

Bài tập tuần 8

Đăng ký học phần

Thông tin lớp (List View)

Mã lớp	Khóa	Loại
18CNTT1	2018	CQ
19TTH1	2019	CQ
19TTH2	2019	TN

Danh sách sinh viên của lớp (DataGridView)

	Mã SV	Họ tên	Ngày sinh	Phái	Lớp	Điểm TB
▶	18120120	Phạm Hữu Hào	08/06/2000	Nam	18CNTT1	7.5
	18120460	Nguyễn Ngọc Lan	14/02/2000	Nữ	18CNTT1	

Kết quả ĐKHP (DataGridView)

	Chọn	Năm học	Học kỳ	Mã sinh viên	Mã môn học	Số TC	Địa điểm	Điểm
▶	<input checked="" type="checkbox"/>	19-20	1	18120120	BAA00012	3.0	LT	
	<input type="checkbox"/>	19-20	1	18120120	CSC10001	4.0	NVC	
	<input type="checkbox"/>	19-20	1	18120120	MTH00086	3.0	NVC	
	<input checked="" type="checkbox"/>	19-20	1	18120120				

Thêm

Xóa

Cập nhật thay đổi

Thoát

DTO

DtoGoods.cs (sản phẩm)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.DTO
{
    public class DtoGoods
    {
        public string ID;
        public string Name;
        public string Unit;
        public string Origin;
    }
}

```

DtoInvoice.cs (hóa đơn)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.DTO
{
    public class DtoInvoice
    {
        public string ID;
        public string NameCustomer;
        public string Address;
        public string Phone;
        public int Total;
        public DateTime Date;
    }
}

```

GUI

frmKiemTraDonHang.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace _1760273
{
    public partial class frmOrderManagement : Form
    {
        public frmOrderManagement()
        {
            InitializeComponent();
        }

        private void frmOrderManagement_Load(object sender, EventArgs e)
        {
            LoadListOrigins();
            LoadListGoodsOrigin(cbMadeIn.SelectedValue.ToString());
            LoadDgvInvoice();
        }

        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void cbInvoiceID_SelectedIndexChanged(object sender, EventArgs e)
        {
            LoadListInvoicesGoodsID(cbInvoiceID.SelectedValue.ToString());
            dgvInvoices.DataSource = dtInvoices;
            CalcTotalInvoice(cbInvoiceID.SelectedValue.ToString());
        }

        private void cbMadeIn_SelectedIndexChanged(object sender, EventArgs e)
        {
            LoadListGoodsOrigin(cbMadeIn.SelectedValue.ToString());
        }

        private void dgvInvoices_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            curInvoiceID = dgvInvoices.Rows[dgvInvoices.CurrentCell.RowIndex].Cells[0].Value.ToString();
            LoadListInvoicesDetailsByInvoiceID(curInvoiceID);
            dgvInvoiceDetail.DataSource = dtInvoicesDetails;
            LoadDgvInvoiceDetail();
        }
    }
}

```

```
private void dgvInvoices_MouseUp(object sender, MouseEventArgs e)
{
    dgvInvoices.CurrentCell = null;
}
}
```

parKiemTraDonHang.cs

```

using _1760273.BUS;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace _1760273
{
    public partial class frmOrderManagement : Form
    {
        static BusGoods busGoods = new BusGoods();
        static BusInvoice busInvoice = new BusInvoice();
        static DataTable dtInvoices = new DataTable();
        static DataTable dtInvoicesDetails = new DataTable();
        static string curInvoiceID = "";

        private void LoadListOrigins()
        {
            cbMadeIn.DataSource = busGoods.LoadListOrigins();
            cbMadeIn.DisplayMember = "xuatxu";
            cbMadeIn.ValueMember = "xuatxu";
        }

        private void LoadListGoodsOrigin(string origin)
        {
            cbInvoiceID.DataSource = busGoods.LoadListGoodsOrigin(origin);
            cbInvoiceID.DisplayMember = "tenhh";
            cbInvoiceID.ValueMember = "mahh";
        }

        private void LoadListInvoicesGoodsID(string goodsID)
        {
            dtInvoices = busInvoice.LoadListInvoicesGoodsID(goodsID);
        }

        private void CalcTotalInvoice(string goodsID)
        {
            txtbTotalInvoice.Text = string.Format("{0:0,0} VND", busInvoice.CalcTotalInvoice(goodsID));
        }

        private void LoadDgvInvoice()
        {
            dgvInvoices.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
            dgvInvoices.AllowUserToAddRows = false;
            dgvInvoices.ReadOnly = true;
        }
    }
}

```

