PLB
FORMATION INFORMATIQUE
01 43 34 90 94
www.plb.fr

1.

janvier 2014 - Michel André

.Net = middleware | entre applica° et OS, de type Framework.
↳ les services, logiques, applicables à pleins de métiers
↳ pls couches dedans.

Framework : ens. de règles, de contraintes.

I-3 : CLR = JVM (en java) (couche entre hardware et langage C#) — V4.5
↳ remonte-mille
↳ traduc° à la volée.

CLS : pas d'alloca° manuelle de la mémoire.
Transmission des params par défaut (valeur ou référence)
...

FCL : foundation class Library (~~V2.0 sous Win~~ ↳ stable)
↳ les tableaux, chaîne
↳ types données
↳ ...

FCL comprend :
↳ BCL (Base Class Library) : V2.0 en 2005 / ↳ stable
↳ ADO.Net.

I-4 :
DLL .Net très #tes des DLL windows.
DLL sont mises ds GAC : global assembly cash
on lance toto.exe → = en .net il y a un stub qui lance le CLR puis le fct main est lancée.

web service | : - décrit ds un XML
- écrit ds plein de langages possible (interopérable)
- "GLU" (?) : Fuyanterie

ETS : common type s... ?
↳ car ds certains langage, par ex, int sur 16bits, et 32bits sur d'autres

odbl (à virer commande console) ds visual studio / Tools.
↳ variables env. bien positionnées.

>ildasm : Intermédiate langage désassembleur
>exit

attributs et mO) commencent par majuscule.

> csc /t:library voiture.cs => génère voiture.dll.

(del *.exe)

> Vbc /r:voiture.dll voitureClient.vb
vb compiler.

DIP : dependancy inversion principle.

---

encapsulation : pour limiter les dépendances (qd évolution)
(en interdisant accès en modif)
objet = fournisseur de service dont le contrôle aux attributs de cet objet.
⇒ contrôle de l'intégrité des données au sein d'un objet.

---

renommer - class
- projet
| clic droit dessus / refactoriser / renommer

---

mode de session | debug => infos stockées pour pas à pas
release

≠ mode de compilation "debug"

ctr + F5 : exec sans débuggage (en sens "debug")

Suppression d'un projet 'voiture'
dans main, on instancie voiture ... => bug.
ds main, on clic droit sur voiture / générer / class
" démarrer / " / stub mO
=squelette.

TDR
Test driven Requirement

depuis VS 2010
snippet = cw puis 2 fois sur tab => écrit console.writeline.

placeholders {0}, vitesse };

(F5) démarrer
(F11)
(F10)

verrouiller/déverrouiller : qd on se place sur variable
petit icone => la valeur reste affichée.

constructeur : si on en définit un (constructeur) par défaut, cela impose une contrainte lors de la fabrica° de l'objet. (= aussi une vérif)
→ se méfier des initialisa° par défaut : Point(int x, int y = 100) { this.x = x; this.y = y; } ⇒ point (e0) compilé

Point p = new Point(x:0, y:30);
↳ le CLR appelle le constructeur (et pas moi)
⇒ pas besoin de "void" devant la défini° du constructeur Point.

internal void démarrer
↳ interne à l'assemblage (= n° physique (disque dur) ; assemblage = dll spécifique, ou = exe spécifique)

⚠ si on ne met rien, ↯ sous C#11, par défaut = le test le ⊕ restrictif (⇒ ici = privable) ⇒ Démarrer n'est plus accessible depuis la classe Main (ou Démarrer de class Voiture)

⚠ si on met "Public" : accessible depuis toutes les assemblages ⇒ pas sécure : trop laxiste...

Public class voiture
↳ mettre seulement si on veut l'exporter cette classe. (≠ de "Public pour attribut).

