```
using System;
using System.Data.SqlClient;
using Microsoft.Win32;
using Application.Utils;

namespace Application.DataStorage
{
    public class ConnectException : Exception
    {
        public ConnectException(Exception innerException) :
            base("Could not be possible to connect the database (" +
                innerException.Message + ").", innerException) { }
    }

    public class DisconnectException : Exception
    {
        public DisconnectException(Exception innerException) :
            base("Could not be possible to disconnect the database (" +
                innerException.Message + ").", innerException) { }
    }

    public abstract class DatabaseStorage : IDisposable
    {
        private const string _dbSourceFormat =
            "Data Source={0};" +
            "Initial Catalog={1};" +
            "User ID={2};" +
            "Password={3};" +
            "Integrated Security=False;" +
            "Persist Security Info=True;" +
            "Connect Timeout=30";
        private const string _dbSourceFormatDefault =
            "Data Source={0};" +
            "Initial Catalog={1};" +
            "Integrated Security=True;" +
            "Connect Timeout=30";
        private const string _dbServer =
            "(local)";
        private const string _dbRegistryValue =
            "Database";

        private string _productName = string.Empty;

        protected SqlConnection Connection { get; private set; } = null;

        private string ConnectionString
        {
            get
            {
                string connStr;
                try
                {
                    RegistryKey rk = Registry.LocalMachine.OpenSubKey("SOFTWARE\\" + _productName);
                    string dbEncoded = rk.GetValue(_dbRegistryValue).ToString();
                    string dbDecoded = CryptographyUtil.Decode(dbEncoded);
                    string[] db = dbDecoded.Split(';');
                    string server = db[0].Trim();
                    string name = db[1].Trim();
                    string account = db[2].Trim();
                    string password = db[3].Trim();
                    connStr = String.Format(_dbSourceFormat, server, name, account, password);
                }
                catch
                {
                    string server = _dbServer;
                    string name = _productName;
                connStr = String.Format(_dbSourceFormatDefault, server, name);
            }
            return connStr;
        }
    }

    public DatabaseStorage(string productName)
    {
        _productName = productName;
        Connection = null;
        Connect();
    }

    ~DatabaseStorage() =>
        Disconnect();

    public void Dispose() =>
        Disconnect();

    private SqlConnection OpenConnection()
    {
        SqlConnection con = new SqlConnection(ConnectionString);
        con.Open();
        return con;
    }

    private void CloseConnection(SqlConnection con) =>
        con.Close();

    private void Connect()
    {
        try
        {
            Connection = OpenConnection();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
            throw new ConnectException(ex);
        }
    }

    private void Disconnect()
    {
        if (Connection != null)
        {
            try
            {
                CloseConnection(Connection);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
                throw new DisconnectException(ex);
            }
            finally
            {
                Connection = null;
            }
        }
    }
  }
}
```