, о том какую марку радиодетали ему стоит приобрести.

2. Выбор модели жизненного цикла информационной системы

Модель жизненного цикла (далее ЖЦ) программного обеспечения информационной системы (далее ПО ИС) при автоматизированном проектировании играет достаточно важную роль. Это обусловлено тем, что каждая из CASE-систем ориентирована на определенную модель ЖЦ ПО ИС.

Жизненным циклом ПО ИС представляет собой непрерывный процесс, начинающийся с момента принятия решения о создании ПО и заканчивающийся при завершении его эксплуатации.

Под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ. Модель ЖЦ зависит от специфики ИС и специфики условий, в которых последняя создается и функционирует.

Состав процессов ЖЦ регламентируется международным стандартом ISO/IEC 12207: 1995 «Information Technologe – Software Life Cycle Processes» («Информационные технологии – Процессы жизненного цикла программного обеспечения»). Стандарт ISO/IEC 12207 не предлагает конкретную модель ЖЦ и методы разработки ПО. Его регламенты являются общими для любых моделей ЖЦ, методологий и технологий разработки. Стандарт ISO/IEC 12207 описывает структуру процессов ЖЦ ПО, но не конкретизирует в деталях, как реализовать или выполнить действия и задачи, включенные в эти процессы.

К настоящему времени наибольшее распространение получили следующие две основные модели ЖЦ:

* каскадная модель (70-85 г.г.);
* спиральная модель (86-90 г.г.).

В изначально существовавших однородных ИС каждое приложение представляло собой единое целое. Для разработки такого типа приложений применялся каскадный способ. Его основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем (рис.1). Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Положительные стороны применения каскадного подхода заключаются в следующем:

* на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
* выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

|  |
| --- |
|  |

Рис.1 «Каскадная модель жизненного цикла»

Каскадный подход хорошо зарекомендовал себя при построении ИС, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования, с тем, чтобы предоставить разработчикам свободу реализовать их как можно лучше с технической точки зрения. В эту категорию попадают сложные расчетные системы, системы реального времени и другие подобные задачи. Однако в процессе использования этого подхода обнаружился ряд его недостатков, вызванных прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывался в такую жесткую схему. В процессе создания ПО постоянно возникала потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. В результате реальный процесс создания ПО принимал следующий вид (рис. 2):

|  |
| --- |
|  |

Рис.2 «Модель жизненного цикла с промежуточным контролем»

Основным недостатком каскадного подхода является существенное запаздывание с получением результатов. Согласование результатов с пользователями производится только в точках, планируемых после завершения каждого этапа работ, требования к ИС "заморожены" в виде технического задания на все время ее создания. Таким образом, пользователи могут внести свои замечания только после того, как работа над системой будет полностью завершена. В случае неточного изложения требований или их изменения в течение длительного периода создания ПО, пользователи получают систему, не удовлетворяющую их потребностям. Модели (как функциональные, так и информационные) автоматизируемого объекта могут устареть одновременно с их утверждением.

Для преодоления перечисленных проблем была предложена спиральная модель ЖЦ (рис. 3), делающая упор на начальные этапы ЖЦ: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем. При итеративном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача – как можно быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований.

Основная проблема спирального цикла – определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

|  |
| --- |
|  |

Рис. 3 «Спиральная модель жизненного цикла»

При проектировании программного обеспечения за основу была взята спиральная модель жизненного цикла (рис. 3), так как при проектировании данной ИС на ранних стадиях можно показать предварительную версию программного обеспечения пользователю, для того чтобы обсудить недостатки, пожелания и замечания клиента и разработчика.

**2.1 Методология RAD**

В основе проекта любой ИС лежат: методологии, технологии и инструментальные средства проектирования.

Методология реализуется через технологии инструментальными средствами.

Для разработки данной информационной системы использовалась методология RAD.

RAD (от англ. rapid application development — быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание быстроте и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. С конца XX века RAD получила широкое распространение и одобрение. Реализует спиральную модель ЖЦ ПО.