---

un pointeur = une adresse (4 octets de mon de 32 bits) (= une référence (à un objet))

voiture v = new voiture();
" v2 = v; } → compteur de référence = 2 ; l'objet Voiture ne sera pas supprimé par le ramasse miette

v2 = null; → objet ref = 1

le ramasse miette déplace tous les objets pour qu'il n'y ait plus de trous dans la mémoire. ⇒ coûteux en CPU.
≃ process de défragmenta° d'un disque dur

On n'appelle pas directement le destructeur (~)
sauf via interface IDisposable.
le Destructeur t'jrs appelé qd shutdown du programme

exp. destructeur : ~Voiture() { }
Public ~~internal~~ privée → on ne met rien.

Interface = classe abstraite, la @ abstraite
si 1 méth @ abstraite ⇒ classe abstraite

- pas de membre
- toutes les m@ sont abstraites

⚠ ~ pour destructeur : pas de mot clé public/private/internal car pas d'export

- Dispose appelé dès qu'on sort un bloc using
- IDisposable
  si I souligné (parce que m@ dispose pas implementée) clic dessus /générer/ interface.

=> génère public void Dispose() { }

classe statique
GC.SuppressFinalize(this); => pour éviter que le GC soit appelé 2 fois de suite.
garbage collector
m@ statique
=> Dispose n'est plus appelé par ce pointeur implicite.
=> l'objet est supprimé de la file du GC.

```
using (SQLconnec°) ...
  //  (transac°)
  //  (requête) ...
```

Dispose très utile pour l'accès aux BD car ça prend de la place

Console = Classe (constructeur privée ⇒ pas droit de faire Console c = new Console();
(classe statique)

Polymorphisme : "le traitement correspond à la donnée"
la prise en charge de cette distinction revient au compilateur (c'est pas au développeur)
Koala : appelle du nouveau code avec des nouveaux codes !

moyen coercitif d'imposer à une équipe le polymorphisme = Interface

PLB
FORMATION INFORMATIQUE
01 43 34 90 94
www.plb.fr

- Richard Par → Addin complété auto → créa° class, m0 ...
- Entreprise Architecte (UML) V10

Pour supprimer complètement un projet, clic droit sur projet / supprimer = pas suffisant, les fichiers sont tjr présents ds
C:\ Documents \ mes Documents \ nom solu° \ nom projet
↓ rép à supprimer ici.

Edition / avancé / mise en forme
et let A + E + D.

Eviter de mettre "set" ; ⇒ bien remplir le constructeur.
et si besoin de dégt en cours de route, ne pas nommer dos 'set' ≃ par ex: accelerVitesse, ...

Héritage = acquisi° autom.
↳ des membres → field (ou attributs)
→ m0
↳ des contrats (implémenter-constructeur) interfaces

Pas d'héritage multiple en .Net (qu'en C++) ↳ utiliser pls interfaces

pointeur : [référence à un objet = une adresse
et accès à une convertion fonctionnelle.

↳ {lion l = new lion();
{Animal a = l;

le CLR sait que a est du type Animal

↳ a adresse m l a accès à la m0 rugir de lion; et pas a...

---

virtual ↳ la m0 peut être surchargée
override ↳ " vient surcharger une m0 d'une classe héritée.
base... pour invoquer superclasse (classe niveau au dessus)

PLB
FORMATION INFORMATIQUE
01 43 34 90 94
www.plb.fr

F12 sur mot clé pour aide
ou alt+droit / atteindre la définition

Projet / propriété / débogger / arguments lignes de commande
(on peut les renseigner ici)

10 clients → 10 threads : si 1 plante alors tout plante !
on pourrait créer 10 instances du pg → lourd en mémoire

solution ⇒ créa° de plus appdomain / classe dynamique pour cloisonner / étanchéifier les parties du pg.

on rattache ensuite des thread à une instance de Appdomain.

struct : type valeur → sur la pile (pas le tas)
meilleure organisa° de la pile : éclaté à la suite les uns des autres
⇒ meilleur temps de traitement.

