

**Console:**

**Một số hàm căn bản:**

Chương trình chính:

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Nội dung cần xuất");

string s = Console.ReadLine();

int i = int.Parse(Console.ReadLine());

//hàm liên quan ngày tháng

DateTime d = DateTime.Now;

string da = "2022-05-05";

DateTime t = DateTime.Parse(da);

}

}

Hàm ko xuất giá trị:

public static void XuatPhanSo(PhanSo P)

{ }

Hàm xuất giá trị:

public static PhanSo TinhTong(PhanSo A, PhanSo B)

{return TaoPhanSo(tuso, mauso);}

Tạo struct dữ liệu:

public struct PhanSo

{

public int TuSo;

public int MauSo;

}

**Hàm đọc file, lưu file:**

* Căn bản:

public static void Luu(PhanSo P)

{

StreamWriter file = new StreamWriter("D:\\phanso.txt");

file.Write("{0}/{1}", P.TuSo, P.MauSo);

file.Close();

}

public static List<PhanSo> Doc()

{

List<PhanSo> L = new List<PhanSo>();

StreamReader file = new StreamReader("D:\\phanso.txt");

while (file.EndOfStream == false)

{

string data = file.ReadLine();

string[] s = data.Split("/");

PhanSo P = TaoPhanSo(int.Parse(s[0]), int.Parse(s[1]));

L.Add(P);

}

file.Close();

return L;

}

* Nâng cao:

public static List<HDnhap> DocHDNhap()

{

StreamReader file = new StreamReader("wwwroot/data/hdnhap.json");

string jsonString = file.ReadToEnd();

file.Close();

List<HDnhap> danhsachHD = JsonConvert.DeserializeObject<List<HDnhap>>(jsonString);

return danhsachHD;

}

public static bool LuuDSNhap(List<HDnhap> danhsachHD)

{

StreamWriter file = new StreamWriter("wwwroot/data/hdnhap.json");

string jsonString = JsonConvert.SerializeObject(danhsachHD);

file.Write(jsonString);

file.Close();

return true;

}

**Ví dụ: giải bài tổng phân số:**

* **Chương trình chính**

static void Main(string[] args)

{

Console.WriteLine("Nhap Phan So 1: - Tu So: ");

int tuso = int.Parse(Console.ReadLine());

Console.WriteLine("- Mau So: ");

int mauso = int.Parse(Console.ReadLine());

PhanSo A = TaoPhanSo(tuso, mauso);

Console.WriteLine("Nhap Phan So 2: - Tu So: ");

tuso = int.Parse(Console.ReadLine());

Console.WriteLine("- Mau So: ");

mauso = int.Parse(Console.ReadLine());

PhanSo B = TaoPhanSo(tuso, mauso);

Console.WriteLine("Tong 2 phan so la : ");

XuatPhanSo(TinhTong(A,B));

Console.ReadLine();

}

* **Khai báo struct**

public struct PhanSo

{

public int TuSo;

public int MauSo;

}

* **Hàm xử lý nghiệp vụ**

public static PhanSo TaoPhanSo(int tuso, int mauso)

{

PhanSo P;

P.TuSo = tuso;

if (mauso != 0)

{

P.MauSo = mauso;

} else

{

while (mauso == 0)

{

Console.Write("Vui long nhap lai mau so: ");

mauso = int.Parse(Console.ReadLine());

}

P.MauSo = mauso;

}

return P;

}

public static PhanSo TinhTong(PhanSo A, PhanSo B)

{

var mauso = A.MauSo \* B.MauSo;

var tuso = A.MauSo \* B.TuSo + B.MauSo \* A.TuSo;

return TaoPhanSo(tuso, mauso);

}

public static void XuatPhanSo(PhanSo P)

{

Console.WriteLine("{0}/{1}", P.TuSo, P.MauSo);

}

**Giao diện web:**

* Trang người dùng:

@using Entities; (nếu cần)

Sử dụng dữ liệu được khai báo trong main: @Model.Chuoi, nếu trong một hàm khác: Model.Chuoi (ví dụ: @if (Model.Chuoi0{ })

Form sử dụng method POST, code vừa đủ dùng không cần quá chi tiết, chỉ cần đủ tính năng

Input type:

* date
* number
* text
* submit
* Chương trình chính:

Chứa các biến dữ liệu và xử lý cơ bản, chú trọng vào OnGet() và OnPost(), sử dụng các hàm XuLy từ Services.

Quan trọng: phải có biến để lấy dữ liệu từ trang người dùng và biến để trả lại dữ liệu để hiển thị.

* Entities: (khai báo dữ liệu): tương tự Console, không sử dụng hàm từ Services hay DAL
* Services: (hàm xử lý):

Chứa các hảm để xử lý dữ liệu, trả kết quả về chương trình chính.

