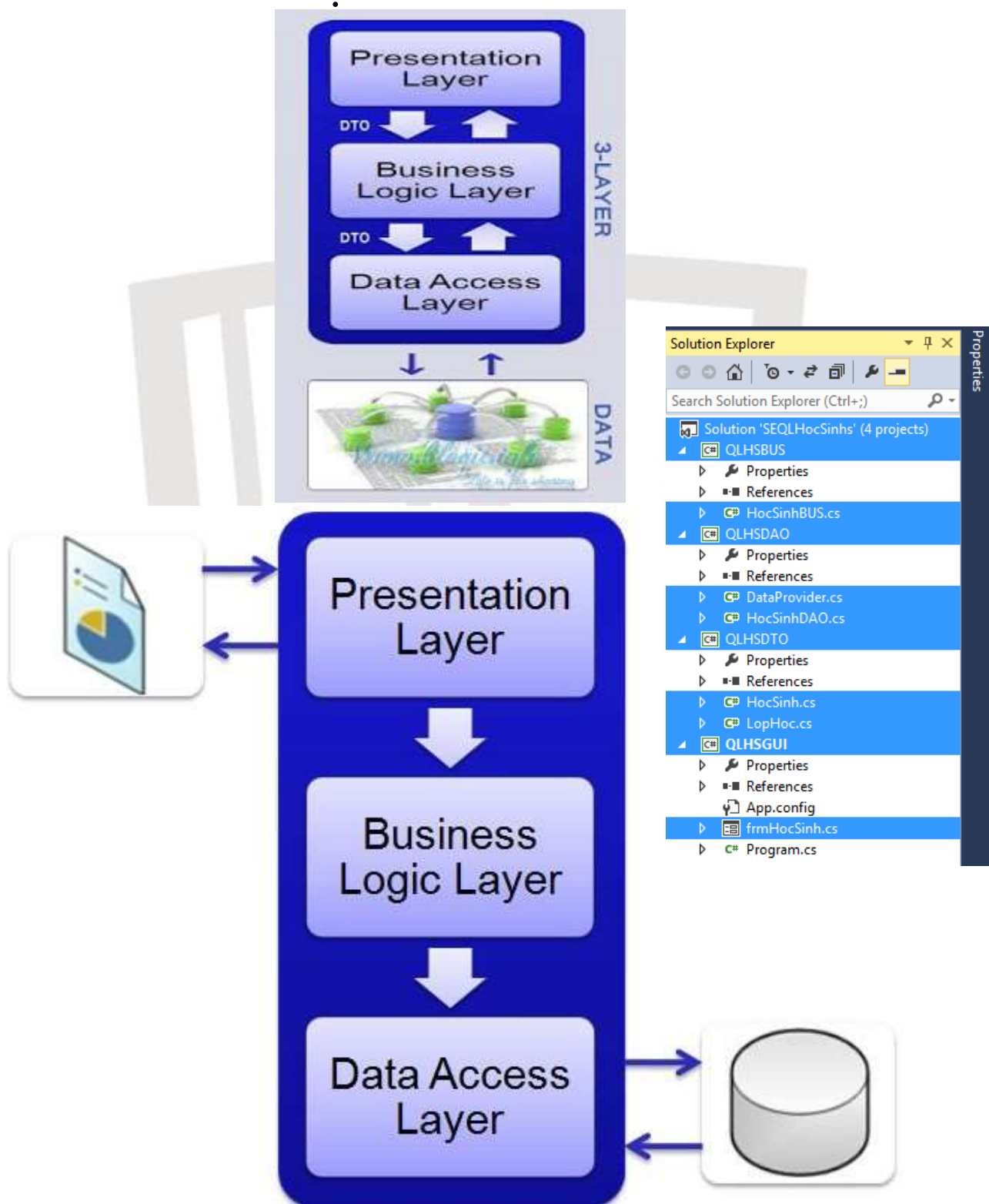


# HƯỚNG DẪN THỰC HÀNH MÔ HÌNH 3LAYER

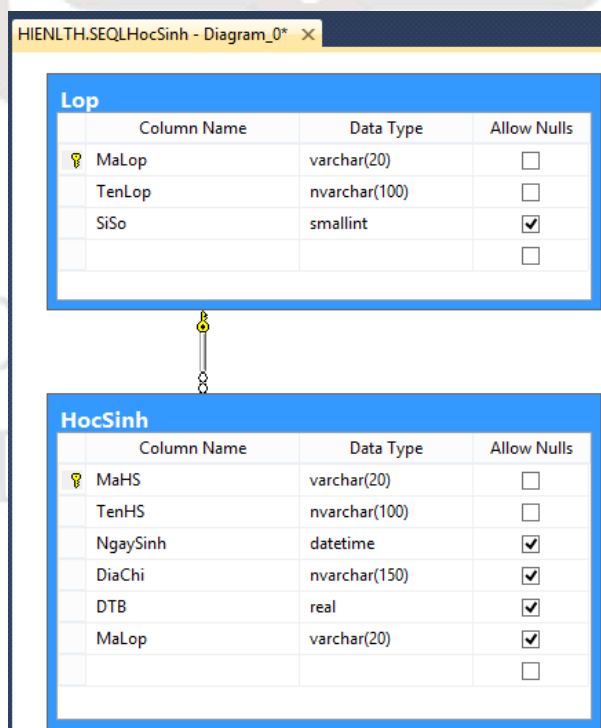


## 1 Giới thiệu

Mô hình 3-layer gồm có 3 layer:

- Layer GUI (Graphics User Interface, tên khác Presentation Layer):
  - Đây là layer tạo lên giao diện cho người dùng, nó sẽ là nơi tiếp nhận và kết xuất ra kết quả của chương trình cho bạn.
  - Nó có nhiệm vụ xử lý, kiểm tra các dữ liệu nhập vào ( ví như ở TextBox này nó phải là số, số phải từ 1-9....).
  - Nó tiếp nhận các Event của người dùng, kiểm tra dữ liệu được nhập vào, gửi yêu cầu xử lý xuống tầng kế tiếp.
- Layer Business Logic :
  - Đây là layer để xử lý các dữ liệu, thông tin trước khi đưa lên giao diện hoặc đưa xuống dữ liệu.
  - Đây là nơi để kiểm tra các yêu cầu nghiệp vụ, tính toán các yêu cầu nghiệp vụ.
  - Tại đây các tính năng tính toán trong chương trình sẽ được thực thi. Ví dụ: tính lương theo một công thức.
- Layer Data Access: Layer này sẽ lo nhiệm vụ là đọc cơ sở dữ liệu lên, cập nhật cơ sở dữ liệu, update cơ sở dữ liệu.

