Содержание

Введение

Информационно-справочная система - это автоматизированная система, предназначенная для организации, хранения, пополнения, поддержки и представления пользователям информации в соответствии с их запросами. Информация, выдаваемая информационной системой потребителю, является одним из ресурсов, позволяющих повысить производительность труда и эффективность его деятельности.

Целью практики является реализация информационно-справочной системы «Отдел кадров» на языке программирования C#, с использованием технологии сериализации объектов.

Для достижения поставленной цели решены следующие задачи:

. Изучить основы информационно-справочных систем

. Изучить основы технологии сериализации объектов.

. Реализовать информационно-справочную систему.

Данная работа состоит из ведения, трех глав, заключения и приложения. В первой главе рассмотрены виды информационно-справочных систем. Вторая глава посвящена технологии сериализации. В третьей главе информационно-справочной системы «Отдел кадров».

1. Информационные системы, их классификация

Информационная система - это система обработки информации и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию. Хотя информационные системы являются обычным программным продуктом, они имеют ряд существенных отличий от стандартных прикладных программ и систем. В зависимости от предметной области информационные системы могут весьма значительно различаться по своим функциям, архитектуре, реализации. Однако можно выделить ряд свойств, которые являются общими.

. Информационные системы предназначены для сбора, хранения и обработки информации, поэтому в основе любой из них лежит среда хранения и доступа к данным.

. Информационные системы ориентированы на конечного пользователя, не обладающего высокой квалификацией в области вычислительной техники. Поэтому клиентские приложения информационной системы должны обладать простым, удобным, легко осваиваемым интерфейсом, который предоставляет конечному пользователю все необходимые для работы функции и в то же время не даёт ему возможность выполнять какие-либо лишние действия.

Таким образом, при разработке информационной системы приходится решать две основные задачи - разработка базы данных для хранения информации и разработка графического интерфейса пользователя клиентских приложений.

Классификация информационных систем

Информационные системы классифицируются по разным признакам.. По масштабу

.Одиночные - реализуются, как правило, на автономном персональном компьютере и рассчитаны на одного пользователя или группы пользователей, разделяющих по времени одно рабочее место.

. Групповые информационные системы - ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе локальной вычислительной сети.

. Корпоративные информационные системы - являются развитием систем для рабочих групп, они ориентированы на крупные компании и могут поддерживать территориально разнесённые узлы и сети.. По сфере применения

. Системы обработки транзакций
. Системы поддержки принятия решений
. Информационно-справочные системы
. Офисные информационные системы. По способу организации
. Системы на основе архитектуры файл-сервер
. Системы на основе архитектуры клиент-сервер
. Системы на основе многоуровневой архитектуры
. Системы на основе Интернет технологий

Области применения информационных систем:

- Бухгалтерский учёт
- Управление финансовыми потоками
- Управление складом, ассортиментом, закупками
- Управление производственным процессом.
- Управление маркетингом.
- Документооборот.
- Оперативное управление предприятием.
- Предоставление информации о фирме.

2. Технология сериализации объектов на платформе.NET

Термин сериализация описывает процесс сохранения (и, возможно, передачи) состояния объекта в потоке (т.е. файловом потоке и потоке в памяти). Последовательность сохраняемых данных содержит всю необходимую информацию, необходимую для реконструкции (или десериализации) состояния объекта с целью последующего использования. Применяя эту технологию, очень просто сохранять большие объемы данных (в различных форматах) с минимальными усилиями.

Для осуществления (де)сериализации.NET предлагает 3 разных варианта (не считая самостоятельной реализации механизма сериализации):

- Сериализация в двоичный формат (BinnaryFormatter)
- Сериализация в формат SOAP (SoapFormatter)
- Сериализация в формат xml (XmlSerializer)

Тип Binary Formatter сериализует состояние объекта в поток, используя компактный двоичный формат. Этот тип определен в пространстве имен System. Runtime. Serialization. Formatters. Binary, которое входит в сборку mscorlib.dll.

В этом случае сериализируются все поля, вне зависимости от их области видимости. Исключение составляют поля помеченные атрибутом [Non Serialized]. Помимо сохранения данных полей, Binary Formatter также сохраняет полное квалифицированное имя каждого типа, полное имя сборки, где он определен, сюда входит информация об имени, версии, маркере общедоступного ключа (public key). Здесь заключается основной минус Binary Formatter - данные, сохраненные с его помощью, могут быть воссозданы только в инфраструктуре CLI. Причем каждый, кто будет восстанавливать данные, должен иметь сборку с сериализуемым типом.