Sử dụng các hàm LuuTru từ DAL để đọc, lưu dữ liệu từ file

* DAL: (lưu trữ)

Tùy vào nhu cầu đề bài sẽ có các hàm Đọc, Lưu, Thêm, Sửa file data

**Ví dụ giải bài tổng phân số:**

Phân tích: 4 biến được submit: tử số và mẫu số của 2 phân số, 1 kết quả trả về

**Trang người dùng:**

<form method="POST">

<label for="tuso1">Tử số A:</label><br />

<input type="number" name="tuso1" /><br />

<label for="mauso1">Mẫu số A:</label><br />

<input type="number" name="mauso1" /><br />

<label for="tuso2">Tử số B:</label><br />

<input type="number" name="tuso2" /><br />

<label for="mauso2">Mẫu số B:</label><br />

<input type="number" name="mauso2" /><br />

<input type="submit" value="Tinh tong"/>

</form>

@Model.Ketqua

**Chương trình chính:**

using Entities;

using Services;

public string Ketqua;

[BindProperty]

public int TuSo1 { get; set; }

[BindProperty]

public int MauSo1 { get; set; }

[BindProperty]

public int TuSo2 { get; set; }

[BindProperty]

public int MauSo2 { get; set; }

public void OnGet()

{

Ketqua = String.Empty;

}

public void OnPost()

{

try

{

PhanSo A = XuLyPhanSo.TaoPhanSo(TuSo1, MauSo1);

PhanSo B = XuLyPhanSo.TaoPhanSo(TuSo2, MauSo2);

Ketqua = "Tong 2 phan so la: " + XuLyPhanSo.XuatPhanSo(XuLyPhanSo.TongPhanSo(A, B));

}

catch (Exception ex)

{

Ketqua = ex.Message;

}

}

}

**Entities:**

public struct PhanSo

{

public int TuSo;

public int MauSo;

}

**Services:**

using Entities;

public static PhanSo TaoPhanSo(int tuso, int mauso)

{

if (mauso == 0)

{

throw new Exception("Mau so phai khac 0, vui long nhap lai");

} else

{

PhanSo P;

P.TuSo = tuso;

P.MauSo = mauso;

return P;

}

}

public static PhanSo TongPhanSo(PhanSo A, PhanSo B)

{

PhanSo P;

P.TuSo = A.TuSo \* B.MauSo + A.MauSo \* B.TuSo;

P.MauSo = A.MauSo \* B.MauSo;

return P;

}

public static string XuatPhanSo(PhanSo P)

{

return $"{P.TuSo}/{P.MauSo}";

}

}

**Ví dụ giải câu 2 đề thi:**

Phân tích: 1 biến keyword được submit, 1 List SanPham được trả về, xuất dữ liệu dưới dạng table.

Chỉ yêu cầu đọc từ file ra, giảm bớt các phần kiểm tra dữ liệu. Coi như dữ liệu không lỗi.

**Trang người dùng:**

@using Entities

<form method="POST">

<input type="text" name="keyword" alt="Tu khoa"/><br />

<input type="submit" value="Tim kiem"/>

</form>

<table class="table table-striped">

<thead>

<tr>

<th>Sản Phẩm</th>

<th>Giá</th>

<th>Năm Sản Xuất</th>

</tr>

</thead>

<tbody>

@foreach (SanPham s in Model.DSSP)

{

<tr>

<td>@s.TenSP</td>

<td>@s.Gia</td>

<td>@s.NamSX</td>

</tr>

}

</tbody>

</table>

@Model.Thongbao

**Chương trình chính:**

using Entities;

using Services;

public string Thongbao;

public List<SanPham> DSSP;

[BindProperty]

public string KeyWord { get; set; }

public void OnGet()

{

Thongbao = String.Empty;

DSSP = new List<SanPham>();

}

public void OnPost()

{

try

{

DSSP = XuLySanPham.TimKiem(KeyWord);

}

catch (Exception ex)

{

Thongbao = ex.Message;

}

}

**Entities:**

public struct SanPham

{

public string TenSP;

public int Gia;

public int NamSX;

}

**Services:**

using Entities;

using DAL;

public class XuLySanPham

{

public static List<SanPham> TimKiem(string keyword)

{

List<SanPham> DSSPfull = LuuTruSanPham.DocSanPham();

if (keyword == null)

{

return DSSPfull;

} else

{

List<SanPham> DSSP = new List<SanPham>();

foreach (SanPham s in DSSPfull)

{

if (s.TenSP.Contains(keyword))

{

DSSP.Add(s);

}

}

if (DSSP.Count > 0)

{

return DSSP;

} else

{

throw new Exception("Khong tim thay san pham");

}

}

}

}

**DAL:**

using Entities;

public static List<SanPham> DocSanPham()

{

StreamReader file = new StreamReader("wwwroot/data/sanpham.json");

string jsonString = file.ReadToEnd();

file.Close();

List<SanPham> dssanpham = JsonConvert.DeserializeObject<List<SanPham>>(jsonString);

return dssanpham;

}