## 2 Mô tả CSDL



### 3 Xây dựng DTO – Data Transfer Object

#### 3.1 Tạo mới Project dạng “Class Library”

Thực hiện ánh xạ 1-1 (1 bảng trong CSDL thành 01 lớp DTO tương ứng).

#### 3.2 Lớp Lop

```
namespace QLHSDTO
{
    public class LopHoc
    {
        public string MaLop { get; set; }
        public string TenLop { get; set; }
        public string SiSo { get; set; }
        public List<HocSinh> dsHocSinh { get; set; }
    }
}
```

#### 3.3 Lớp HocSinh

```
namespace QLHSDTO
{
    public class HocSinh
    {
        public string MaHS { get; set; }
        public string TenHS { get; set; }
        public DateTime NgaySinh { get; set; }
        public string DiaChi { get; set; }
        public double DTB { get; set; }
        public string MaLop { get; set; }
    }
}
```

#### 3.4 Biên dịch project thành dll

### 4 Xây dựng lớp DAL (Data Access Layer – tên khác DAO)

Ứng với một lớp DTO ta xây dựng một lớp DAO tương ứng để thao tác trực tiếp với CSDL. Bao gồm: Thêm, Xóa, Sửa, Tìm kiếm, lấy Danh sách.

Để làm việc cần References các thư viện đã tạo ở trên.