ЖЦ ПО по этой методологии состоит из 4-х этапов:

- фаза анализа и планирования требований;

- проектирование;

- построение;

- внедрение.

**3. Этапы ЖЦ ИС**

**3.1 Фаза анализа и планирования требований.**

Приоритетные функции ИС «Справочник радиолюбителя»:

- генерация различных отчетов о имеющихся радиодеталях;

- просмотр и подборка по функциональным возможностям;

- поиск по конкретным параметрам.

**3.2 Фаза проектирования**

**Даталогическое проектирование**

Основная цель проектирования БД – это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти, уменьшение затрат на многократные операции обновления избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте. Так называемый, "чистый" проект БД ("Каждый факт в одном месте") можно создать, используя методологию нормализации отношений. Теория нормализации основана на концепции нормальных форм. Каждой нормальной форме соответствует набор ограничений и, говорят, что отношение соответствует нормальной форме, если оно удовлетворяет этому набору ограничений.

Начнём составление базы данных с нескольких ненормализованных таблиц, в которых заключаются все интересующие атрибуты (поля):

**Ненормализованная таблица**

Таблица 1 «Ненормализованная таблица»

|  |  |  |  |
| --- | --- | --- | --- |
| Марка | Производитель | Полуп материал | Подкласс |
| TXD5100 | Япония, Panasonik | Германий | Биполярный |
| MX300 | США, AMDX | Кремний | Полевой |
| ТВ23У | Россия, Радио деталь | Галий | Полевой |
| ИБ10М | Россия, Наш транзистор | Индий | Биполярный |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функц возможности | Тип перехода | Классификация | Тип сборки | Цена |
| 10, малая, от 50 до 50 | pnp | 2, 3 | Транз сборка | 100 |
| 5, малая, от 50 до 150 | npn | 22, 6 | Безкорпусный | 250 |
| 20, средняя, от 50 до 50 | npn | 88, 1 | Транз сборка | 20 |
| 10, высокая от 50 до 100 | npn | 35, 9 | Безкорпусный | 25 |

**1NF**

Таблица находится в **первой нормальной форме** (1НФ) тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица 2 «1NF»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Марка | Страна | Фирма | Полуп материал | Подкласс |
| TXD5100 | Япония | Panasonik | Германий | Биполярный |
| MX300 | США | AMDX | Кремний | Полевой |
| ТВ23У | Россия | Радио деталь | Галий | Полевой |
| ИБ10М | Россия | Наш транзистор | Индий | Биполярный |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Мощность | Частота | Рабочая темпер | Тип перехода | Номер разработки |
| 10 | Малая | от минус 50 до 50 | pnp | 2 |
| 5 | Малая | от минус 50 до 150 | npn | 22 |
| 20 | Средняя | от минус 50 до 50 | npn | 88 |
| 10 | Высокая | от минус 50 до 100 | npn | 35 |

|  |  |  |
| --- | --- | --- |
| Модернизация | Тип сборки | Цена |
| 3 | Транз сборка | 100 |
| 6 | Безкорпусный | 250 |
| 1 | Транз сборка | 20 |
| 9 | Безкорпусный | 25 |

**2NF**

Таблица находится во **второй нормальной форме** (2НФ), если она удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны **полной функциональной зависимостью** с первичным ключом.

Таблица 3 «производитель»

|  |  |  |  |
| --- | --- | --- | --- |
| Код производителя | Код транзистора | Страна | Фирма |
| 1 | 1 | Япония | Panasonik |
| 2 | 2 | США | AMDX |
| 3 | 3 | Россия | Радио деталь |
| 4 | 4 | Россия | Наш транзистор |

