

# ***III-Environnement de développement MS Visual Studio 2012***

EDI

VB .Net

Interface Builder

## **complémentarité des cours**

*Algo/C++* : programmation de base

*Architecture* : fonctionnement des machines

*ACSI* : conception d'applications / Bases de données

*BD* : gestion de données organisées

*Système* : gestion de l'environnement de travail

*Réseau* : communication entre les machines

*POO* : programmation 'objet'

➔ ***IIHM* : conception et programmation  
d'applications interactives**

# rappels sur le contexte (2)

Introduction à l'Interaction Hommes-Machine  
utilise les acquis des autres cours

Algorithmique / Structures de données  
Programmation Orientée Objet

introduit de nouveaux concepts

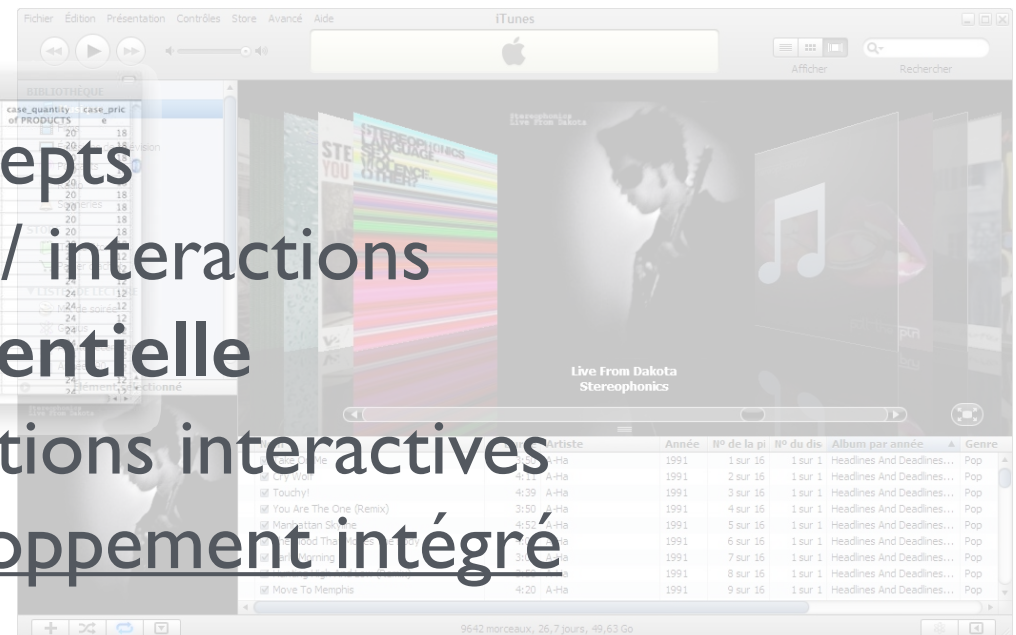
conception d'interfaces / interactions

programmation événementielle

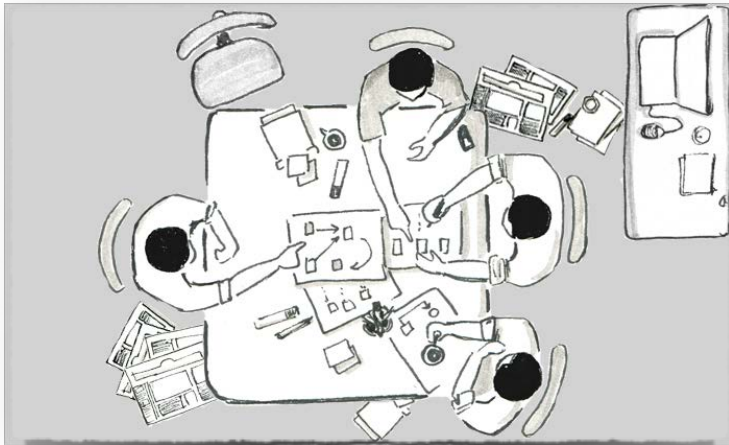
architecture des applications interactives

environnement de développement intégré

...



