

Training Skeleton

(Sử dụng Console application cho các bài 1, 2, 3, 5.1, 5.3, 5.4)

Môi trường phát triển:

- Công cụ lập trình: Microsoft Visual Studio 2015, .NET framework 4.5

- Ngôn ngữ phát triển: C#, aspx (dùng thêm CSS, Javascript)

- Cơ sở dữ liệu: SQL Server 2008R2

1. Primitive types

- Mỗi mục là 1 class, mỗi bài là 1 method.
- Mỗi method làm 1 việc độc lập. **Chú ý** Input/Process/Output của từng method (trong trường hợp cần ghép nối với các method khác)
- Không dùng LINQ

1.1. String

1

Input: 1 chuỗi.

Nếu chuỗi null -> Thông báo: **Chuỗi bị null**

Nếu chuỗi không có ký tự nào -> Thông báo: **Chuỗi rỗng**

Nếu chuỗi toàn ký tự space -> Thông báo: **Chuỗi Space**

Còn lại: In chuỗi ký tự ra màn hình

2

Như 1.

Còn lại: In chuỗi ký tự sau khi bỏ ký tự trắng ở đầu và cuối

3

Như 1.

Còn lại: In chuỗi ký tự sau khi bỏ hết ký tự trắng

4

Như 1.

Còn lại: In chuỗi ký tự sau khi chuyển thành chữ hoa các ký tự đứng sau dấu Space

5

Như 1.

Còn lại: Đếm và thông báo số ký tự Space theo mẫu: **Có [count] ký tự space**

6

Như 1.

Còn lại: Đếm và thông báo số ký tự không phải Space theo mẫu: **Có [count] ký tự không phải space**

7

Input: 2 chuỗi đầu vào (input1, input2)

Nếu 2 chuỗi giống nhau (Phân biệt hoa thường) -> Hiển thị kết quả: **Chuỗi [input1] giống chuỗi [input2]**

Ngược lại, Hiển thị kết quả: **Chuỗi "input1" khác chuỗi "input2"**

8

Như 7 nhưng không phân biệt hoa thường

9

Như 1.

Thay thế các đoạn "ABC" có trong chuỗi đầu vào bằng "DEF", **CÓ** phân biệt hoa thường

10

Như 1.

Hiển thị chuỗi: Kính chào ông **input**. Chúc ngon miệng.

(Giả sử input = "Vũ Văn Chiến" -> "Kính chào ông Vũ Văn Chiến. Chúc ngon miệng.")

11

Như 1.

Đảo ngược các ký tự của chuỗi và in ra màn hình chuỗi đầu vào và chuỗi kết quả

12

Input: 1 chuỗi và 1 số (count)

Nếu chuỗi null -> Thông báo: **Chuỗi bị null**

Nếu chuỗi không có ký tự nào -> Thông báo: **Chuỗi rỗng**

Nếu chuỗi toàn ký tự space -> Thông báo: **Chuỗi Space**

Nếu số < 0 -> Thông báo: **Số nhỏ hơn 0**

Còn lại: Bỏ các ký tự trắng ở đầu chuỗi, cắt lấy count ký tự ĐẦU và in kết quả ra màn hình

13

Như 12

Còn lại: Bỏ các ký tự trắng ở đầu và cuối chuỗi, cắt lấy count ký tự ĐẦU và in kết quả ra màn hình

14

Như 12

Còn lại: Bỏ các ký tự trắng ở đầu và cuối chuỗi, cắt lấy count ký tự CUỐI và in kết quả ra màn hình

1.1. Int

1

Input: 1 số

Nếu số ≤ 0 -> Thông báo: **Số ≤ 0**

Còn lại: Nếu là số chẵn -> Thông báo: **Số chẵn**

Nếu là số lẻ -> Thông báo: **Số lẻ**

2

Input: 1 số (x)

Hiển thị trị tuyệt đối của x

3

Input: 1 số

Nếu số ≤ 0 -> Thông báo: **Số ≤ 0**

Còn lại: Hiển thị 2 mũ x

4

Input: 2 số (X, Y)

Hiển thị kết quả: **X chia Y bằng ... dư...**

5

Input: 1 số

Nếu số ≤ 0 -> Thông báo: **Số ≤ 0**

Còn lại: Nếu là số nguyên tố -> Thông báo: **Số nguyên tố**, ngược lại thông báo **“Không**

phải số nguyên tố”

6

Input: 1 chuỗi.