III-9 float f = 56,2 → plante ! (pourquoi ? ...) → mettre = 56,2f

byte = 8 bits → 0 à 255
checked { byte s1 = 100;
byte s2 = (byte)(s1 x s1); } ⇒ il n'y a pas troncature ⇒ pas d'exception

propriété projet / générer / opt° avancées / vérifier les dépassements de capacité.
⇔ (ici si on avait mis des checked partout : coûteux en temps ⇒ en débug seulement)

code managé

III-10 double d = 23.5
int n = (int) d ou n = Convert.ToInt32(d); (plus joli)
cast

III-12 : le textreader 'décore' le stream (= apporte des fonctionnalités suppl.)
ex : " peut interpréter un \n (retour ligne) ; pas le stream.

VI-14 : si on définit Point ici étant struct ⇒ ⇐ va sur la Pile (pas tas)
ici si on écrit Point p = new Point(1,2,3); (c'est le concepteur qui choisit)
(sert juste pour véhiculer argument (constructeur))

["struct : mêmes règles que pour int."].

⚠ Pas de ~~valeur par défaut~~ constructeur sans param pour les structs car elles ont déjà une valeur par défaut (ici tous les types sur la pile)

C++ / C# : pile (stack) | tas (heap) | segment lecture-écriture ⇒ pas de variable globale en C#

cliquer sur nom argument / refactoriser / renommer

III-25 : a.Ajouter(montant : 3)
↳ ⊕ clair en le mettant → (pas besoin de commentaires comme ça : seuls " doivent être la doc XML)
et permet de générer autom mθ avec arguments

Enum : prend ⊖ de place que tableau de string
⊕ sécuriser (choix limités)

Les éléments de ⊕ haut niveau doivent être initialisés
Ⓐ int n;
consolewriteline(n);) → bug : il faut mettre int n=3;
Ⓑ int[] tab = new int[3];
cw(tab[0]);) → ne bug pas et affiche "0"

foreach (tab)(tab) ⇒ marque (int item in collec°) autom.

Count = mθ d'extens° de linq
↳ rapide car ds le CLR : pas d'aller-retour entre la machine virtuelle (CLR) et la collec° [si ds une boucle FOR → ⊖ rapide]

Array.Sort(tab) ⇒ tri le tableau

ds Applica° / info de l'assembly : on peut renseigner la vers° d'une dll à utiliser ; puis ds code (pour compil conditionnelle) :
if (a.getType().GetCustomAttributes().GetType() ==
pour incrémenter le nom de la vers°

#region MYREGION
=) lignes de code
#endregion
) si obligés d'utiliser ça, c'est que c'est mal programmé (mθ trop longue, donc pas modulaire ...)

Add Reference → ajoute physiquem' une dll sur le disque dur
NB : si on met un "using XXX" en début de code, il faut que le namespace XXX soit référencé
la dll physique qui contient le namespace XXX soit référencée (chargée) → sinon bug.

ds c/windows/assembly (=GAC) ↳ biblio de dll
↳ il peut y avoir plus assembly avec le m̂ nom (en version ≠ ou nom pays ≠.)
non devers° avec 4 compartiments
ou ds fichier de config (XML) utiliser <assemblyBinding> pour dire quelle vers° utiliser ⇒ ne recompile pas !! cool
assembly.GetExecutingAssembly

CVS ou SourceSafe : gestion de configura° (GitHub)

overload : plsrs constructeurs avec signatures ≠tes
overide : redéfini° ds une sous-classe
) en français, un mot : "surcharge"...

```
private int n;
public A(int n){
    this.n = n; }
Public A(): this(3) { }
```
] 2 constructeurs ≠ts, dont 1 fait appel à l'autre => évite de recopier tout le corps d'un constructeur ds l'autre

ou (+court) avec paramètre par défaut :

```
Public B(int n = 3) {
    this.n = n; }
```

IV-6 private readonly int n;
(déf ds constructeur) ← initialisa° de n dedans ⚠ il doit l'être sinon bug
(ne affichera) ← n ne peut plus être modifié dedans.