démarche



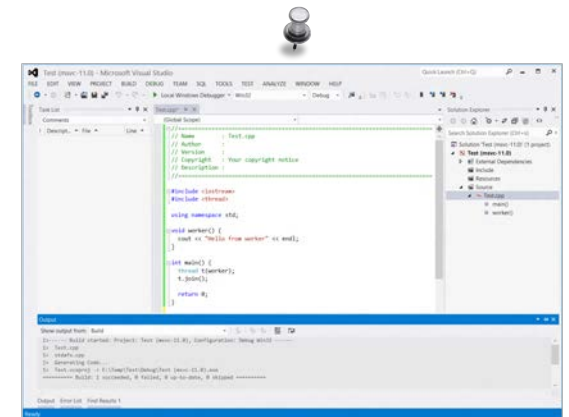
# Conception



[balsamiq mockups]



# Développement



[Visual Studio 2012]

# environnements de développement

en **BD**

SQL en mode console ou scripts

en **Système**

écriture avec éditeur 'standard'  
compilation ligne de commande

'make' et 'makefiles'

mise au point et débogage

traçage de variables

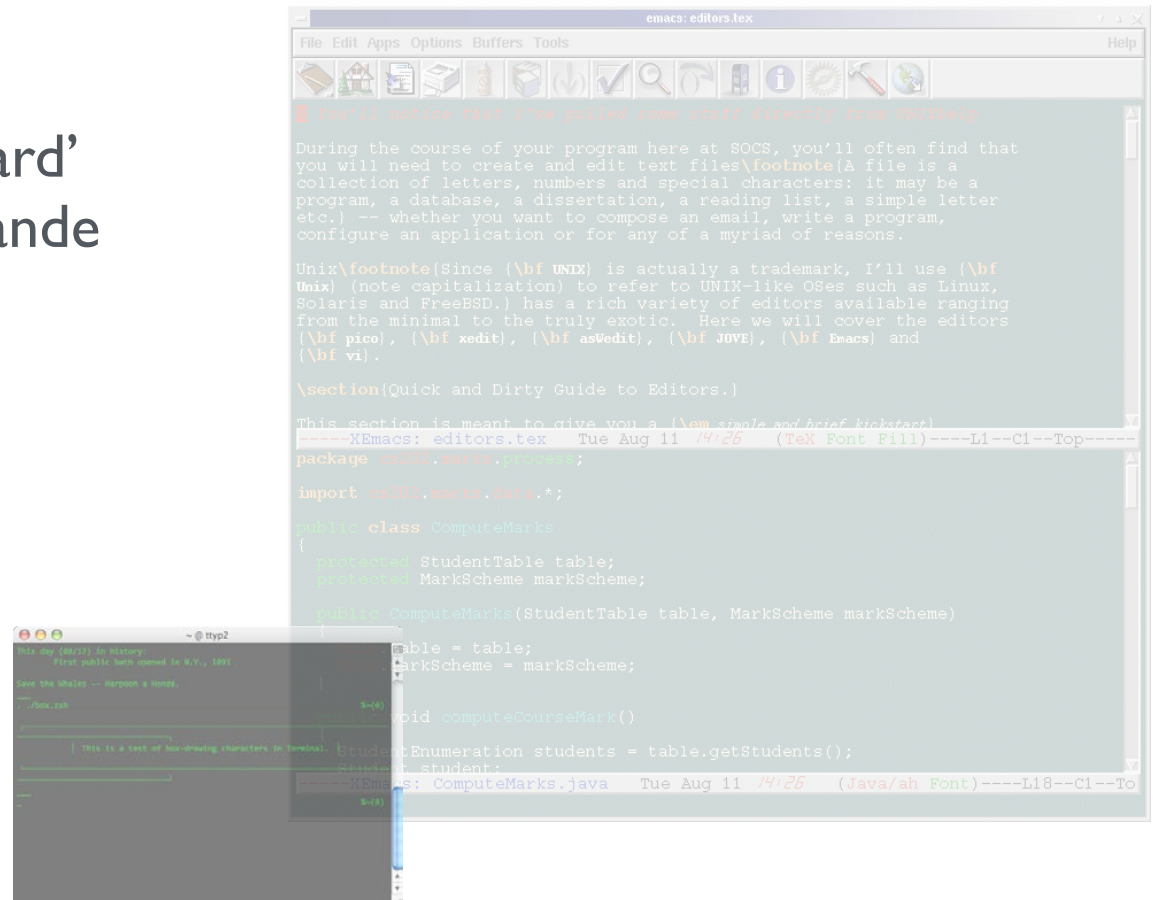
messages d'erreur

en **Algo/C++**

Code::Blocks

en **Prog. Objet**

Eclipse



**EDI = Environnement de Développement Intégré**  
**= IDE = Integrated Development Environment**

centré sur la notion de **solution** et de **projets**

*solution* = ensemble de projets (bibliothèques, applications, ...)