Таблица 4 «Радиодеталь»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код радиодетали | Марка | Полуп материал | Подкласс | Мощность |
| 1 | TXD5100 | Германий | Биполярный | 10 |
| 2 | MX300 | Кремний | Полевой | 5 |
| 3 | ТВ23У | Галий | Полевой | 20 |
| 4 | ИБ10М | Индий | Биполярный | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Частота | Рабочая темпер | Тип перехода | Номер разработки |
| Малая | от минус 50 до 50 | pnp | 2 |
| Малая | от минус 50 до 150 | npn | 22 |
| Средняя | от минус 50 до 50 | npn | 88 |
| Высокая | от минус 50 до 100 | npn | 35 |

|  |  |  |
| --- | --- | --- |
| Модернизация | Тип сборки | Цена |
| 3 | Транз сборка | 100 |
| 6 | Безкорпусный | 250 |
| 1 | Транз сборка | 20 |
| 9 | Безкорпусный | 25 |

**3NF**

Таблица находится в **третьей нормальной форме** (3НФ), если она удовлетворяет определению 2НФ и каждый не ключевой атрибут транзитивно зависит от первичного ключа.

Таблица 5 «Производитель»

|  |  |  |  |
| --- | --- | --- | --- |
| Код производителя | Код радиодетали | Код страны | Код фирмы |
| 1 | 1 | 3 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 1 | 3 |
| 4 | 4 | 1 | 4 |

Таблица 6 «Страна»

|  |  |
| --- | --- |
| Код страны | Страна |
| 1 | Россия |
| 2 | США |
| 3 | Япония |

Таблица 7 «Фирма»

|  |  |
| --- | --- |
| Код фирмы | Фирма |
| 1 | Panasonik |
| 2 | AMDX |
| 3 | Радио деталь |
| 4 | Наш транзистор |

Таблица 8 «Радиодеталь»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код радиодетали | Марка | Код материала | Код подкласса | Мощность |
| 1 | TXD5100 | 1 | 1 | 10 |
| 2 | MX300 | 2 | 2 | 5 |
| 3 | ТВ23У | 3 | 2 | 20 |
| 4 | ИБ10М | 4 | 1 | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Код частоты | Код температуры | Код типа перехода | Номер разработки |
| 1 | 1 | 1 | 2 |
| 1 | 3 | 2 | 22 |
| 2 | 1 | 2 | 88 |
| 3 | 2 | 2 | 35 |

|  |  |  |
| --- | --- | --- |
| Модернизация | Код типа сборки | Цена |
| 3 | 1 | 100 |
| 6 | 2 | 250 |
| 1 | 1 | 20 |
| 9 | 2 | 25 |

Таблица 9 «Материал»

|  |  |
| --- | --- |
| Код материала | Материал |
| 1 | Германий |
| 2 | Кремний |
| 3 | Галий |
| 4 | Индий |

Таблица 10 «Подкласс»

|  |  |
| --- | --- |
| Код подкласса | Подкласс |
| 1 | Биполярный |
| 2 | Полевой |

Таблица 11 «Частота»

|  |  |
| --- | --- |
| Код частоты | Частота |
| 1 | Малая |
| 2 | Средняя |
| 3 | Высокая |

Таблица 12 «Температура»

|  |  |
| --- | --- |
| Код температуры | Температура |
| 1 | от минус 50 до 50 |
| 2 | от минус 50 до 100 |
| 3 | от минус 50 до 150 |

Таблица 13 «Тип перехода»

|  |  |
| --- | --- |
| Код типа перехода | Тип перехода |
| 1 | pnp |
| 2 | npn |

Таблица 14 «Тип сборки»

|  |  |
| --- | --- |
| Код типа сборки | Тип сборки |
| 1 | Транз сборка |
| 2 | Безкорпусный |

Таблицы 5-14. по определению находятся в 3NF, потому что находятся в 2NF и здесь не присутствует неполных функциональных зависимостей не первичных атрибутов от атрибутов возможного ключа, а также нет транзитивных зависимостей.

Каждое отношение полученной реляционной модели находится в 3NF.

Т.к. в большинстве случаев достижение третьей нормальной формы считается достаточным для реальных проектов баз данных, то остановились на выше приведенной модели.