IV-15 Classes qu'avec des mé static : Math, Convert...
défini° d'une nvlle : static class MyMath
{ ← dedans, que des mé static, sinon bug
}

IV-8 Personne p = New Personne { nom = "Valjean", prenom = "jean" }
↳ accolades (et pas parenthèses)
Microsoft avait besoin des expressions d'initialisa° par LINQ, donc nous la mette à dispo
Mais mieux vaut rester avec des constructeurs restrictifs (car de champs private)

IV-9 et 10 var v = new { Nom = "Toto", Prenom = "Titi" };) classe anonyme créée à la volée. utile pour Linq : on n'a pas à créer une classe contenant toutes les colonnes de notre select.
v.Nom = ... pas Personne

IV-11 surcharge d'Equals ⇒ il faut la redéfinir
sinon applique la mé Equals propre à un objet (= par référence)

```
public override bool Equals(object obj){
    Point other = (Point) obj;
    return this.x == other.x && this.y == other.y
```

voir aussi GetHashCode

Point(int x, int y)
puis if P1.Equals(P2)...

Si maintenant on redéfinit l'opérateur "==" et qu'on veut qd même comparer 2 références (pointeurs) d'objet, on peut faire : if(object.ReferenceEquals(P1, P2)) { ...

NB : un opérateur est une mé.

---

opérateurs de conversion
" d'affecta° pas redéfinissable en C#

public static explicit operator Point(PointD p){}

p = (Point)pDouble;
pas de cast à faire ← mot-clé : implicite : si pas de perte de donnée
cas à faire de la main ← explicite : si perte de donnée

01 43 34 90 94
www.plb.fr

Pour créer des propriétés autom. : sélectionner le champ / clic droit / refactoriser / encapsuler le champ.

propriétés autom

```
class Ellipse {
    public int Largeur { get; set; }
    // Longueur //
}
```

2 propriétés

pas champs Largeur, Longueur dans ressortent lorsque ...

⚠ car en fait dès qu'on utilise un constructeur paramétré (best practice), on doit définir longueur et largeur ...

mieux de ne les utiliser qu'avec Linq (et DB)

se place sur accolade ouvrante } ctrl + [↑] ⇒ se place sur accolade fermante.

Protected éviter ⇒ utiliser propriétés. et laisser les champs Private

⚠ ne jamais rendre un set public !

⚠ tjs écrire le code le ⊕ abstrait possible

ex : Figure r = new Rectangle (...)

plutôt que Rectangle r = " "

Propc / clic / Ajouter / Diagramme de classe ⇒ (... .cd) (2010)
la arborescence

Constructeur tjs du sens (temporel) superclasse puis sous-classe

n'apparaît pas dans cet onglet classe, on si classe A : base (...)
constructeur hérité déclaratif

Destructeur du sens contraire au Constructeur

mercredi 22/01

(WPF) s'impose / winform / ⊕ orienté graphiques.

Web : ASP = active server Page / pour IIS (?)

compléments = widgets

VBA a remplacé par C# ⇒ du visuel C#/Office/ Classeur Excel 2010

/Cloud : compte windows Azure pour déb. sur serveur cloud.

/Reporting : avec crystal Reports

/Sharepoint : gest° docs, adresses mails ...

en abandon ⇐ (/Silverlight (comme flash) ⇒ action° site web

/Tst : pour faire batteries de tests unitaires.

(WCF) = window communica° fondat° : couche au dessus des web services (propre à .Net).

/Workflow : à sharepoint, ~ avant, et en ⊖ puissant...

/windows store : pour windows8 (tactile)

readonly : initialisé ds constructeur uniquement

- outils / gestionnaire des extraits de code : pour voir tous les snippets on peut en ajouter.