#### 4.1 Class LopHocDAO

```

namespace QLHSDAO
{
    public class LopHocDAO
    {
        public static List<LopHoc> layDSLop()
        {
            List<LopHoc> dsLop = new List<LopHoc>();
            try
            {
                DataTable dtLop =
                DataProvider.TruyVan_LayDuLieu("SELECT * FROM Lop");
                foreach (DataRow r in dtLop.Rows)
                {
                    LopHoc lh = new LopHoc
                    {
                        MaLop = r["MaLop"].ToString(),
                        TenLop = r["TenLop"].ToString(),
                        SiSo = int.Parse(r["SiSo"].ToString())
                    };
                    dsLop.Add(lh);
                }
            }
            catch
            {
                dsLop = new List<LopHoc>();
            }
            return dsLop;
        }
    }
}

```

#### 4.2 Class HocSinhDAO

```

namespace QLHSDAO
{
    //Viết các hàm liên quan tới đối tượng HocSinh bao gồm Thêm,
    Xóa, Sửa, Lấy, tìm kiếm
    public class HocSinhDAO
    {
        public static List<HocSinh> layDSHocSinh()
        {
            List<HocSinh> dsHS = new List<HocSinh>();

```

```

        DataTable dtHS =
        DataProvider.TruyVan_LayDuLieu("SELECT * FROM HocSinh");
        foreach(DataRow r in dtHS.Rows)
        {
            HocSinh hs = new HocSinh
            {
                MaHS = r["MaHS"].ToString(),
                TenHS = r["TenHS"].ToString(),
                DiaChi = r["DiaChi"].ToString(),
                MaLop = r["MaLop"].ToString()
            };
            if (r["NgaySinh"].ToString() != "")
                hs.NgaySinh =
                DateTime.Parse(r["NgaySinh"].ToString());
            if (r["DTB"].ToString() != "")
                hs.DTB = double.Parse(r["DTB"].ToString());
            dsHS.Add(hs);
        }
        return dsHS;
    }

    /// <summary>
    /// Thêm 1 học sinh mới
    /// </summary>
    /// <param name="hs">Thông tin học sinh</param>
    public static void themHocSinh(HocSinh hs){
        string sql = "INSERT INTO HocSinh(MaHS, TenHS,
        NgaySinh, DiaChi, DTB, MaLop) VALUES(@MaHS, @TenHS, @NgaySinh,
        @DiaChi, @DTB, @MaLop)";
        SqlParameter[] pa = new SqlParameter[6];
        pa[0] = new SqlParameter("MaHS", hs.MaHS);
        pa[1] = new SqlParameter("TenHS", hs.TenHS);
        pa[2] = new SqlParameter("NgaySinh",
        SqlDbType.DateTime);
        pa[2].Value = hs.NgaySinh;
        pa[3] = new SqlParameter("DiaChi", hs.DiaChi);
        pa[4] = new SqlParameter("DTB", SqlDbType.Real);
        pa[4].Value = hs.DTB;
        pa[5] = new SqlParameter("MaLop", hs.MaLop);
        DataProvider.ExcuteNonQuery(sql, CommandType.Text,
        pa);
    }
}

