

1. Описание автоматизированной информационной системы «Учет курсовых работ»

1.1. Постановка задачи

Ведение учета выполненных и сданных работ является весьма важным аспектом работы колледжа. В течение многих лет функционирования при большом количестве выполняемых работ хранение данных о работах превращается в настоящую проблему. При ведении бумажного каталога поиск работы в каталоге весьма затруднен. Поиск может понадобиться в том случае, если нужно сверить данные работы, проверить новую работу на совпадение с работами в каталоге. Кроме поиска могут понадобиться такие задачи как отбор работ по определенным критериям, периодам выполнения работ, авторам и руководителям работ. Все эти задачи успешно могут быть выполнены в случае ведения автоматизированного учета при помощи специального приложения. При этом все данные о работах хранятся в базе данных, а приложение позволяет вносить информацию в базу данных и работать с ней.

Введение автоматизированного учета выполненных курсовых работ повысит эффективность их поиска, ведения их каталога и отбора работ по определенным критериям.

В результате работы должно быть разработано приложение, позволяющее автоматизировать ведение учета курсовых работ, выполненных студентами колледжа. Учет курсовых работ должен вестись в виде каталога выполненных работ, хранящего развернутый набор данных о каждой работе. Ведение такого каталога должно позволить упростить поиск работ в каталоге и их отбор по различным критериям за счет:

- 1) автоматизации ведения каталога выполненных работ;
- 2) ведения списков студентов, руководителей, групп и специальностей и выбор данных из них при внесении новых работ и поиске существующих работ;
- 3) повышения оперативности доступа к информации о выполненных работах;
- 4) автоматизации процесса поиска и отбора данных;
- 5) обеспечения полноты и актуальности данных, необходимых для принятия управленческих решений;
- 6) применения современных средств анализа информации;
- 7) наглядного и удобного представления информации;
- 8) печати списка работ и карточки каждой работы;
- 9) получения развернутой информации о работе путем прикрепления к записям базы данных документов самой работы.

Программа ведения автоматизированного каталога выполненных работ должна иметь развитый и удобный пользовательский интерфейс.

Многие Windows – приложения имеют такой вид защиты информации, как парольная защита. Поэтому, чтобы не пренебрегать информацией хранящейся в базе данных, нужно создать парольную защиту для разрабатываемого приложения, с возможностью изменения пароля.

Ведение списка связанных документов позволит повысить информативность каталога. Если информации в базе данных по выполненным работам недостаточно, пользователь всегда сможет получить развернутую информацию о работе из самих документов работы. Важно обеспечить открытие приложений, связанных с документами, непосредственно из каталога выполненных работ.

1.2. Описание разрабатываемого приложения

В основу программной реализации решения задачи был положен объектно-

ориентированный подход. Выбранный в качестве языка программирования язык Delphi включает в себя мощный аппарат для поддержания этой наиболее перспективной технологии: классы и объекты, иерархия структур, объединений и классов с различными механизмами наследования, инкапсуляция данных и обрабатывающих функций с несколькими уровнями доступа, полиморфизм операций, функций и объектов, обеспечивающий как статический так и виртуальный методы доступа. Встроенный аппарат объектно-ориентированного программирования позволяет описывать сложные предметные области и создавать гибкие, легко адаптируемые к нуждам пользователя программные пакеты как системного так и прикладного назначения.

Основным инструментом реализации объектно-ориентированного подхода в Delphi является класс. Для реализации задачи были созданы следующие классы. Класс TFMMain – класс главной формы приложения, наследник от стандартного класса TForm библиотеки VCL. Этот класс обеспечивает функционирование главной формы приложения и главного меню приложения. Следует отметить также объект TApplication, создаваемый в программе неявно и обеспечивающий функционирование приложения.

Для реализации каждой подзадачи в проекте были созданы несколько классов, наследников стандартного класса библиотеки VCL TForm.

Программа разрабатывалась как Windows-приложение и удовлетворяет всем требованиям к интерфейсу Windows-приложения. Программа разбита на модули, что соответствует принципам модульного программирования. Рассмотрим каждый из этих модулей подробно.