- Intellisense
  ↳ on tape c → il fait toutes les propos° commençant par c, ≡ se positionne sur le + probable qui nous intéresse

- outils / importa° & exporta° de paramètres
  génère un fichier vs settings
  ↳ reprend les snippets (paramètres généraux)
  ↳ " l'état de Intellisense (paramètre d'analyse de code)

- sur champ privé (nom) ctrl + r + e ⇒ génère propriété.
- objets déclenchés par le système = objects bubble ; = excep°.
  car il remonte toute la pile d'appel (toute les classes qui s'appellent) tant qu'elle n'est pas gérée ; si va ds Main ⇒ msg erreur
  Si finally, ce bloc est exécuté ds tous les cas, m̄ si bug ds un catch
  ↳ utilisé pour libéra° de ressource (fermer fichier, DB, ..)
  ex : Parse ⇒ on le try/catch., ≡ ne pas en mettre partout car VS voit que c'est géré donc il ne livre plus ses alertes habituelles

- Add-on de VS pour tester une expressio régulière (sans taper tout le code)
- Documenta° d'un champ ou m θ (≠ commentaire) avec "///"
  ↳ juste au-dessus de class Contact.
  ↳ apparaît quand on survole l'objet.
  ↳ exploration d'objet : on voit toutes les doc.
  "Affichage/
  doc au format standard oxygen ⇓ généra° possible PDF

au projet / propriété / générer / fichier de doc XML ← le cocher
  ⇒ génère fichier .XML ds debug/bin.

---

nomenclature

Toutes les classes : | maj
namespace |

! public → maj
private = minuscule

nom composés : mot à l'intérieur commençant par maj

Ajouter / nouvel élément / fichier de paramètres.
ou
Propriété / Paramètres / créer un "

⇒ crée "Settings.setting" (= fichier XML) ds arborescence.

nom	type	Portée	valeur
chemin	string	Applica° / Utilisateur	c:\contact.csv

c/allusers/ ... c:/gourdan

internal = public au niveau de l'assembly
opérateur "+" très gourmand en mémoire si on va au delà des 250 caractères (c'est exponentiel...) ⇒ utiliser StringBuilder. → ou Directory. → ou File. (en anglais)

2 collec° :

① ArrayList vs ② List<T>

Liste d'objet hétérogène ⇒ bcp traitement avec cast... lent

↳ ⊕ Typée
↳ optimisa° traitem (mémoire, ...)
↳ ⇒ polymorphisme

ordre respecté : Queue (FIFO) / Stack (LIFO) — Queue<T> / Stack<T>

IEnumerable = Interface commune à toutes les collec°.

---

using System.Linq ⇒ donne accès aux m0 d'extens° avec "<>" à la fin
qd complétion d'un objet.

---

c. capacity
↳ on peut fixer le nbre de case remplies avant de retailler toute la liste.

Enumérateur : déjà "caché" ds le foreach ; donc ne sert pas bcp,
sauf on peut trier la list selon le champ de tri désiré puis on fait le foreach.

---

utilisa° de sort sur un contact ⇒ contact doit implémenter IComparable
⇒ on doit redéfinir CompareTo.

interface = garde-fou ! à travers le respect d'un contrat.
pour éviter les appels tardifs

CompareTo renvoie :
+1 si this > other
0 =
-1 < → string implémente IComparable ⇒ on peut se servir de CompareTo (et donc de sort) ...

Explorateur d'objet, appliqué à une dll : on voit les signatures des m0 = par les défini° des m0 ⇔ framework gratuit .Net = sources non accessibles
= VS payant.

Tuple :/ - swap (inversion de valeurs)
- plutôt que créer Contact, si on sait qu'il y a ts nom, prenom, mail.

---

délégué = pointeur sur m0
↳ ayant la signature (hors nom)
↳ toutes

Predicate = un élément de type Délégué.
(ex List est un élément de type Class...)

Linq : (3 types optimisés) | to object / to XML / to SQL

using ... Linq donne accès à des m0 d'extens° (qui complète des classes)
utilisa° d'expression lambda.