Сериализация происходит с помощью двух ключевых методов Serialize и Deserialize. Первый сохраняет граф объектов в виде последовательности байт в указанный поток. Второй наоборот - преобразует сохраненную последовательность байт в граф объектов.

Тип Soap Formatter сохраняет состояние объекта в виде сообщения SOAP (стандартный XML-формат для передачи и приема сообщений от веб-служб). Этот тип определен в пространстве имен System. Runtime. Serialization. Formatters. Soap, находящемся в отдельной сборке. Также как и BinaryFormatter сериализирует все поля, вне зависимости от их области видимости, кроме полей помеченных атрибутом [NonSerialized]. В отличие от Binary Formatter, платформа и операционная система не влияют на успешное восстановление данных, сериализированных с помощью Soap Formatter. Как и в случае с Binary Formatter (де) сериализация происходит с помощью ключевых методовSerialize()иDeserialize().

И, наконец, для сохранения дерева объектов в документе XML имеется тип Xml Serializer. Чтобы использовать этот тип, нужно указать директиву using для пространства имен System. Xml. Serialization и установить ссылку на сборку System.Xml.dll. Данный тип сериализации не сохраняет приватные данные. Хотя это можно сделать, инкапсулировав такое поле в общедоступном свойстве.

Также Xml Serializer не сохраняет точную информацию о типе (квалифицированное имя, имя сборки и т.д.), что делает его идеальным кандидатом, когда необходимо сохранить объект для дальнейшего использования в другом языке программирования, а также на любой платформе, в любой операционной системе. Сериализация с помощью Xml Serializer немного отличается от сериализации с помощью Binary Formatter и Soap Formatter. Xml Serializer требует указания информации о типе, который

нужно сериализовать.

3. Реализация информационно-справочной системы «Отдел кадров»

Информационно-справочная система «Отдел кадров» включает в себя следующий функционал:

- Добавление
- Изменение
- Удаление
- Просмотр данных
- Загрузка данных из файла
- Сохранение данных в файл

Таблица данных имеет следующие поля:

- Табельный номер
- Фамилия
- Имя
- Отчество
- Должность
- Номер кабинета
- Внутренний телефон

В приложении представлен программный код, реализующий систему с заявленным функционалом.

Заключение

В результате практики я изучила технологию сериализации объектов на платформе.NET, а также реализовала информационно-справочную систему с использованием этой технологии.

Реализуя систему, изучила основы еще одного объектно-ориентированного языка программирования C#. Кроме того, научилась реализовывать Windows-приложения, использовать средства конструктора баз данных Windows Forms.

Список используемых источников

1. Троелсен Э. Язык программирования C# 2010 и платформа.NET 4.0, 5 издание. Пер. с англ. - М.: ООО "И.Д. Вильяме", 2011. - 1392 с.

. Шилдт Г. C# 4.0: полное руководство. Пер. с англ. - М.: ООО "И.Д. Вильямс", 2011. -1056 с.

Приложение.