Nếu chuỗi là chuỗi số Int32 -> Thông báo: **Chuỗi số.**

Ngược lại -> Thông báo: **Không phải chuỗi số**

1.2. Bool

1

Input: 1 chuỗi.

Nếu chuỗi là “true” hoặc “false” -> Chuyển thành kiểu bool và thông báo: **Chuỗi hợp lệ.**

Ngược lại -> Thông báo: **Không phải chuỗi bool**

1.3. Decimal

1

Input: 1 chuỗi.

Nếu chuỗi là chuỗi số thực (gồm các số, dấu ngăn cách phần nghìn là “,”, dấu phân cách thập phân là “.”): **Chuỗi hợp lệ.**

Ngược lại -> Thông báo: **Không phải chuỗi decimal**

Ví dụ chuỗi hợp lệ: 100; 1,000,000; 1,000.5

2

Input: số decimal.

Chuyển số input thành string và in ra màn hình theo định dạng dấu ngăn cách phần nghìn là “,”, dấu phân cách thập phân là “.”, làm tròn 02 số sau dấu thập phân

1.4. DateTime

1

Lấy ngày hôm nay

Thông báo: **Hôm nay là thứ ..., tháng..., năm...**

2

Input: 1 ngày.

Thông báo: **Hôm nay là thứ ..., tháng..., năm...**

3

Input: 1 ngày.

Thông báo: **Hôm sau là thứ ..., tháng..., năm...**

4

Input: 1 ngày.

Thông báo: **Hôm trước là thứ ..., tháng..., năm...**

5

Input: 1 ngày.

Nếu là hôm nay -> Thông báo: Ngày hôm nay

Nếu là sau hôm nay -> Thông báo: Ngày tương lai

Nếu là trước hôm nay -> Thông báo: Ngày quá khứ

6

Nhập 1 ngày.

Hiển thị ngày dưới dạng: Ngày/tháng/năm. VD: **18/03/2014**

Hiển thị ngày dưới dạng: Năm/tháng/ngày. VD: **2014/03/18**

Hiển thị ngày dưới dạng: Tháng/năm. VD: **03/2014**

Hiển thị ngày dưới dạng: Tháng-năm. VD: **03-2014**

7

Input: 1 ngày.

Hiển thị 10 ngày trước là thứ mấy

8

Input: 1 ngày.

Hiển thị cuối tháng này là thứ mấy

9

Input: 1 ngày.

Hiển thị cuối năm nay là thứ mấy

10

Input: 2 ngày.

Hiển thị 2 ngày này cách nhau bao nhiêu ngày

2. Collection

2.1. List

Note: Các số trong bài đều kiểu int

1

Input: None

Tạo 1 danh sách số int

Thêm vào danh sách 10 số bất kỳ

In kết quả ra màn hình và trả về danh sách vừa tạo

2

Input: danh sách 10 số bất kỳ

Sắp xếp danh sách theo thứ tự tăng dần, in kết quả ra màn hình

3

Input: danh sách 10 số bất kỳ

Đảo ngược vị trí các số trong danh sách, in kết quả ra màn hình

4

Input: danh sách 10 số bất kỳ và 1 số (X)

Bỏ đi các số nhỏ hơn X khỏi danh sách, in kết quả ra màn hình

5

Input: danh sách 10 số bất kỳ và 1 số (X)

Bỏ đi các số chia hết cho X khỏi danh sách, in kết quả ra màn hình

6

Input: danh sách 10 số bất kỳ và 1 số (X)

Với mỗi số trong danh sách, nếu nhỏ hơn X thì cộng với X

In kết quả ra màn hình

7

Input: danh sách 10 số bất kỳ và 1 số lớn hơn 0 (X)

In X số cuối ra màn hình

8

Input: danh sách 10 số bất kỳ và 1 số lớn hơn 0 (X)
In X số cuối ra màn hình theo chiều ngược lại

9

Input: 1 số lớn hơn 0 (X)
Tính dãy Fibonacci của số nhập vào
In 10 giá trị kết quả ra màn hình
Trả về dãy đã tạo

2.2. Dictionary

1

Tạo 1 Dictionary chứa key chứa mã Sinh viên kiểu int, value là Tên SV
Thêm vào Dictionary 5 giá trị
Trả về Dictionary vừa tạo