```

        dgvInvoices.Columns[0].HeaderText = "Mã HĐ";
        dgvInvoices.Columns[1].HeaderText = "Tên khách hàng";
        dgvInvoices.Columns[2].HeaderText = "Địa chỉ";
        dgvInvoices.Columns[3].HeaderText = "SĐT";
        dgvInvoices.Columns[4].HeaderText = "Tổng tiền";
        dgvInvoices.Columns[5].HeaderText = "Ngày";

        dgvInvoices.Columns[0].Width = 100;
        dgvInvoices.Columns[1].Width = 200;
        dgvInvoices.Columns[2].Width = 300;
        dgvInvoices.Columns[3].Width = 150;
        dgvInvoices.Columns[4].Width = 150;
        dgvInvoices.Columns[5].Width = 100;
    }

    private void LoadDgvInvoiceDetail()
    {
        dgvInvoiceDetail.AllowUserToAddRows = false;
        dgvInvoiceDetail.ReadOnly = true;

        dgvInvoiceDetail.Columns[0].HeaderText = "Mã HĐ";
        dgvInvoiceDetail.Columns[1].HeaderText = "Mã hàng hóa";
        dgvInvoiceDetail.Columns[2].HeaderText = "Tên hàng hóa";
        dgvInvoiceDetail.Columns[3].HeaderText = "Đơn vị tính";
        dgvInvoiceDetail.Columns[4].HeaderText = "Xuất xứ";
        dgvInvoiceDetail.Columns[5].HeaderText = "Số lượng";
        dgvInvoiceDetail.Columns[6].HeaderText = "Đơn giá";

        dgvInvoiceDetail.Columns[0].Width = 100;
        dgvInvoiceDetail.Columns[1].Width = 200;
        dgvInvoiceDetail.Columns[2].Width = 300;
        dgvInvoiceDetail.Columns[3].Width = 130;
        dgvInvoiceDetail.Columns[4].Width = 100;
        dgvInvoiceDetail.Columns[5].Width = 80;
        dgvInvoiceDetail.Columns[6].Width = 80;
    }

    private void LoadListInvoicesDetailsByInvoiceID(string invoiceID)
    {
        dtInvoicesDetails = busInvoice.LoadListInvoicesDetailsByInvoiceID(invoiceID);
    }
}

```

BUS

BusGoods.cs

```
using _1760273.DAO;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.BUS
{
    public class BusGoods
    {
        static DaoGoods daoGoods = new DaoGoods();

        public DataTable LoadListOrigins()
        {
            return daoGoods.LoadListOrigins();
        }

        public DataTable LoadListGoodsOrigin(string origin)
        {
            return daoGoods.LoadListGoodsOrigin(origin);
        }
    }
}
```

BusInvoice.cs

```

using _1760273.DAO;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.BUS
{
    public class BusInvoice
    {
        static DaoInvoice daoInvoice = new DaoInvoice();

        public DataTable LoadListInvoicesGoodsID(string goodsID)
        {
            return daoInvoice.LoadListInvoicesGoodsID(goodsID);
        }

        public int CalcTotalInvoice(string goodsID)
        {
            return daoInvoice.CalcTotalInvoice(goodsID);
        }

        public DataTable LoadListInvoicesDetailsByInvoiceID(string invoiceID)
        {
            return daoInvoice.LoadListInvoicesDetailsByInvoiceID(invoiceID);
        }
    }
}

```

DAL

DalDB.cs


```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.DAO
{
    public class DaoDB
    {
        SqlConnection sqlConnection = new SqlConnection(
            @"Data Source=LAPTOP-LN7FVV9Q\SQLEXPRESS;Initial Catalog=BANHANG;Integrated Security=1
        public void OpenConnection()
        {
            if (sqlConnection.State == ConnectionState.Closed)
            {
                sqlConnection.Open();
            }
        }

        public void CloseConnection()
        {
            if (sqlConnection.State == ConnectionState.Open)
            {
                sqlConnection.Close();
            }
        }

        public DataTable GetData(string sql)
        {
            var datatable = new DataTable();
            var cmd = new SqlCommand(sql, sqlConnection);
            var adp = new SqlDataAdapter(cmd);

            cmd.CommandType = CommandType.StoredProcedure;
            adp.Fill(datatable);

            return datatable;
        }

        public DataTable GetDataWithParameters(string sql, SqlParameter[] sqlParameters)
        {
            var datatable = new DataTable();
            var cmd = new SqlCommand(sql, sqlConnection);
            var adp = new SqlDataAdapter(cmd);

            cmd.CommandType = CommandType.StoredProcedure;
            cmd.Parameters.AddRange((sqlParameters));
            adp.Fill(datatable);
        }
    }
}

```

```

        return datatable;
    }

    public int ExecuteStoredProcedure(string sql, SqlParameter[] sqlParameters)
    {
        int affRows = 0;
        SqlCommand cmd = new SqlCommand(sql, sqlConnection);