```

```

public static void xoaHocSinh(string MaHS) {
    string sql = "DELETE FROM HocSinh WHERE MaHS =
@MaHS";

    SqlParameter[] pa = new SqlParameter[1];
    pa[0] = new SqlParameter("MaHS", MaHS);
    DataProvider.ExcuteNonQuery(sql, CommandType.Text,
pa);
}

public static void suaHocSinh(HocSinh hs) {
    string sql = "UPDATE HocSinh SET TenHS=@TenHS,
NgaySinh=@NgaySinh, DiaChi=@DiaChi, DTB=@DTB, MaLop=@MaLop WHERE
MaHS = @MaHS";

    SqlParameter[] pa = new SqlParameter[6];
    pa[0] = new SqlParameter("MaHS", hs.MaHS);
    pa[1] = new SqlParameter("TenHS", hs.TenHS);
    pa[2] = new SqlParameter("NgaySinh",
SqlDbType.DateTime);
    pa[2].Value = hs.NgaySinh;
    pa[3] = new SqlParameter("DiaChi", hs.DiaChi);
    pa[4] = new SqlParameter("DTB", SqlDbType.Real);
    pa[4].Value = hs.DTB;
    pa[5] = new SqlParameter("MaLop", hs.MaLop);
    DataProvider.ExcuteNonQuery(sql, CommandType.Text,
pa);
}

public static List<HocSinh> timTheoTenHS(string ten) {
    List<HocSinh> dsHS = new List<HocSinh>();
    SqlParameter[] pa = new SqlParameter[1];
    pa[0] = new SqlParameter("Ten", ten);
    DataTable dtHS = DataProvider.SelectData("SELECT *
FROM HocSinh WHERE TenHS LIKE N'%' + @Ten + '%'",
CommandType.Text, pa);
    foreach (DataRow r in dtHS.Rows)
    {
        HocSinh hs = new HocSinh
        {
            MaHS = r["MaHS"].ToString(),
            TenHS = r["TenHS"].ToString(),
            DiaChi = r["DiaChi"].ToString(),

```

```

        MaLop = r["MaLop"].ToString()
    };
    if (r["NgaySinh"].ToString() != "")
        hs.NgaySinh =
DateTime.Parse(r["NgaySinh"].ToString());
    if (r["DTB"].ToString() != "")
        hs.DTB = double.Parse(r["DTB"].ToString());
    dsHS.Add(hs);
}
return dsHS;
}

public static List<HocSinh> layDSHSTheoLop(string malop)
{
    List<HocSinh> dsHS = new List<HocSinh>();
    SqlParameter[] pa = new SqlParameter[1];
    pa[0] = new SqlParameter("MaLop", malop);
    DataTable dtHS = DataProvider.SelectData("SELECT *
FROM HocSinh WHERE MaLop = @MaLop", CommandType.Text, pa);
    foreach (DataRow r in dtHS.Rows)
    {
        HocSinh hs = new HocSinh
        {
            MaHS = r["MaHS"].ToString(),
            TenHS = r["TenHS"].ToString(),
            DiaChi = r["DiaChi"].ToString(),
            MaLop = r["MaLop"].ToString()
        };
        if (r["NgaySinh"].ToString() != "")
            hs.NgaySinh =
DateTime.Parse(r["NgaySinh"].ToString());
        if (r["DTB"].ToString() != "")
            hs.DTB = double.Parse(r["DTB"].ToString());
        dsHS.Add(hs);
    }
    return dsHS;
}
}
}
}
}
}
}
}
}
}
}
}

```

#### 4.3 Lớp DataProvider kết nối dữ liệu

```
public class DataProvider
```

```
{
    public static string ChuoiKetNoi
    {
        get { return
ConfigurationManager.ConnectionStrings["QLHS"].ConnectionString;
}
    }
    public static bool TruyVan_XuLy(string sql)
    {
        try
        {
            SqlConnection con = new
SqlConnection(ChuoiKetNoi);
            SqlCommand cmd = new SqlCommand(sql, con);
            cmd.Connection.Open();
            cmd.ExecuteNonQuery();
            cmd.Connection.Close();
            return true;
        }
        catch
        {
            return false;
        }
    }

    public static DataTable TruyVan_LayDuLieu(string sql)
    {
        SqlConnection con = new SqlConnection(ChuoiKetNoi);
        SqlDataAdapter da = new SqlDataAdapter(sql, con);
        DataTable kq = new DataTable();
        da.Fill(kq);
        return kq;
    }

    public static void ExcuteNonQuery(string sql, CommandType
type, SqlParameter[] paras)
    {
        SqlConnection sqlcon = new
SqlConnection(ChuoiKetNoi);
        sqlcon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = sqlcon;
    }
}
```



```
cmd.CommandText = sql;
cmd.CommandType = type;

if (paras != null)//có tham số
    cmd.Parameters.AddRange(paras);

cmd.ExecuteNonQuery();

sqlcon.Close();
}

public static DataTable SelectData(string sql,
CommandType type, SqlParameter[] paras)
{
    DataTable kq = new DataTable();
    SqlConnection sqlcon = new
SqlConnection(ChuoiketNoi);
sqlcon.Open();

SqlCommand cmd = new SqlCommand();
cmd.Connection = sqlcon;
cmd.CommandText = sql;
cmd.CommandType = type;

if (paras != null)
    cmd.Parameters.AddRange(paras);

SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(kq);

sqlcon.Close();
return kq;
}

public static DataSet SelectMultiData(string sql)
{
    DataSet kq = new DataSet();
    SqlConnection sqlcon = new
SqlConnection(ChuoiketNoi);
sqlcon.Open();

SqlDataAdapter da = new SqlDataAdapter(sql, sqlcon);
da.Fill(kq);
```

```

        sqlcon.Close();
        return kq;
    }
}

