

Bài tập thực hành 06:

THỰC HÀNH VỀ TRIỂN KHAI HỆ THỐNG

Mục tiêu

- ✓ Ánh xạ được tài liệu phân tích và thiết kế thành triển khai hệ thống
- ✓ Ánh xạ được các thành phần thành các gói trong hệ thống
- ✓ Ánh xạ được các Node thành các subSystem.
- ✓ Cài đặt được các lớp xử lý thực hiện từng UC trong hệ thống, dựa vào biểu đồ lớp thiết kế cho từng UC, biểu đồ tuần tự, biểu đồ triển khai.

Bài tập 1: Mô tả chức năng và yêu cầu của HỆ THỐNG QUẢN LÝ THƯ VIỆN

Xây dựng hệ thống quản lý thư viện cho trường Đại học gồm các hoạt động quản lý thông tin sách, quản lý thông tin độc giả, quản lý hoạt động mượn trả sách:

- ✓ Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.
- ✓ Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngaynhap). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.
- ✓ Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).
- ✓ Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).

- ✓ Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.
- ✓ Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.
- ✓ Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.
- ✓ Người dùng hệ thống phải đăng nhập trước khi thực hiện.

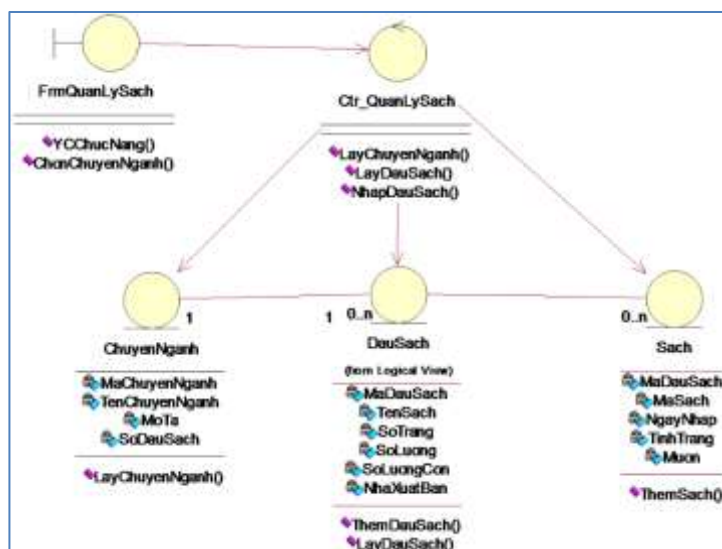
Yêu cầu: Sử dụng ngôn ngữ lập trình C# viết mã lệnh triển khai cho UC QuanLySach.

Hướng dẫn thực hiện:

Phân tích:

- ✓ Sau khi phân tích xong các UC phân rã của UC QuanLySach, gộp tất cả các lớp của các UC phân rã vào một biểu đồ.
- ✓ Căn cứ vào biểu đồ triển khai, xây dựng các lớp trong gói bị phụ thuộc trước.
- ✓ Với mỗi lớp khai báo đối tượng sử dụng, xây dựng các phương thức
- ✓ Xây dựng mã lệnh dựa vào biểu đồ tuần tự và biểu đồ hoạt động của UC, phương thức.

Minh họa: Ví dụ đây là biểu đồ lớp UC QuanLySach, sau khi phân tích và thiết kế UC phân rã là Thêm đầu sách mới gồm có các lớp, và các phương thức như sau:



Chúng ta thấy các lớp Entity được ánh xạ thành các lớp tương tác với DataBase thực hiện các hành động tương tác với cơ sở dữ liệu.

Do tất cả các lớp tương tác với DataBase đều thực thi các thao tác kết nối, thêm, sửa, xóa với cơ sở dữ liệu nên nghiệp vụ này sẽ được tách ra thành một lớp Cls_Data chuyên thực hiện các thao tác với CSDL. Và các lớp đảm nhận thực hiện thao tác dữ liệu sẽ gọi các phương thức của lớp này.

Như vậy mã lệnh của ứng dụng để thực hiện UC QuanLySach và UC phân rã Thêm đầu sách mới như sau:

a) Lớp DataHelper: Được dùng chung, gồm các phương thức chung thao tác với Database.

```
Namespace QuanLyThuVien.DataAccess
{
    class DataHelper
    {
        DataSet DS = new DataSet();
        static string ChuoiKetNoi;
        SqlConnection con;
        public DataHelper(string SeverName, string DataName,
                           string UserName, string Pw)
        {
            ChuoiKetNoi = "server=" + SeverName + "; database=" +
                DataName + "; Uid= " + UserName + "; pwd=" + Pw;
            con = new SqlConnection(ChuoiKetNoi);
        }
        public DataHelper(string chuoiketnoi)
        {
            ChuoiKetNoi = chuoiketnoi;
            con = new SqlConnection(chuoiketnoi);
        }
        public DataHelper()
        { }
        public DataHelper(string SeverName, string DataName)
        {
            ChuoiKetNoi = "server=" + SeverName + "; database=" +
                DataName + ";Integrated security=SSPI"; //"data source= "
                + SeverName + "; Initial Catalog=" + DataName + ";
                Integrated security=SSPI";
            con = new SqlConnection(ChuoiKetNoi);
        }
    }
}
```

```
public DataTable FillDataTable(string strSQL)
{
    SqlDataAdapter DA = new SqlDataAdapter(strSQL,
                                        ChuoiKetNoi);

    DataTable dt = new DataTable();
    DA.Fill(dt);
    return dt;
}
/// <summary>
/// Dùng để cập nhật dữ liệu từ một bảng trong DataSet lên Cơ sở dữ liệu
/// </summary>
/// <param name="tenbang1"> Bảng trong cơ sở dữ liệu</param>
/// <param name="tenbang2"> Bảng trong DataSet</param>
public void UpdateFromDataTableToDataBase(string tenBangDB,
                                        DataTable dt)
{
    //try
    //{
        SqlDataAdapter da = new SqlDataAdapter("Select * from " +
        tenBangDB, ChuoiKetNoi);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        da.Update(dt);
    //}
    //catch (Exception ex)
    //{
        // throw new Exception(ex.Message);
    //}
}
public void InsertRowtable(DataTable dt, params object[] Value)
{
    // Tạo ra một dòng có cấu trúc giống bảng
    DataRow dr = dt.NewRow();
    try
    {
        for (int i = 0; i < Value.Length; i++)
        {
            dr[i] = Value[i]; // Gán giá trị cho các trường
        }
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

```
    }
    dt.Rows.Add(dr);
}
/// <summary>
/// Sử dụng để xóa các dòng của một bảng trong DataSet
/// </summary>
/// <param name="TableName">Tên bảng</param>
/// <param name="dieukien">Điều kiện của các dòng cần xóa</param>
public Boolean DeleteRowTable(DataTable dt, string dieukien)
{
    try
    {
        DataView dv = new DataView(dt);
        dv.RowFilter = dieukien;
        dv.AllowDelete = true;
        int i = 0;
        bool kt = false;
        while (i < dv.ToTable().Rows.Count)
        {
            dv[i].Delete(); // Tại sao không phải tăng chỉ số i
            kt = true;
        }
        return kt;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

public void UpdateRowTable(DataTable dt, string dieukien, params
                                                                    object[]
                                                                    field_GiaTri)
{
    DataView dv = new DataView(dt);
    dv.RowFilter = dieukien;
    for (int i = 0; i < dv.ToTable().Rows.Count; i++)
    {
        for (int j = 0; j < field_GiaTri.Length; j += 2)
        {
            dv[i][field_GiaTri[j].ToString()] = field_GiaTri[j + 1];
        }
    }
}
```

```

    }
}
}

```

b) Cài đặt lớp ChuyenNganh.

```

namespace QuanLyThuVien.DataAccess
{
    class ChuyenNganhDL
    {
        DataTable ChuyenNganh;
        DataHelper dt = new DataAccess.DataHelper();
        public DataTable LayChuyenNganh()
        {
            ChuyenNganh = dt.FillDataTable("Select * from ChuyenNganh");
            ChuyenNganh;
        }
    }
}

```

c) Cài đặt lớp Ctr_QuanLySach

```

namespace QuanLyThuVien.Bussiness
{
    class Ctr_QuanLySach
    {
        ChuyenNganhDL cn = new DataAccess.ChuyenNganhDL();
        public DataTable LayChuyenNganh()
        {
            return cn.LayChuyenNganh();
        }
    }
}

```

d) Cài đặt lớp Frm_QuanlySach

```

namespace QuanLyThuVien.Bound
{
    public partial class frmQuanLySach : Form
    {
        Ctr_QuanLySach ql = new Ctr_QuanLySach();
        public frmQuanLySach()
        {
            InitializeComponent();
        }
    }
}

```

```
    }  
    private void frmQuanLySach_Load(object sender, EventArgs e)  
    {  
        cboChuyenNganh.DataSource = ql.LayChuyenNganh();  
        cboChuyenNganh.DisplayMember = "TenChuyenNganh";  
        cboChuyenNganh.ValueMember = "ChuyenNganhID";  
        dataGridView1.DataSource =  
            ql.LayDauSach(cboChuyenNganh.SelectedValue.ToString());  
    }  
}
```

Bài tập tự làm

1. Hoàn thiện mã lệnh cho các lớp DauSachDL, SachDL
2. Hoàn thiện mã lệnh của lớp Ctr_QuanLySach, Frm_QuanLySach
3. Hoàn thành mã lệnh cho các UC khác.

-----**Hết Lab 06**-----