*projet* = regroupement de fichiers/ressources

réponse à la complexité croissante des systèmes à développer

outils d'analyse et de modélisation

outils de gestion du **cycle de vie**

outils de **documentation** automatique

### gestion du **code**

- versions locales

- gestion de versions concurrentes et centralisées

  - pour le développement en équipe (CVS, SVN, git, ...)

- archivage, import, export

### **documentation** intégrée

- liens directs dans le code (tooltips, completion, ...)

- documentation ‘intelligente’

- erreurs de compilation et d'exécution **explicites**



**compilation automatique et 'à la volée'**  
(Just-In-Time compilation, **JIT**)

- visualisation des erreurs de compilation pendant la saisie
- anticipation des erreurs d'exécution
- re-compiler des parties du code pendant l'exécution
- outils de débogage intégrés**

outils visuels et interactifs de création d'interfaces graphiques  
(**GUI** : **G**raphical **U**ser **I**nterface)  
**constructeur d'interfaces** (Interface Builder)



- Visual Studio
  - Microsoft
  - VisualBasic/C++/C#/J#
  - .NET
- CodeGears
  - Borland
  - C++ Builder/JBuilder/C#/Delphi/etc
  - Java API/.NET
- XCode
  - Apple
  - C++/ObjectiveC/Java
  - Carbon/Cocoa
- WinDev
  - PC SOFT
  - WLangage
  - Windows/.NET

- Eclipse
  - Java/C++/... (LaTeX)
- KDevelop
  - KDE
  - C++
- Code::Blocks
  - C++
- SharpDevelop
  - VisualBasic/C#
  - .NET

- Visual Studio
  - Microsoft
  - VisualBasic/C++/C#/J#
  - .NET
- CodeGears
  - Borland
  - C++ Builder/JBuilder/C#/Delphi/etc
  - Java API/.NET
- XCode
  - Apple
  - C++/ObjectiveC/Java
  - Carbon/Cocoa
- WinDev
  - PC SOFT
  - WLangage
  - Windows/.NET

- Eclipse
  - Java/C++/... (LaTeX)
- KDevelop
  - KDE
  - C++
- Code::Blocks
  - C++
- SharpDevelop
  - VisualBasic/C#
  - .NET

programmation **procédurale**  
une procédure par “action”

depuis VB7 (VB .NET): *Programmation Orientée Objet*  
tout est objet (ou presque)  
**entités de base** qui composent une application  
(composants graphiques, composants logiciels, etc.)  
propriétés: attributs  
méthodes

programmation **évènementielle**  
**réactions** de l'application en **fonction de l'environnement**  
(entrées utilisateur, événements système, etc.)

**framework .NET** = ensemble de bibliothèques, outils, modèles et conventions pour le développement d'application sur les plates-formes Microsoft