```

## 5 Xây dựng lớp BUS (Business Layer)

Để làm việc cần References các thư viện đã tạo ở trên.

### 5.1 LopHocBUS

```

namespace QLHSBUS
{
    public class LopHocBUS
    {
        public static List<LopHoc> LayDSLop()
        {
            return LopHocDAO.layDSLop();
        }
    }
}

```

### 5.2 HocSinhBUS

```

namespace QLHSBUS
{
    public class HocSinhBUS
    {
        public static bool themHocSinh(HocSinh hs)
        {
            //kiểm tra ràng buộc tự nhiên
            //RBTN1: tên phải có
            if (hs.TenHS == "") return false;
            //kiểm tra ràng buộc nghiệp vụ
            //RBNV1 : tuổi tối thiểu 15
            int tuoi = DateTime.Now.Year - hs.NgaySinh.Year;
            if (tuoi < 15) return false;
            //thực hiện thêm
            try
            {
                HocSinhDAO.themHocSinh(hs);
                return true;
            }
        }
    }
}

```

```
    }
    catch
    {
        return false;
    }
}

public static bool xoaHocSinh(string mahs)
{
    try
    {
        HocSinhDAO.xoaHocSinh(mahs);
        return true;
    }
    catch
    {
        return false;
    }
}

public static bool suaHocSinh(HocSinh hs)
{
    //kiểm tra ràng buộc tự nhiên
    //RBTN1: tên phải có
    if (hs.TenHS == "") return false;
    //kiểm tra ràng buộc nghiệp vụ
    //RBNV1 : tuổi tối thiểu 15
    int tuoi = DateTime.Now.Year - hs.NgaySinh.Year;
    if (tuoi < 15) return false;
    //thực hiện thêm
    try
    {
        HocSinhDAO.suaHocSinh(hs);
        return true;
    }
    catch
    {
        return false;
    }
}

public static List<HocSinh> layDSHocSinh()
```