2

Nhập 1 Dictionary, 1 mã SV
Nếu mã SV đó có trong Dictionary, in tên SV đó
Ngược lại thông báo: *Không có SV có mã...*

3

Nhập 1 Dictionary, 1 mã SV và tên SV
Nếu mã SV đó chưa có trong Dictionary, thêm SV đó vào Dictionary, thông báo: *Đã thêm SV có mã..., tên... vào Dictionary*
Ngược lại thông báo: *Đã có SV có mã...trong Dictionary*

4

Nhập 1 Dictionary, 1 mã SV và tên SV
Nếu mã SV đó chưa có trong Dictionary, thêm SV đó vào Dictionary, thông báo: *Đã thêm SV có mã..., tên... vào Dictionary*
Ngược lại thay tên SV cũ bằng tên SV mới, thông báo: *Đã thay tên SV có mã...trong Dictionary từ... thành...*

5

Nhập 1 Dictionary, 1 mã SV
Nếu mã SV đó có trong Dictionary, bỏ mã SV đó khỏi Dictionary
Ngược lại thông báo: *Không có SV có mã...*

3. Class, Object

3.1. Property

3.2. Method

3.3. Reference

3.4. Inherit

3.5. Static

3.6. Test

1

Áp dụng kiến thức hướng đối tượng để:

Tạo class Employee chứa field: Tên, Tuổi, Thời gian tạo và 2 method là ShowInfo() và ExtInfo().

Tạo class Student kế thừa Employee và bổ sung thêm field Chuyên ngành.

Tạo class Teacher kế thừa Employee và bổ sung thêm field Bộ môn

Trong đó:

+ ShowInfo(): Hiện thị thông tin

- Nếu là Người: "Tôi tên là ..., ... tuổi."

- Nếu là Sinh Viên: "Tôi tên là ..., ... tuổi. Tôi là Sinh Viên"

- Nếu là Giáo Viên: "Tôi là Giáo Viên, ... tuổi, tôi tên là ..."

+ ExtInfo(): Hiện thị

- Nếu là Người: "Tôi tên là ..., được tạo ra lúc..." (**hiển thị đến giây**)

- Nếu là Sinh Viên: "Tôi là Sinh viên tên ..., học chuyên ngành..., được tạo ra lúc..."

- Nếu là Giáo Viên: "Tôi là Giáo Viên tên..., thuộc bộ môn..., được tạo ra lúc..."

4. Exception

Tìm hiểu về throw, try catch, re-throw Exception

Chú ý sự khác nhau giữa 3 cách re-throw exception:

- `throw;`
- `throw ex;`
- `throw new Exception("Error", ex);`

Phân loại các Exception, xử lý từng loại exception cụ thể

5. Data storage

5.1. File

1

Nhập đường dẫn đến 1 file đầu vào và 1 file đầu ra.

Cấu trúc file đầu vào: mỗi dòng là 1 dãy số kiểu Int32 cách nhau bởi 1 hoặc nhiều dấu space

Ví dụ: 123 456 7634 2334 567

Yêu cầu:

Tạo file đầu ra theo cấu trúc:

- Số dòng của file đầu ra bằng số dòng của file đầu vào
- Với mỗi dòng của file đầu vào, sắp xếp dãy số theo thứ tự giảm dần và ghi vào một dòng của file đầu ra theo định dạng mỗi số cách nhau bởi 04 dấu space
- Nếu dãy số trong dòng dữ liệu của file đầu vào lỗi thì ghi vào file đầu ra thông báo lỗi
- + Số [số bị sai] bị sai format (không phải kiểu Int32)
- + Số [số vượt giới hạn] bị vượt giới hạn kiểu Int32
- Nếu có bất kỳ lỗi nào xảy ra khi mở file đầu vào để đọc, ghi ra file đầu ra 01 dòng với nội dung “Không đọc được file đầu vào. Lý do: [thông báo nguyên nhân lỗi]”

5.2. Database + DataTable

1

Tạo Database tên StudentMng

Tạo bảng Class

Tên trường	Kiểu	Mô tả
ClassID	Int tự tăng	
Name	nvarchar(256)	
Desc	nvarchar(256)	Mô tả