**Анализ предметной области**

Предметная область данной курсовой работы – Справочник радиолюбителя. Рассмотрены и систематизированы данные о производителях и продаваемых ими радиодеталях, такие как: данные производителя, страна, фирма, радиодеталь, материал, подкласс, частота, температура, тип перехода, тип сборки, номер разработки, и другие данные, касающиеся радиодетелей

**Инфологическое проектирование**

Сущность «Производитель» - показывает информацию о производителе выпускаемых радиодеталей. Сущность имеет следующие атрибуты: «Код производителя», «Код радиодетали», «Код страны», «Код фирмы».

Сущность «Страна» - предоставляет информацию о стране в которой выпускается радиодеталь. Сущность имеет следующие атрибуты: «Код страны », «Страна».

Сущность «Фирма» - отражает информацию о названии фирмы, которой выпускаются радиодетали. Сущность имеет следующие атрибуты: «Код фирмы», «Фирма».

Сущность «Радиодеталь» - содержит информацию, свойствах и параметрах радиодетали. Сущность имеет следующие атрибуты: «Код транзистора», «Марка», «Код материала», «Код подкласса», «Мощность», «Код частоты», «Код температуры», «Код типа перехода», «Номер разработки», «Модернизация», «Код типа сборки», «Цена».

Сущность «Материал» - показывает из какого материала смонтирована радиодеталь. Сущность имеет следующие атрибуты: «Код материала», «Материал».

Сущность «Подкласс» - содержит информацию о том, какому подклассу относится радиодеталь. Сущность имеет следующие атрибуты: «Код подкласса», «Подкласс».

Сущность «Частота» - содержит информацию о частотности радиодетали. Сущность имеет следующие атрибуты: «Код частоты», «Частота».

Сущность «Температура» - отражает информацию о допустимом температурном режиме в котором будет эксплуатироваться радиодеталь. Сущность имеет следующие атрибуты: «Код температуры», «Температура».

Сущность «Тип перехода» - содержит информацию о том, какой тип перехода используется в радиодетали. Сущность имеет следующие атрибуты: «Код типа перехода», «Переход».

Сущность «Тип сборки» - показывает собой, в каким исполнении спроектирована радиодеталь. Сущность имеет следующие атрибуты: «Код типа сборки», «Тип сборки».

Итого мы имеем 10 полностью нормализованных таблиц.

Данная программа будет написана на языке Pascal в среде разработки приложений Delphi 7.0. Это позволит использовать мощный аппарат языка в совокупности с возможностью визуализации программирования, предоставляемой средой.

Многочисленные компоненты управления, предоставляемые средой разработки, позволяют сделать интерфейс приложения более понятным пользователю и похожим на вид стандартных приложений широко распространенной операционной системы Windows.

Т.к. база данных для хранения информации о радиодеталях невелика, и не осуществляется никаких сложных работ по обработке этих данных, то в качестве СУБД выбираем Microsoft Access.

В качестве CASE-средства для проектирования модели данных эквивалентной реляциионной модели в третьей нормальной форме выбираем ERwin.

**3.3 Фаза построения**

На данном этапе происходит создание БД, программная реализация ИС, создание пользовательской документации. Тестирование и интеграция данной ИС.

Программный продукт “Справочник радиолюбителя” представляет собой набор таблиц баз данных, содержащих необходимую информацию для ведения справочной системы.

Для создания модели данных эквивалентной реляционной модели в третьей нормальной форме выбираем ERwin. После создания данной модели осуществляем генерацию базы данных в СУБД Microsoft Access.

Для последующей работы с этой базой в среде Delphi необходимо настроить источник данных, с помощью источника данных ODBC (там необходимо создать источник данных, выбрать тип Microsoft Access driver, и прописать путь к нашей базе. Затем необходимо создать Alias с помощью BDE Administrator, где будет указан источник данных и путь к базе данных.

Разработка ПО для нашей информационной системы осуществляется в среде Borland Delphi. Здесь создаются таблицы хранения данных, с помощью компонента DBGrid. Для выбора необходимой таблицы в главном окне программы расположены кнопки (компонент Button) с названиями этих таблиц. Для осуществления запроса на выборку компонент Edit.