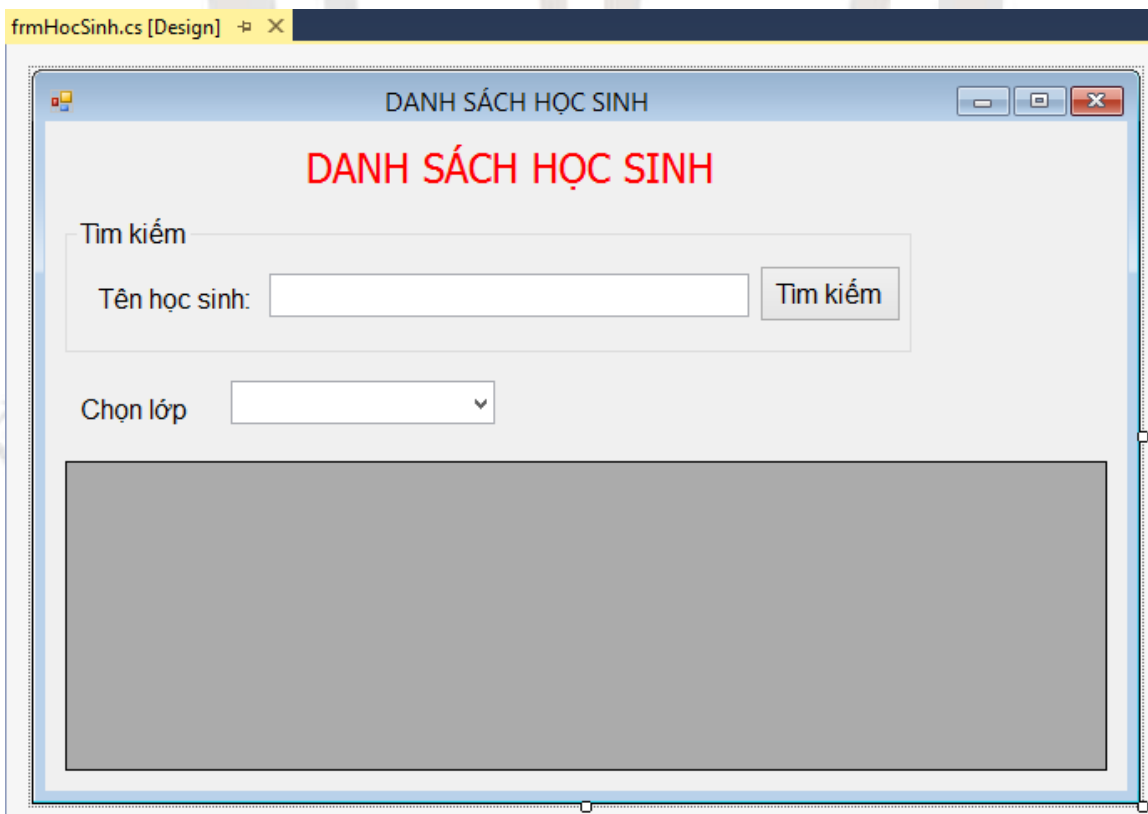
```
{
    return HocSinhDAO.layDSHocSinh();
}
public static List<HocSinh> layDSHocSinhTheoLop(string
malop)
{
    return HocSinhDAO.layDSHSTheoLop(malop);
}
public static List<HocSinh> timHocSinhTheoTen(string ten)
{
    return HocSinhDAO.timTheoTenHS(ten);
}
}
}
```

## 6 Lớp GUI (còn gọi là Presentation)

Xây dựng giao diện Windows Form để gọi lớp xử lý nghiệp vụ BUS.

Để làm việc cần References các thư viện đã tạo ở trên.

### 6.1 Xây dựng giao diện



## 6.2 Chỉnh sửa tập tin App.config

Thêm vào chuỗi kết nối

```
<connectionStrings>
  <add name="QLHS" connectionString="Server=.;
Database=SQLHocSinh; Integrated Security=True;" />
</connectionStrings>
```

## 6.3 Các sự kiện trên form

```
private void frmHocSinh_Load(object sender, EventArgs e)
{
    //load ds lớp
    cboLop.DataSource = LopHocBUS.LayDSLop();
    cboLop.DisplayMember = "TenLop";
    cboLop.ValueMember = "MaLop";
    cboLop.SelectedIndexChanged += new
EventHanlder(cboLop_SelectedIndexChanged);

    loadDSHocSinh();
    dgvHocSinh.AutoSizeColumns();
}

private void cboLop_SelectedIndexChanged(object sender,
EventArgs e)
{
    string malop = cboLop.SelectedValue.ToString();
    dgvHocSinh.DataSource =
HocSinhBUS.layDSHocSinhTheoLop(malop);
    dgvHocSinh.AutoSizeColumns();
}

private void loadDSHocSinh()
{
    dgvHocSinh.DataSource = HocSinhBUS.layDSHocSinh();
}

private void btnTimKiem_Click(object sender, EventArgs e)
{
    dgvHocSinh.DataSource =
HocSinhBUS.timHocSinhTheoTen(txtTenHSSearch.Text.Trim());
    dgvHocSinh.AutoSizeColumns();
}
```