MySql
↳ Wampserver
↳ phpMyAdmin

Téléchargement gratuit de MSDN fourni par Microsoft
Pack office : Blue edition

VS
versions Express gratuites
Pro payantes → abonnement VS Pro avec MSDN
avec VS efficace installer SQL Server Express

List<string> l VS Dictionnary<string, int> notes;
≈ hashtable

	clé	valeur
inaccessible par nous	0	
	1	
	2	
	3	

hashcodes ↔ (clé rester unique)

clé	valeur
hascode 1 "math"	18
hascode 2 "eco"	15
... "sport"	15 20

int i = notes["sport"];
notes["sport"] = 20;

algo de hashage : MD5, sha1

hash code = code unique (chaîne de caractères) généré de façon à ce qu'on ait jamais le même hash code si on part de 2 objets à hasher.
"hasher" (≠ crypter)
Par contre pas de retour possible : hashcode → objet (≠ cryptage).

Dictionnary<Facture, List<LigneFacture>> d;
↳ qd on ne prend pas un type primitif ≈ clé, il faut overrider la m° Equals pour spécifier que 2 factures avec le même FactureId sont identiques (alors que l'objet Facture peut contenir d'autres FactureId de date

---

Webservice : dll normalement sur PC client avec .exe(s) d'applica°(s)
⇒ dès qu'il y a un chgt, écraser dll sur PC client, et éventuellement recompila° de(s) exe(s)
⇒ on met à dispo une dll sur un serveur unique, (= web service) (qd chgt → 1 seul dépl sur serveur unique)
↳ (flux XML interprétés par notre PC (≠ SOAP)
plus de données ↳ (≠ flux HTML mis à dispo par un site)

---

iis : informa° ... server = serveur d'applica° web.
VS contient un (iis) → en mode debug.
Appli visible par le web si - class public
et - Attribut [WebMethod]) au dessus de chaque m°

- langage SOAP
. le client veut consommer des webservice
Au dév. qui veut consommer de webservice, on fournit le fichier de description du service (rajouter : ?WSDL à la fin de l'URL)

Remoting : avantage car sur le serveur qui met à dispo le XML le fichier XML peut être généré en n'importe quel langage ; et les clients qui l'interrogent aussi.

snippet_ prop (tab) (tab)
– propa ?

Pour avoir accès à des Addons VS :
outils / extens° et mise à jour /

NB : 1 projet / Autres type de projet / extensibilité ...
↳ pour ajouter un Addon.
pas documentés.

- recherche (mySQL)
  ↳ dbforge fusion (gratuit 30 jours)

(↳ permet de dév du mysql dans l'envir VS (comme pour SQL server)

- recherche (PhP)
  ↳ .. gratuit !

les gratuits ne marchent pas tjs ...

- recherche (regular Expres°)
  ↳ Regular expres° Tester.

- ... (CSV)
  ↳ csv object generator.

ou googlehost / Code ↳ google repository https / souvent public en lecture
source Forge (équivalent Github)
tortoise SVN

- .. (SVN)
  ↳ gestion de conf SVN
  ↳ git source control

- commit] pour versioning - repository

windows Azure (gratuit pour repository) ↳ avec Teamfoundation Server [TFS]

on peut m héberger soit m ensemble fichiers (cf github) sur un server Apache

---

ADO (.NET) : Active Data object.

(ODBC : vieux ... ↳ connex° persistante : lourd mobilise ressources)

fonctionne en → Pool de connec° : logiciel à côté de la DB
forme 25 (par default) connec°, les stocke
et qd besoin d'une connex°, elle est allouée puis libérée ↳ une de ce pool (une fois oubliée)

(ADO.NET) → Driver MySQL / " SQL server / " oracle } évent modifier / rajouter m° en fct de la base attaquée (SQLserver, ...)

si le connecteur .NET n'existe pas, pour une BD donnée, on travaille en ODBC

- Connector MySQL pour .NET ↳ à télécharger (dll)
- " oracle data provider for .NET
- (sinon connector SQLserver déjà ds VS..)