Доступ к базе данных, связь БД и компонентов DBGrid осуществляется с помощью компонентов DataSource и Table. Для реализации запросов применяется компонент Query.

Код созданной программы можно просмотреть в приложении.

На этапе реализации необходимо разработать так же документацию пользователя, с подробным описанием функций программы и указанием действий пользователя для работы с данными. Руководство пользователя можно увидеть в приложении.

**3.4 Фаза внедрения ИС**

После создания программного продукта необходима его первоначальная загрузка, для того, чтобы проверить все функции данной программы. Проверяется возможность ввода данных (вводится некоторый тестовый набор данных), проверяется правильность выполнения запросов. Пользователю выдается Руководство пользователя для того, чтобы тот ознакомился с функциями программы, научился с ней работать.

В процессе эксплуатации данного программного продукта возможно появление ошибок – например, в программе не была предусмотрена обработка исключительных ситуаций на неверный ввод данных (не того типа), соответственно при вводе текста в строку с типом дата, программа будет выводить системную ошибку или зависать.

Так же в ходе эксплуатации может возникнуть потребность в модификации этой программы по причине изменения объекта и функций (например, возникнет необходимость в реализации новых запросов, вывода отчетов, появиться потребность в хранении еще каких то данных, или вследствие появления новых информационных технологий, потребуется усовершенствование этого программного средства).

1. **Выбор CASE-средства.**

В качестве CASE-средства для проектирования модели данных эквивалентной реляционной модели в третьей нормальной форме выбираем ERwin.

ERwin - средство разработки структуры базы данных. ERwin сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных.

Место ERwin в информационном моделировании

Процесс построения информационной модели состоит из следующих шагов:

* определение сущностей;
* определение зависимостей между сущностями;
* задание первичных и альтернативных ключей;
* определение атрибутов сущностей;
* приведение модели к требуемому уровню нормальной формы;
* переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы; задание триггеров, процедур и ограничений;
* генерация базы данных.

ERwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения как части документации, необходимой в цикле разработки. Однако ERwin далеко не только инструмент для рисования. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

Для проектирования модели данных запускаем Erwin 4.0. Выбираем создание новой модели данных. В Erwin возможно выбрать при проектировании физическую или логическую модель данных мы будем создавать, выбираем Logycal/Physical.

Для создания сущностей модели на панели инструментов выбираем компонент entity, задаем имя создаваемой сущности, имена и типы атрибутов, если атрибут ключевой ставим для этого атрибута соответствующее значение primary key. Для создания связей между сущностями выбираем компонент relationship, выбираем идентифицирующую связь. Связь устанавливаем, щелкая сначала по родительской, а затем по дочерней сущности. Ниже приведена диаграмма модели проектируемой ИС

Рис.4 «Схема модели ИС в Erwin (Logical Model)»

Рис.5 «Схема модели ИС в Erwin (Phisical Model)»

После того, как создана модель данных необходимо ее сгенерировать для конкретной СУБД. Для этого в меню Database->Choose database выбираем тип Access, версию 2000. (рис. 6)

Рис.6 «Выбор СУБД»

А затем осуществляем генерацию нашей модели для этой СУБД: Tools->Format Engeneer/Schema Generation. Выбираем компоненты, для генерации. После нажатия на клавишу Generate осуществляется генерация модели. Для СУБД Access перед генерацией необходимо изначально создать пустую базу данных Microsoft Access, при генерации модели будет запрашиваться пароль и путь для доступа к этой б.д. (рис. 7)

Рис.7 «Генерация модели данных»

На рис. 8 изображена схема данных ИС в Microsoft Access

Рис.8 «Схема данных»

**5. Описание программы**

**5.1 Общие сведения**

Программное приложение рассчитано на работу с базой данных «Справочник радиолюбителя» оно предназначено для выполнения следующих функций:

1. Вывод информации на экран;

2. Выполнение SQL запросов, осуществляющих необходимую выборку данных;

3. Удаление данных из таблиц;

4. Добавление информации.