**infrastructure unifiée** commune aux différents langages pour les applications standard ou web (C++, VB, J#, C#, asp)

**bibliothèques et infrastructure**

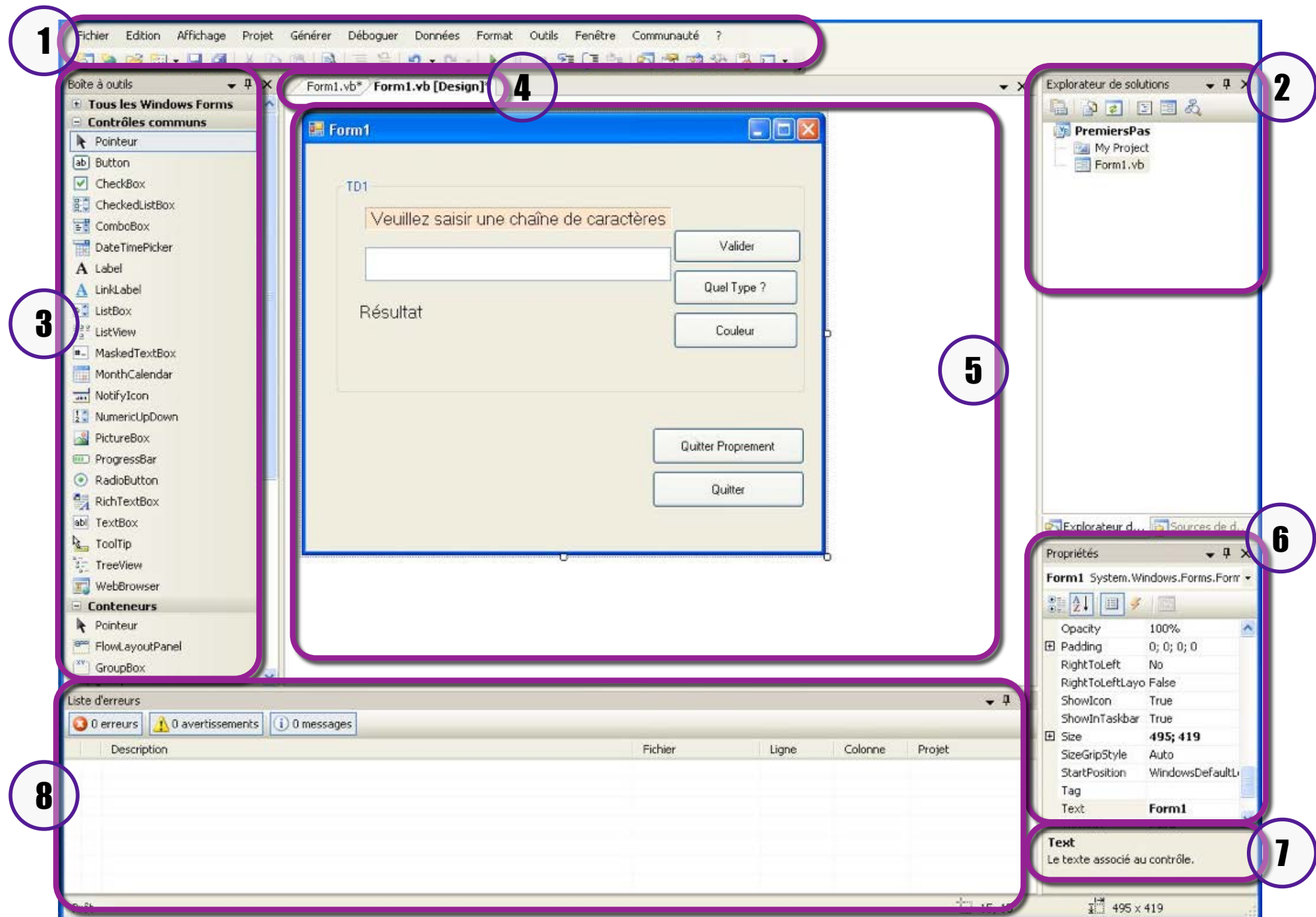
- composants d'interface graphique

- accès aux bibliothèques système Windows

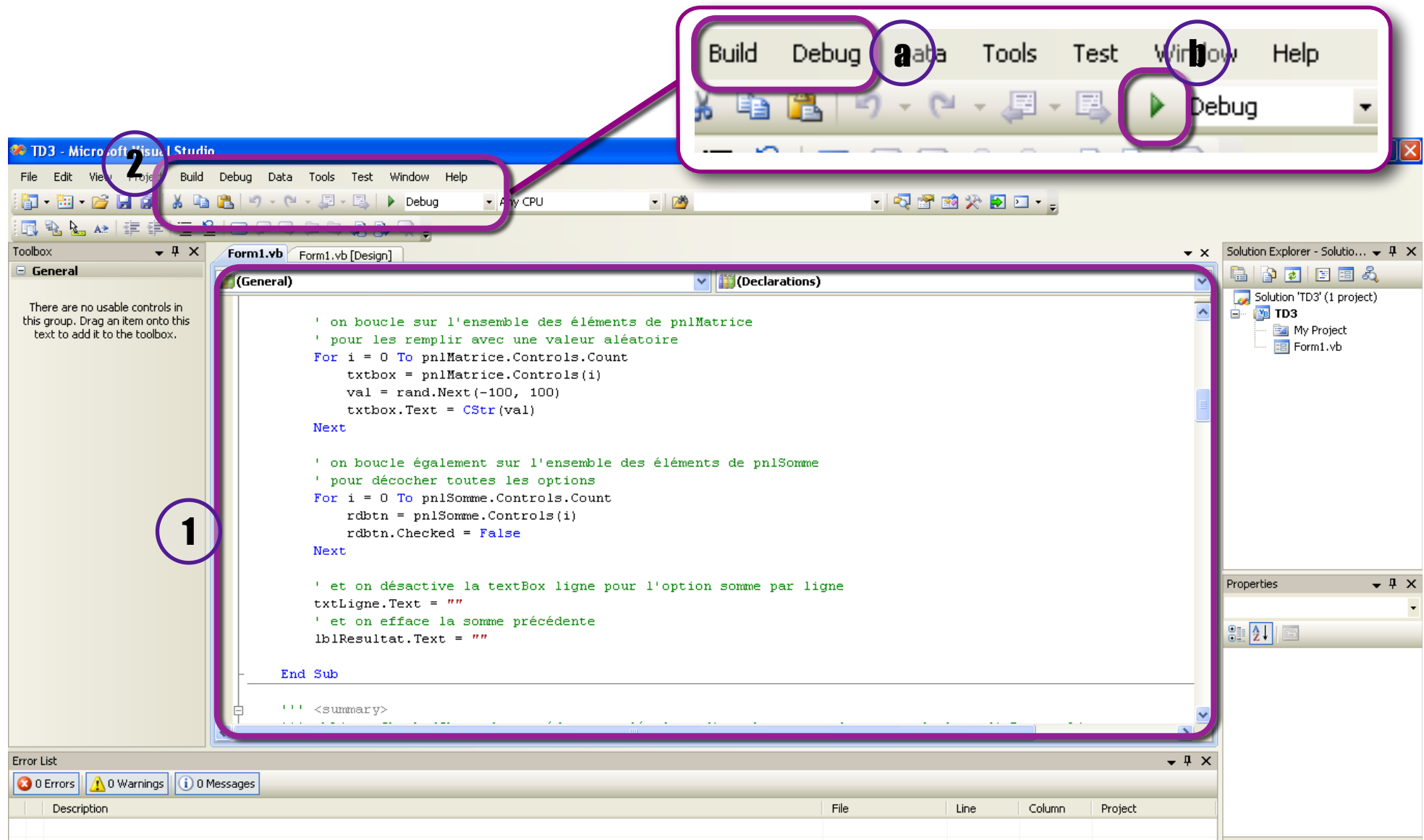
- accès aux données (BD)

- etc.

# Visual Studio: Interface Builder



# Visual Studio: éditeur de code



ensemble de fichiers/ressources pour la conception d'une bibliothèque, une application, d'un site web...

## avec Visual Studio

- une *solution* (fichier .sln)

- des *projets* dans la solution (fichiers .vbproj)

- des *fichiers* par 'feuille' ou form (fenêtres de l'application)

  - MaForm.designer.vb (éléments graphiques)

  - MaForm.vb (méthodes événementielles)

  - MaForm.resx (ressources et initialisations)

## arborescence de répertoires

- bin: fichiers binaires compilés (versions Debug et Release)

- MyProject: informations et ressources diverses

...



proche des structures de données vue en  
Algo: entités du programme

propriétés:

nom

couleur

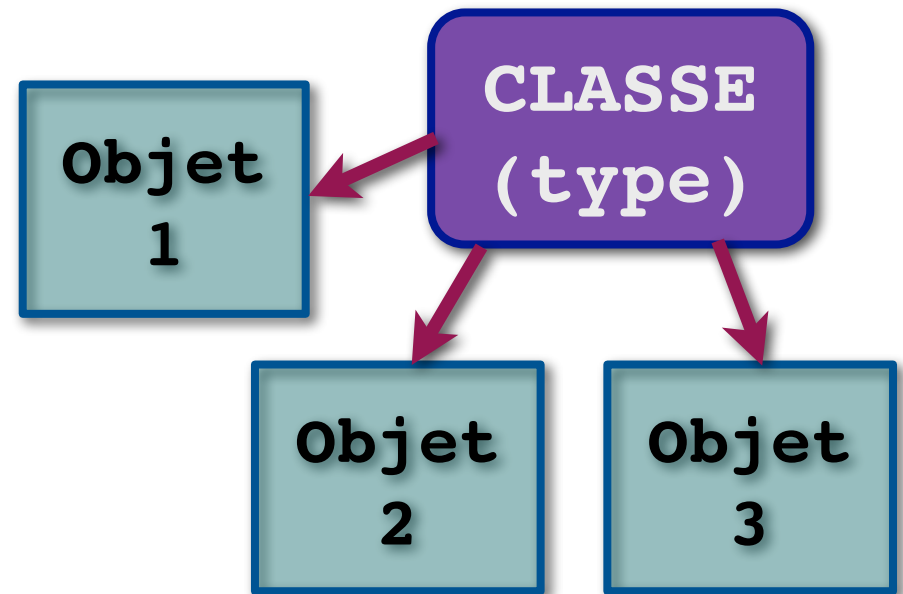
géométrie (taille, position, ...)

...

méthodes :

internes

réactions aux événements (ou réflexes)



C++ et Java (S2)

# *exemple 1: le formulaire / la fenêtre (form)*

---

composant de plus haut niveau

propriétés

géométrie (taille, position, ...)

comportement (visible, modale, ...)

...

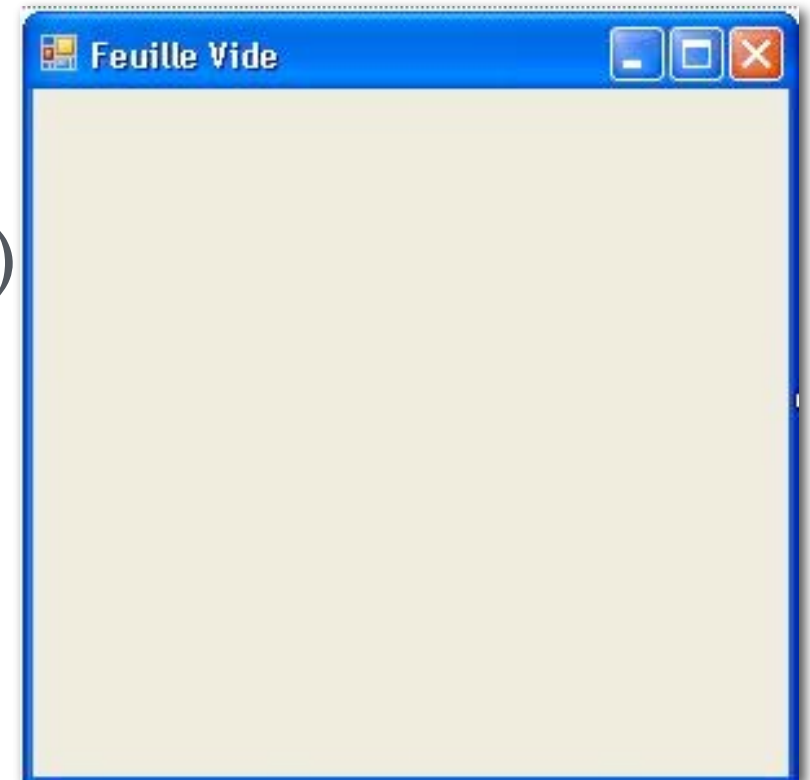
réactions

clic

activation/apparition

redimensionnement

...