---

outils / op° / Editeur de texte / C# / Mise en forme. : pour modifier lois d'indenta°, etc...

forme 3 onglets avec lien sur autre couleurs onglets. ← / exten° et mise à jour / recherche (productivity Power Tools 2012)

try (connec)
{ }
Finally (fermeture connec°)

meilleur → using (connec°) { }
avec + rapide performante
ça qd sort du using → nO dispose qui ferme la connexo.

⚠ il faut que l'objet (ici connexo) implémente IDisposable (⇒ faire implémenter nos objets qu'on crée, par IDisposable pour pouvoir utiliser using

étapes : ADO.NET
typer :
Connexo → MySqlConnexo
Requête → string
Paramètres → MySqlParameter
Exécution → MySqlCommand
Résultat → MySqlDataReader

la connectString peut être renseignée ds le fichier config : on peut ainsi la changer sans avoir à recompiler.

string req = "Insert into pays (nom, code) VALUES ('" + p1.Nom + "', '" + p1.Code + "');";

faire requête paramétrée ← (⚠ ne pas faire !! pas sécurisé : on peut introduire un delete, etc... ne gère pas formats date, prix (US) ...

web service n'envoie que des propriétés des objet (pas les nO) : les POJO.

Windows XP(3) mini) mini pour WPF .NET 3.0 | interop WPF : ne marche pas sur tous les composants WPF

clic droit sur élément VS de la boîte à outils
Ajouter les éléments
• NET ex scrollbar, ...
• WPF
• COM : ex : adobe PDF reader (pour qu'un Pdf reader soit visible ds un winform) ↓ s'il installé sur notre PC ⇒ mise à dispo de la dll adéquate
↓ toutes les dll installées sur notre PC
↳ les ajouter ds Références de notre projet.

on se place sur la classe et on clic F7 pour faire apparaître le code ds l'explorateur

Pour supprimer un évént, si on ne supprime que la nO ds la forms.cs ⇒ ça ne compile plus car le délégué est tjs ds le fichier Designer ⇒ Il faut aller ds Propriété / Event et supprimer celui-ci

PLB
FORMATION INFORMATIQUE
01 43 34 90 94
www.plb.fr

MSDN
Didacticiel du code Unsafe
Le pour mettre du code
dans managed (C++, ...)
DllImport → soucis si dll pas compatible com win32

- Le type d'un évènement est un délégué.
- On associe la m° button1_Click = l'évènement Click, via le délégué EventHandler.
- Depuis 3.0, on peut remplacer :
  this.button1.Click += new System.EventHandler(this.button1_Click)
  Par :
  += this.button1.Click
  "

---

web services

→ on lance F5 ds le VS webService
→ on récupère l'adresse en haut "... .asmx"
→ Ds le VS Form, ajout d'une service Reference
  ↳ on colle l'adresse ".asmx" ds "Adresse :" → on voit ma dispo
  ↳ Donner un nouveau nom = l'espace de nom
  ↳ ds bouton "Avancé", type de collec°, choisir 'Generic.List'
  ↳ puis OK
→ cette nvelle ref de service apparaît ds l'explorateur de solu°
  → ds reference.svcmap / reference.cs : détail du mapping
  → ds .wsdl, : détail du flux.
→ Ds form, on ajoute : using windowsFormApplication°Janvier.ServiceReference1;

Si on ajoute une nvelle m° côté webService, pour qu'elle soit utilisable côté VS Form, il faut clic droit sur service Reference Contact / mettre à jour la ref de service.

---

in V&M : modèle vue modèle de WPF (design pattern) (cf cours)
⚠ Pas de design Pattern de WinForm et WPF.

WPF : binding : pas en Form. → : one way
⇄ : 2 ways

WPF. DataGrid = équivalent de dataGridView (en Form) ≈ en bcp ⊕ puissant (& flexible)

public abstract class BaseObjects<T>
{
  public static T Read(
  void Delete(T obj, Connexion connexion)
  {
    connexion.session.Delete(obj);