Первый модуль приложения предназначен для защиты приложения. Защита осуществляется при помощи проверки пароля, вводимого пользователем.

Модуль главной формы отображает список выполненных студентами работ, а так же связанных с выделенной работой документами. Интерфейс данного модуля включает в себя меню, которое обеспечивает доступ к другим модулям приложения, и кнопки для добавления, удаления работы и печати карточки выбранной работы. Этот модуль так же позволяет производить поиск данных по различным критериям. К ним относятся: поиск по краткому названию, по полному названию, по ключевым словам, по дате сдачи работы, по группе, по специальности, по имени автора, по фамилии автора, по имени руководителя, а так же по фамилии руководителя. Отсортированные данные так же отображаются в этом же модуле.

Третий модуль представляет собой модуль для отображения справочника Специальности. Этот модуль позволяет осуществлять добавление и редактирование таблицы Spes, а так же осуществлять поиск по всему справочнику.

Четвертый модуль отображает справочник Группы. Он позволяет просматривать поля таблицы Gr, добавлять новые, редактировать любое выбранное поле, осуществлять поиск по группе.

Пятый модуль предназначен для отображения справочника Руководители проектов. Этот модуль так же позволяет добавлять, редактировать и удалять записи из справочника, производить поиск руководителя по имени, фамилии и отчеству.

Шестой модуль отображает справочник Автор. Этот модуль тесно взаимосвязан с третьим и четвертым модулями. При добавлении работы открываются вышеперечисленные модули, для выбора данных. Шестой модуль позволяет просмотреть весь справочник авторов работ, добавлять, редактировать и удалять записи, осуществлять поиск по имени, отчеству и фамилии автора.

Седьмой модуль представляет собой отчет, который содержит в себе весь список работ.

Восьмой модуль предназначен для смены пароля, который запрашивается при открытии приложения. Пароль необходимо вводить дважды, чтобы программа разрешила смену пароля. Этот модуль так же как и первый взаимодействует с текстовым файлом parol.txt, в котором прописан пароль.

Девятый модуль предназначен для добавления новой работы в базу данных. Этот модуль активно взаимодействует с пятым и шестым модулем приложения для выбора данных. Так же этот модуль позволяет добавлять документы к новой работе.

Десятый модуль так же представляет собой отчет. Это карточка работы. На нем расположена такая информация: краткое наименование работы, полное наименование работы, руководителя работы, автора работы, дата сдачи работы и ключевые слова.

Одиннадцатый модуль – это модуль подтверждения выхода из программы.

В визуальной интегрированной среде разработки Borland Delphi 7 основной составной частью программы является форма. Форма инкапсулирует в себе элементы интерфейса пользователя и программный код, заключающий в себе реакцию на действия пользователя. Это полностью отвечает принципам объектно-ориентированного программирования. Delphi автоматически создает для формы класс формы, который содержит компоненты, размещенные на форме, свойства и методы формы как автоматически создаваемые Delphi, так и определяемые пользователем. В разработанном приложении определены следующие формы:

1) Форма Form1 – форма приложения, осуществляющая парольную защиту, реализованная в файле Unit1.cpp;

2) Форма Form2 – главная форма приложения, реализованная в файле Unit2.cpp;

3) Форма Form3 – форма приложения, представляющая собой справочник Специальности, реализованная в файле Unit3.cpp;

4) Форма Form4 – форма приложения, представляющая собой справочник Группы, реализованная в файле Unit4.cpp;

5) Форма Form5 – форма приложения, представляющая собой справочник Руководители, реализованная в файле Unit5.cpp;

6) Форма Form6 – форма приложения, представляющая собой справочник Авторы, реализованная в файле Unit6.cpp;

7) Форма Form7 – форма приложения, представляющая собой отчет Список работ, реализованная в файле Unit7.cpp;

8) Форма Form8 – форма приложения, которая предназначена для изменения пароля, реализованная в файле Unit8.cpp;

9) Форма Form9 – форма приложения, реализующая добавление новой работы со всеми атрибутами в список работ, реализованная в файле Unit9.cpp;

10) Форма Form10 – форма приложения, представляющая собой отчет Карточка работы, реализованная в файле Unit10.cpp;

11) Форма Form11 – форма приложения, предназначенная для подтверждения выхода из приложения, реализованная в файле Unit11.cpp.