2

Tạo Database tên StudentMng

Tạo bảng Student

Tên trường	Kiểu	Mô tả
StudentID	Int tự tăng	Key
ClassID	Int	Khóa ngoại bảng Class. Có thể NULL
Name	nvarchar(256)	
DOB	datetime	Ngày sinh
Code	nvarchar(32)	Mã (SV0001,...)
Math	int	Điểm toán. Có thể NULL
Phys	int	Điểm lý. Có thể NULL

3

Viết lệnh Sql để:

1. Insert 20 bản ghi vào bảng Class và Student vừa tạo để làm dữ liệu test
2. Tìm sinh viên có mã "SV0001"
3. Tìm SV có điểm vật lý < 4
4. Tìm SV có cả 2 điểm < 5
5. Update tất cả điểm toán thêm 1
6. Update điểm vật lý thêm 1 nếu < 4
7. Update điểm vật lý và toán thêm 1 nếu cả 2 < 5
8. Xóa sinh viên có mã "SV0001"

4

Viết lệnh Sql để:

1. Đếm số SV có điểm vật lý < 4
2. Đếm số SV có cả 2 điểm < 5
3. Đếm số SV có tổng 2 điểm < 10
4. Tính tổng số điểm của mỗi SV
5. Tính tổng số điểm toán của tất cả SV
6. Tính điểm toán trung bình của tất cả SV
7. Tìm 2 SV có điểm toán cao nhất
8. Tìm 2 SV có điểm lý thấp nhất
9. Tìm 2 SV có tổng điểm cao nhất

5

Viết lệnh Sql để:

1. Đưa ra danh sách sinh viên theo từng lớp (Bao gồm cả lớp ko có SV). Hiển thị: Tên lớp, Tất cả thông tin còn lại của SV
2. Đếm số lượng sinh viên theo từng lớp. Hiển thị: Tên lớp, Số sinh viên

5.3. Xml

Tạo 02 class tương ứng với 02 bảng trong mục 5.2 và thể hiện được mối quan hệ giữa Class và Student là 1 – n.

Khởi tạo object chứa thông tin Class và 5 sinh viên trong Class. Serialize object vừa tạo ra file XML. Desialize file XML vừa tạo ra object.

5.4. LinQ

1

Có 1 tập Employee gồm 2 thông tin: Tên và năm sinh.

Viết lệnh đề:

1. Tìm tất cả Employee có Tên chứa chuỗi "Test"
2. Tìm Employee đầu tiên có Tên chứa chuỗi "Test"
3. Lấy danh sách (string) tất cả Tên
4. Lấy danh sách (string) tất cả thông tin theo format: "Ông {0} sinh năm {1}"

1

Có 2 tập Employee gồm 2 thông tin: Tên và Năm sinh.

Viết lệnh đề:

1. Tìm tất cả Employee có Tên trong tập 1 mà ko có Tên trong tập 2
2. Tìm tất cả Employee có Tên trong cả 2 tập

6. Application Structure

6.1. Coding Convention

Đặt tên solution

SM.Training.BookingCar (Các form UI)

SM. Training.BookingCar.Biz (Các class xử lý nghiệp vụ)

SM. Training.BookingCar.Dao.(Các class tương tác dữ liệu)

SM. Training.BookingCar.SharedComponent (Các class dùng chung)

SM. Training.BookingCar.Util (Các class dùng chung, không liên quan nghiệp vụ)

6.2. Application

6.3. Dll

6.4. Layer (3 tiers)

7. WebForm

7.1. Page

7.2. Label, Literal: lbl, ltr

7.3. TextBox: txt

7.4. CheckBox: chk

7.5. RadioButton: rdb

7.6. ComboBox, Dropdownlist: cbo, ddl

7.7. Button: btn

7.8. DataGrid: grd

7.9. Repeater: rep

7.10. UserControl: uc..

7.11. MasterPage

8. DAO use Framework

Hướng dẫn cách sử dụng DAO dll

9. Prepare for actual project

9.1. SourceControl - SVN

9.2. Param

9.3. Design pattern (Facade)

10. Project: Booking Car

Mục đích

Cung cấp giao diện cho khách hàng đặt xe

Cung cấp giao diện cho quản lý setup các thông tin về Loại xe, Đơn vị thời gian thuê, Xe.

Cung cấp giao diện cho quản lý cập nhật thông tin về order: Hủy order, Giao xe, nhận lại xe, thay đổi thời gian đặt cho từng xe trong đơn đặt. Một lần order có thể order nhiều xe.