Программа написана на языке Object Pascal в среде визуального программирования Borland Delphi 7. Разработано для операционных систем Windows 95/98/2000/NT/XP.

**5.2. Функциональное назначение**

Даная программа предназначена для работы с реляционной базой данных в Windows, решает задачи внесения, хранения, изменения и удаления информации из базы данных «Справочник радиолюбителя».

###### Заключение

В ходе выполнения работы укрепил знания по проектированию информационных систем, применению инструментальных средств поддержки проектирования ИС.

Литература.

1. Вендров А.М. Один из подходов к выбору средств проектирования баз данных и приложений. "СУБД", 1995, №3.

2. Калянов Г.Н. CASE. Структурный системный анализ (автоматизация и применение). М., "Лори", 1996.

ПРИЛОЖЕНИЕ 1

Для запуска программы открыть файл “Radiodetal.exe”, который располагается в рабочей папке программы.

Главная форма программы:

Рис.9 «Главная форма программы»

В данной программе реализованы следующие запросы на выборку:

По производителю

* Россия
* США
* Япония

Выбор страны производителя:

 Рис. 10 «Выбор страны производителя»

При выборе производителя, выводится информация о радиодеталях произведенных в конкретной стране, а именно марка, фирма производитель, технические характеристики и цены.

Рис. 11 «Радиодетали произведенные в России»

Рис. 12 «Радиодетали произведенные в США»

По типу электронно-дырочного перехода:

* pnp
* npn

При выборе какого-либо перехода, отображается информация о радиодетали с выбранным типом перехода.

Рис. 13 «Выбор типа перехода»

При выборе типа: npn, отображаются все радиодетали имеющие данный переход.

Рис. 14 «Тип перехода: npn»

Организована возможность просмотра справочников по материалам и типам сборки.

Рис. 15 «Выбор справочника»

Справочник «Материал»

Рис. 16 «Справочник материал»

В базе данных предусмотрен поиск.

В поисковое поле нужно ввести номер разработки радиодетали.

Рис. 17 «Поисковое поле»

В результате выполнения поиска, в таблице будет отображена радиодеталь с номером разработки введенным в поиск, если таковой номер имеется.

 Рис. 18 «Результат поиска»

ПРИЛОЖЕНИЕ 2

unit Unit1;

interface

uses

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

 Dialogs, DB, ADODB, Grids, DBGrids, Menus, StdCtrls;

type

 TForm1 = class(TForm)

 ADOConnection1: TADOConnection;

 ADOTable1: TADOTable;

 DataSource1: TDataSource;

 DBGrid1: TDBGrid;

 ADOTable1DSDesigner: TAutoIncField;

 ADOTable1DSDesigner2: TWideStringField;

 ADOTable1DSDesigner3: TIntegerField;

 ADOTable1DSDesigner4: TIntegerField;

 ADOTable1DSDesigner5: TIntegerField;

 ADOTable1DSDesigner6: TIntegerField;

 ADOTable1DSDesigner7: TIntegerField;

 ADOTable1DSDesigner8: TIntegerField;

 ADOTable1DSDesigner9: TIntegerField;

 ADOTable1DSDesigner10: TIntegerField;

 ADOTable1DSDesigner11: TIntegerField;

 ADOTable1DSDesigner12: TBCDField;

 MainMenu1: TMainMenu;

 N1: TMenuItem;

 N2: TMenuItem;

 N3: TMenuItem;

 N4: TMenuItem;

 N5: TMenuItem;

 pnp1: TMenuItem;

 npn1: TMenuItem;

 N6: TMenuItem;

 N7: TMenuItem;

 N8: TMenuItem;

 N9: TMenuItem;

 N10: TMenuItem;

 N11: TMenuItem;

 ADOQuery1: TADOQuery;

 GroupBox1: TGroupBox;

 Edit1: TEdit;