        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddRange(sqlParameters);

        try
        {
            OpenConnection();
            affRows = cmd.ExecuteNonQuery();
            CloseConnection();
        }
        catch
        {
            affRows = 0;
        }

        return affRows;
    }

    public DataTable ExecuteFunction(string sql, SqlParameter[] sqlParameters)
    {
        DataTable res = new DataTable();
        SqlCommand cmd = new SqlCommand(sql, sqlConnection);
        SqlDataAdapter adt = new SqlDataAdapter(cmd);

        cmd.Parameters.AddRange(sqlParameters);
        adt.Fill(res);

        return res;
    }

    //public void TotalCupom(int cupom)
    //{
    //    float SAIDA;
    //    SqlDataAdapter da2 = new SqlDataAdapter();
    //    if (conex1.State == ConnectionState.Closed)
    //    {
    //        conex1.Open();
    //    }
    //    SqlCommand Totalf = new SqlCommand("SELECT dbo.Tcupom(@code)", conex1);
    //    SqlParameter code1 = new SqlParameter("@code", SqlDbType.Int);
    //    code1.Value = cupom;
    //    SAIDA = Totalf.ExecuteScalar();
    //}

```

```
        //      return SAIDA;
        //}

    }
}
```

DaoGoods.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.DAO
{
    public class DaoGoods
    {
        static DaoDB daoDB = new DaoDB();

        public DataTable LoadListOrigins()
        {
            return daoDB.GetData("sLoadListOrigins");
        }

        public DataTable LoadListGoodsOrigin(string origin)
        {
            List<SqlParameter> paras = new List<SqlParameter>();
            paras.Add(new SqlParameter("@iOrigin", origin));

            return daoDB.GetDataWithParameters("sLoadListGoodsOrigin", paras.ToArray());
        }
    }
}
```

DaoInvoice.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _1760273.DAO
{
    public class DaoInvoice
    {
        static DaoDB daoDB = new DaoDB();

        public DataTable LoadListInvoicesGoodsID(string goodsID)
        {
            List<SqlParameter> paras = new List<SqlParameter>();
            paras.Add(new SqlParameter("@iGoodsID", goodsID));

            return daoDB.GetDataWithParameters("sLoadListInvoicesGoodsID", paras.ToArray());
        }

        public int CalcTotalInvoice(string goodsID)
        {
            int rs = 0;
            List<SqlParameter> paras = new List<SqlParameter>();
            paras.Add(new SqlParameter("@iGoodsID", goodsID));
            DataTable res = daoDB.ExecuteFunction(
                "select dbo.fCalcTotalInvoice(@iGoodsID) as res", paras.ToArray());

            try
            {
                rs = int.Parse(res.Rows[0][0].ToString());
            }
            catch
            {
                rs = 0;
            }

            return rs;
        }

        public DataTable LoadListInvoicesDetailsByInvoiceID(string invoiceID)
        {
            List<SqlParameter> paras = new List<SqlParameter>();
            paras.Add(new SqlParameter("@iInvoiceID", invoiceID));

            return daoDB.GetDataWithParameters("sLoadListInvoicesDetailsByInvoiceID", paras.ToArray());
        }
    }
}

```



SQL

method.sql

```

use BANHANG
go

-- load list of origins
if OBJECT_ID('sLoadListOrigins', 'P') is not null
    drop proc sLoadListOrigins
go
create proc sLoadListOrigins
as
    select distinct xuatu from HANGHOA
go

-- load list of goods base on origin
if OBJECT_ID('sLoadListGoodsOrigin', 'P') is not null
    drop proc sLoadListGoodsOrigin
go
create proc sLoadListGoodsOrigin @iOrigin nvarchar(15)
as
    select hh.mahh, hh.tenhh
    from HANGHOA hh
    where hh.xuatu = @iOrigin
go

-- load list of invoices base on goodsID
if OBJECT_ID('sLoadListInvoicesGoodsID', 'P') is not null
    drop proc sLoadListInvoicesGoodsID
go
create proc sLoadListInvoicesGoodsID @iGoodsID nvarchar(15)
as
    select hd.*
    from HOADON hd join CTHOADON ct on ct.mahd = hd.mahd
    where ct.mahh = @iGoodsID
go

-- calculate total money of a invoice base on ID
if OBJECT_ID('fCalcTotalInvoice') is not null
    drop function fCalcTotalInvoice
go
create function fCalcTotalInvoice(@iGoodsID nvarchar(15)) returns int
as
begin
    return (select sum(ct.sl * ct.dongia)
            from HOADON hd join CTHOADON ct on ct.mahd = hd.mahd
            where ct.mahh = @iGoodsID)
end
go

-- load list of invoice detail base on invoiceID
if OBJECT_ID('sLoadListInvoicesDetailsByInvoiceID', 'P') is not null
    drop proc sLoadListInvoicesDetailsByInvoiceID
go

```

```
create proc sLoadListInvoicesDetailsByInvoiceID @iInvoiceID nvarchar(15)
as
    select ct.mahd, ct.mahh, hh.tenhh, hh.dvt, hh.xuatxu, ct.sl, ct.dongia
    from HANGHOA hh join CTHOADON ct on ct.mahh = hh.mahh
    where ct.mahd = @iInvoiceID
go
```