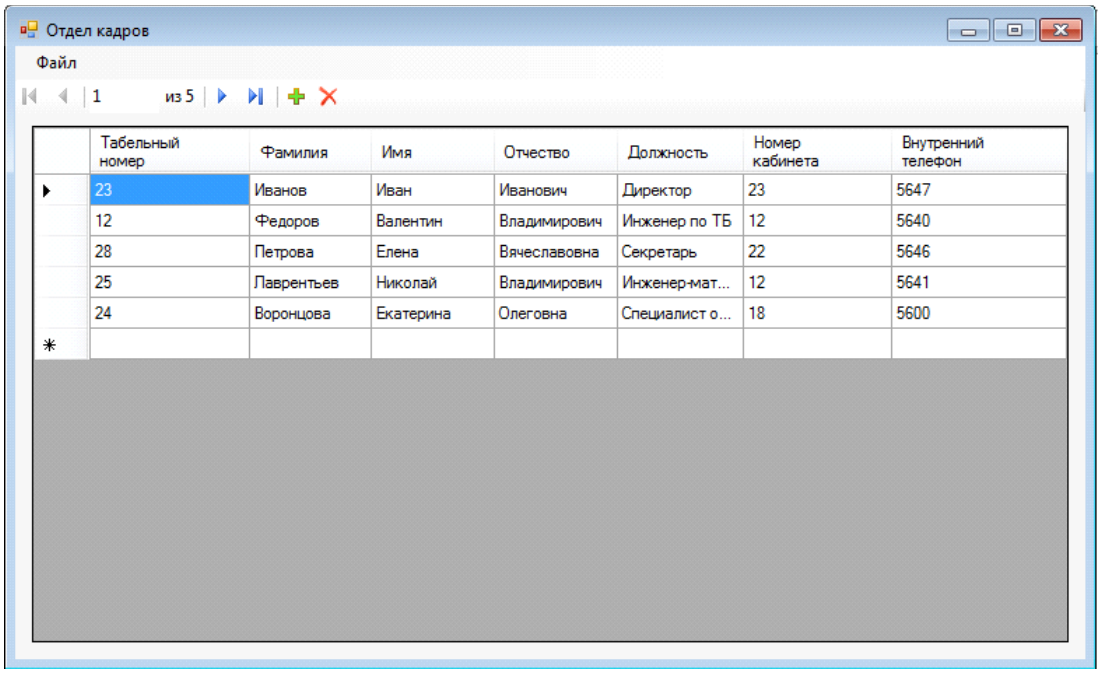
```
//staff.cs
using System;System.Xml.Serialization;StaffOffice
{
/// <summary>
/// Сотрудник
/// </summary>
public class Staff
{
private string _surname; //фамилия
private string _name; //имя
private string _middleName; //отчество
private string _office; //должность
private int interPhone; //внутренний телефон
[XmlAttribute]
public int TabNumber { get; set; }
[XmlAttribute]
public int Cabinet { get; set; }int InterPhone { get; set; }string Name
{
get { return _name; }
set { _name = value.Trim(); }
}string Surname
{
get { return _surname; }
set { _surname = value.Trim(); }
}string MiddleName
```

```
        {
        get { return _middleName; }
        set { _middleName = value.Trim(); }
        }string Office
        {
        get { return _office; }
        set { _office = value.Trim(); }
        }
        }
        }
        //Otdel.cs
        using
System;System.Collections.Generic;System.Linq;System.Text;System.Component
Model;System.IO;System.Xml.Serialization;StaffOffice
        {
        [XmlRoot("Otdel")]
        public class Otdel: BindingList<Staff>
        {
        public void Load(string fileName)
        {
        Clear();
        var serializer = new XmlSerializer(typeof(Otdel));
        TextReader textReader = new StreamReader(fileName);
        var list = (Otdel)serializer.Deserialize(textReader);
        foreach (Staff staff in list)
        Add(staff);
        }void Save(string fileName)
```

```
        {
        var serializer = new XmlSerializer(typeof(Otdel));
        TextWriter textWriter = new StreamWriter(fileName);
        serializer.Serialize(textWriter, this);
        textWriter.Close();
        }
        //Program.cs
        using
System;System.Collections.Generic;System.Linq;System.Windows.Forms;StaffOff
ice
        {
        static class Program
        {
        [STAThread]
        static void Main()
        {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
        }
        //Form1.cs
        using
System;System.Collections.Generic;System.ComponentModel;System.Data;Syste
m.Drawing;System.Linq;System.Text;System.Windows.Forms;StaffOffice
        {
        public partial class Form1: Form
        {
```

```
private readonly Otdel _otdel;Form1()
{
InitializeComponent();
_otdel = new Otdel();
otdelBindingSource.DataSource = _otdel;
openFileDialog.InitialDirectory = Environment.CurrentDirectory;
saveFileDialog.InitialDirectory = Environment.CurrentDirectory;
}void FileOpen_Click(object sender, EventArgs e)
{
if (openFileDialog.ShowDialog() == DialogResult.OK)
_otdel.Load(openFileDialog.FileName);
}void FileSave_Click(object sender, EventArgs e)
{
if (saveFileDialog.ShowDialog() == DialogResult.OK)
_otdel.Save(saveFileDialog.FileName);
}void Exit_Click(object sender, EventArgs e)
{
Application.Exit();
}
```

Пример работы программы:


Отдел кадров
Файл
1
из 5
Табельный номер
Фамилия
Имя
Отчество
Должность
Номер кабинета
Внутренний телефон
23
Иванов
Иван
Иванович
Директор
23
5647
12
Федоров
Валентин
Владимирович
Инженер по ТБ
12
5640
28
Петрова
Елена
Вячеславовна
Секретарь
22
5646
25
Лаврентьев
Николай
Владимирович
Инженер-мат...
12
5641
24
Воронцова
Екатерина
Олеговна
Специалист о...
18
5600