 Button1: TButton;

 procedure N2Click(Sender: TObject);

 procedure N3Click(Sender: TObject);

 procedure npn1Click(Sender: TObject);

 procedure N10Click(Sender: TObject);

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

uses Unit2, Unit3, Unit4, Unit5, Unit6;

{$R \*.dfm}

procedure TForm1.N2Click(Sender: TObject);

begin

form2.ADOQuery1.Active:=false;

form2.ADOQuery1.SQL.Clear;

form2.ADOQuery1.SQL.Add('SELECT Радиодеталь.Марка, Фирма.Фирма, Радиодеталь.[Код материала], Радиодеталь.Мощность, Радиодеталь.Цена, Страна.Страна');

form2.ADOQuery1.SQL.Add('FROM Фирма INNER JOIN (Страна INNER JOIN (Радиодеталь INNER JOIN Производитель ON Радиодеталь.[Код радиодетали] = Производитель.[Код радиодетали]) ON Страна.[Код страны] = Производитель.[Код страны]) ON Фирма.[Код фирмы] = Производитель.[Код фирмы]');

form2.ADOQuery1.SQL.Add('WHERE (((Страна.Страна)="Россия"));');

form2.ADOQuery1.Active:=true;

 Form2.ShowModal;

end;

procedure TForm1.N3Click(Sender: TObject);

begin

form3.ADOQuery1.Active:=false;

form3.ADOQuery1.SQL.Clear;

form3.ADOQuery1.SQL.Add('SELECT Радиодеталь.Марка, Фирма.Фирма, Радиодеталь.[Код материала], Радиодеталь.Мощность, Радиодеталь.Цена, Страна.Страна');

form3.ADOQuery1.SQL.Add('FROM Фирма INNER JOIN (Страна INNER JOIN (Радиодеталь INNER JOIN Производитель ON Радиодеталь.[Код радиодетали] = Производитель.[Код радиодетали]) ON Страна.[Код страны] = Производитель.[Код страны]) ON Фирма.[Код фирмы] = Производитель.[Код фирмы]');

form3.ADOQuery1.SQL.Add('WHERE (((Страна.Страна)="США"));');

form3.ADOQuery1.Active:=true;

 Form3.ShowModal;

end;

procedure TForm1.npn1Click(Sender: TObject);

begin

 form4.ADOQuery1.Active:=false;

form4.ADOQuery1.SQL.Clear;

form4.ADOQuery1.SQL.Add('SELECT Радиодеталь.Марка, Радиодеталь.[Код материала], Радиодеталь.[Код подкласса], Радиодеталь.[Код частоты], [Тип перехода].[Тип перехода]');

form4.ADOQuery1.SQL.Add('FROM [Тип перехода] INNER JOIN Радиодеталь ON [Тип перехода].[Код типа перехода] = Радиодеталь.[Код типа перехода]');

form4.ADOQuery1.SQL.Add('WHERE ((([Тип перехода].[Тип перехода])="npn"));');

form4.ADOQuery1.Active:=true;

 Form4.ShowModal;

end;

procedure TForm1.N10Click(Sender: TObject);

begin

 Form5.ShowModal;

end;

procedure TForm1.Button1Click(Sender: TObject);

 begin

 if edit1.Text='' then

 ShowMessage('Введите № разработки!')

 else

 begin

form6.ADOQuery1.Active:=false;

form6.ADOQuery1.SQL.Clear;

form6.ADOQuery1.SQL.Add('SELECT Радиодеталь.[Номер разработки], Страна.Страна, Радиодеталь.Марка, Радиодеталь.Цена');

form6.ADOQuery1.SQL.Add('FROM Страна INNER JOIN (Радиодеталь INNER JOIN Производитель ON Радиодеталь.[Код радиодетали]=Производитель.[Код радиодетали]) ON Страна.[Код страны]=Производитель.[Код страны]');

form6.ADOQuery1.SQL.Add('WHERE Радиодеталь.[Номер разработки]='+form1.edit1.text+';');

form6.ADOQuery1.Active:=true;

 Form6.ShowModal;

end;

 end.