Chức năng cho khách hàng

Khách hàng nhập thông tin cần thuê (thời gian dự định thuê, loại xe cần thuê) và click tìm kiếm. Hệ thống trả ra danh sách các xe còn. Khách hàng có thể chọn 1 hoặc nhiều xe và click "Đặt xe". Hệ thống sẽ chuyển sang trang xác nhận đặt xe. Khách hàng nhập thông tin về mình và click "Xác nhận". Hệ thống sẽ tiến hành order và thông báo kết quả cho khách hàng sau đó tự động chuyển về trang đặt xe để khách hàng tiến hành đặt tiếp.

Chức năng cho quản lý

- Quản lý Đơn vị thời gian thuê (giờ, ngày)
- Quản lý Loại xe
- Quản lý Danh sách Xe
- Xem danh sách order (tìm kiếm theo Tên khách hàng, Trạng thái, Loại xe, Thời gian bắt đầu, Thời gian kết thúc), thay đổi thời gian đặt xe của order khi trong trạng thái Đang đặt (thay đổi thời gian bắt đầu và kết thúc) hoặc Đang sử dụng (thay đổi thời gian kết thúc), hủy order, giao xe cho khách hàng, nhận lại xe từ khách hàng.

Road map: ADO.Net thuần -> dùng DataContext -> Áp dụng pattern facade -> Dùng SM UI control -> Dùng SM dynamic report

Chi tiết DB như bên dưới

Tự define và validate các trường bắt buộc nhập để đáp ứng nghiệp vụ, các trường Date nhập vào theo định dạng dd/MM/yyyy

Có thể thay đổi DB cho phù hợp

Car:

Column Name	Type	Description
CarID	int	Mã ID
CategoryID	int	Loại xe
Color	nvarchar(32)	Màu sắc
Description	nvarchar(512)	Mô tả (MultiLine)
PlateNumber	nvarchar(8)	Biển số xe
Price	decimal	Giá xe (>0)
Deleted	int	0: Đang dùng, 1: Đã xóa
Version	int	Mới tạo: 1. Mỗi lần update tăng lên 1
CreatedBy	nvarchar(32)	Người tạo
CreatedDTG	DateTime	Giờ tạo
UpdatedBy		Người update
UpdatedDTG		Giờ update

ServiceOrder: Đặt xe

Column Name	Type	Description
OrderID	int	Mã ID
CarID	int	Mã xe
TimeCategory	int	Đơn vị thời gian
OrderDuration	int	Thời gian thuê (số giờ/ngày.. theo Đơn vị thời gian)
PlanStartDTG	Datetime	Thời gian bắt đầu

Column Name	Type	Description
PlanEndDTG	Datetime	Thời gian kết thúc
ActualStartDTG	Datetime	Thời gian thực tế thuê
ActualEndDTG	Datetime	Thời gian thực tế trả
CustomerName	nvarchar(512)	Tên khách hàng
Description	nvarchar(512)	Mô tả
Status	int	Trạng thái
Deleted	int	
Version	int	
CreatedBy	nvarchar(32)	
CreatedDTG	DateTime	
UpdatedBy	nvarchar(32)	
UpdatedDTG	DateTime	

CarCategory : Danh mục xe

Column Name	Type	Description
CarCategoryID	Int	Mã loại xe
Name	nvarchar(256)	Tên loại xe
Description	nvarchar(512)	Mô tả
Deleted	int	
Version	int	
CreatedBy	nvarchar(32)	
CreatedDTG	DateTime	
UpdatedBy	nvarchar(32)	
UpdatedDTG	DateTime	

TimeCategory

Column Name	Type	Description
TimeCategoryID	int	Mã thời gian thuê
Name	nvarchar(32)	Tên thời gian thuê
Description	nvarchar(5000)	Mô tả
Deleted	int	
Version	int	
CreatedBy	nvarchar(32)	
CreatedDTG	DateTime	
UpdatedBy	nvarchar(32)	
UpdatedDTG	DateTime	

Chú ý. Các trường hệ thống

Column Name	Type	Description
Deleted	int	0: Đang dùng 1: Đã xóa
Version	int	Mới tạo: 1. Mỗi lần update tăng thêm 1
CreatedBy	nvarchar(32)	Người tạo
CreatedDTG	DateTime	Thời gian tạo (Theo app)
UpdatedBy	nvarchar(32)	Người update
UpdatedDTG	DateTime	Thời gian update