Каждая форма выполняет функции соответствующего модуля при помощи компонентов среды Delphi, помещенных на форму из палитры компонентов. Среда быстрого создания приложений или RAD-среда (Rapid Application Development – RAD) Delphi используют библиотеку визуальных компонентов VCL (Visual Component Library – VCL), которая состоит из готовых к употреблению визуальных и не визуальных объектов и оболочек. Она позволяет с минимальными затратами создавать приложения, в то же время предоставляя определенную степень независимости от библиотеки VCL.

При работе с компонентами Delphi широко использует принцип повторного использования объектов. Компоненты являются экземплярами классов которые доступны с помощью палитры компонентов Component Palette. Что может быть проще при создании приложения чем просто опустить нужный компонент на форму, задав его свойства, затем определив обработчики событий. Именно при помощи компонентов формы выполняют те функции, которые заложены в них.

Отдельно остановимся на таком моменте как открытие связанного с работой документа в том приложении, с которым связан данный документ.

При проектировании базы данных невозможно предусмотреть все поля, которые могут понадобиться для описания выполненной работы, так как могут быть разные требования к содержимому работ. Кроме того, нередко возникает необходимость хранения каких-либо данных работы, содержащихся в различных документах. В программе предусмотрена возможность связать с каждой работой список файлов, подготовленных в других приложениях Windows. Это может быть документ Word, таблица Excel, рисунок Bitmap или любой другой файл. Для каждой работы в таблице Docs.db хранится список связанных файлов. Связь таблиц spisok.db и docs.db выполняется по уникальному идентификатору работы (поле ID_SPISOK в таблице spisok.db и поле ID_DOCS в таблице docs.db). Каждая запись в таблице Docs.db содержит полный путь к файлу, связанному с данной работой. Файл должен быть создан заранее в каком-либо приложении Windows.

Для запуска программы на выполнение необходимо запустить выполняемый файл KursovRabota.exe. Сделать это можно любым доступным в операционной системе методом. Для быстрого запуска программы рекомендуется создать ярлык для выполняемого файла и запускать программы щелчком мыши.

Перед первым запуском программы необходимо создать алиас базы данных с именем kursov при помощи программы BDEAdministrator. Данная программа находится в панели управления Windows и устанавливается вместе с системой BDE. Сама же система BDE входит в комплект поставки системы разработки Borland Delphi 7 и ставится с этой системой по умолчанию. Если же программа устанавливается на компьютер, на котором отсутствует система разработки, необходимо установить систему BDE отдельно, либо с диска установки Delphi, выбрав в списке устанавливаемых компонент только систему BDE. Сама база данных (файлы таблиц и индексов) также должны быть скопированы на жесткий диск компьютера перед первым запуском программы в отдельный каталог с установочного диска программы. На этот и должен быть настроен система kursov.

Входными данными для программы должны являться таблицы, содержащие информацию о:

- 1) Специальностях колледжа, в котором учатся авторы работ;
- 2) Группам, в которых учатся студенты-авторы выполненных работ;
- 3) Авторах выполненных работ;
- 4) Руководителях выполненных работ.

Все эти данные вводятся в формах соответствующих модулей при помощи клавиатуры и мыши. Эти данные являются справочниками, из которых в дальнейшем будут выбираться значения для заполнения соответствующих полей при вводе основных входных данных программы – списка выполненных работ. Данные по каждой работе заполняются также с клавиатуры в полях соответствующей формы и выбираются из соответствующих справочников.

Программа должна предоставлять возможность корректировки данных в таблицах, а также внесения данных о новых специальностях, группах, авторах, руководителях, выполненных работах и связанных документах. Кроме того, программа должна предоставлять возможность удалять данные, не нарушая при этом ссылочной целостности базы данных.

Выходными данными программы является сформированный список выполненных работ, который может быть отфильтрован по любому полю и который предоставляет возможности быстрого поиска в нем работы по любому из полей. К выходным данным можно также отнести распечатку списка работ, а также карточки каждой работы на бумажных носителях.