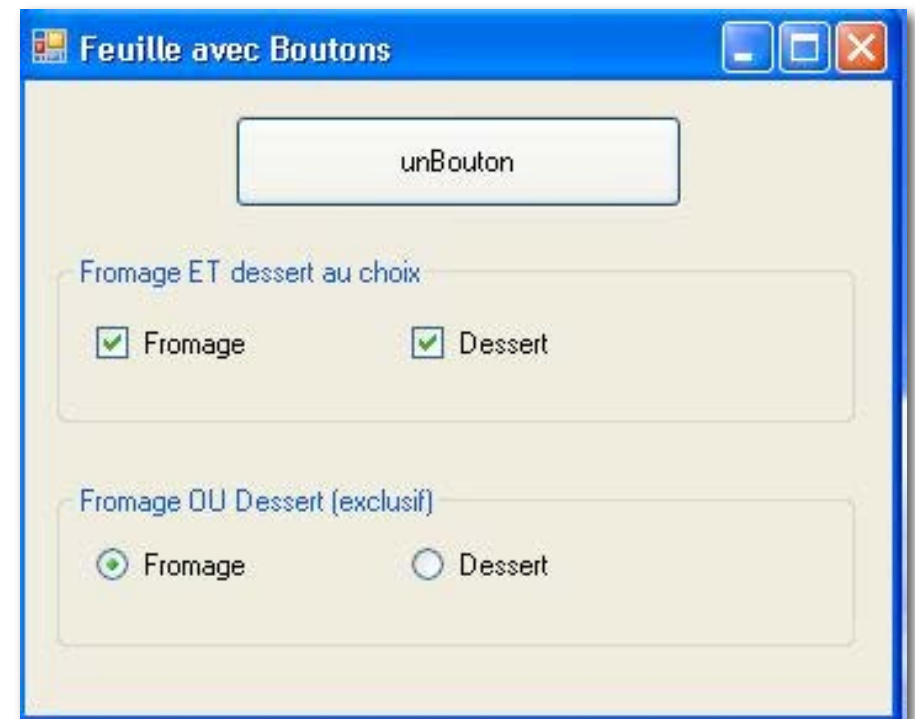


## *exemple 2: les boutons*

Button  
réaction au clic

CheckBox  
choix d'options

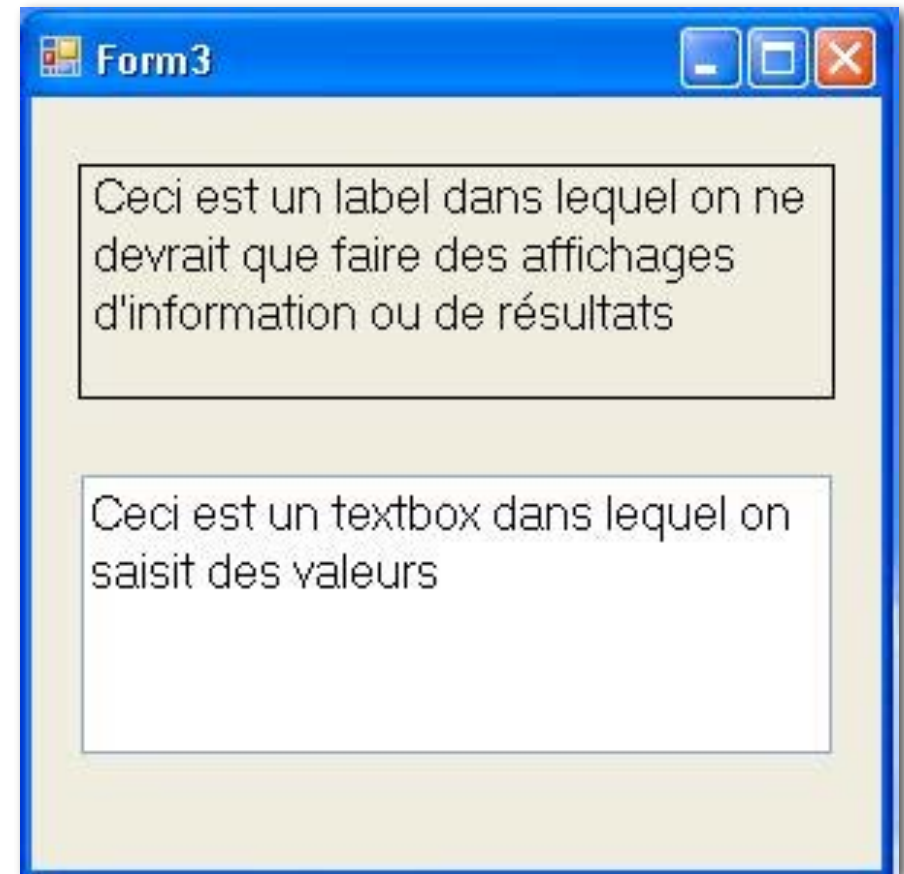
RadioButton  
choix unique  
dans un groupe



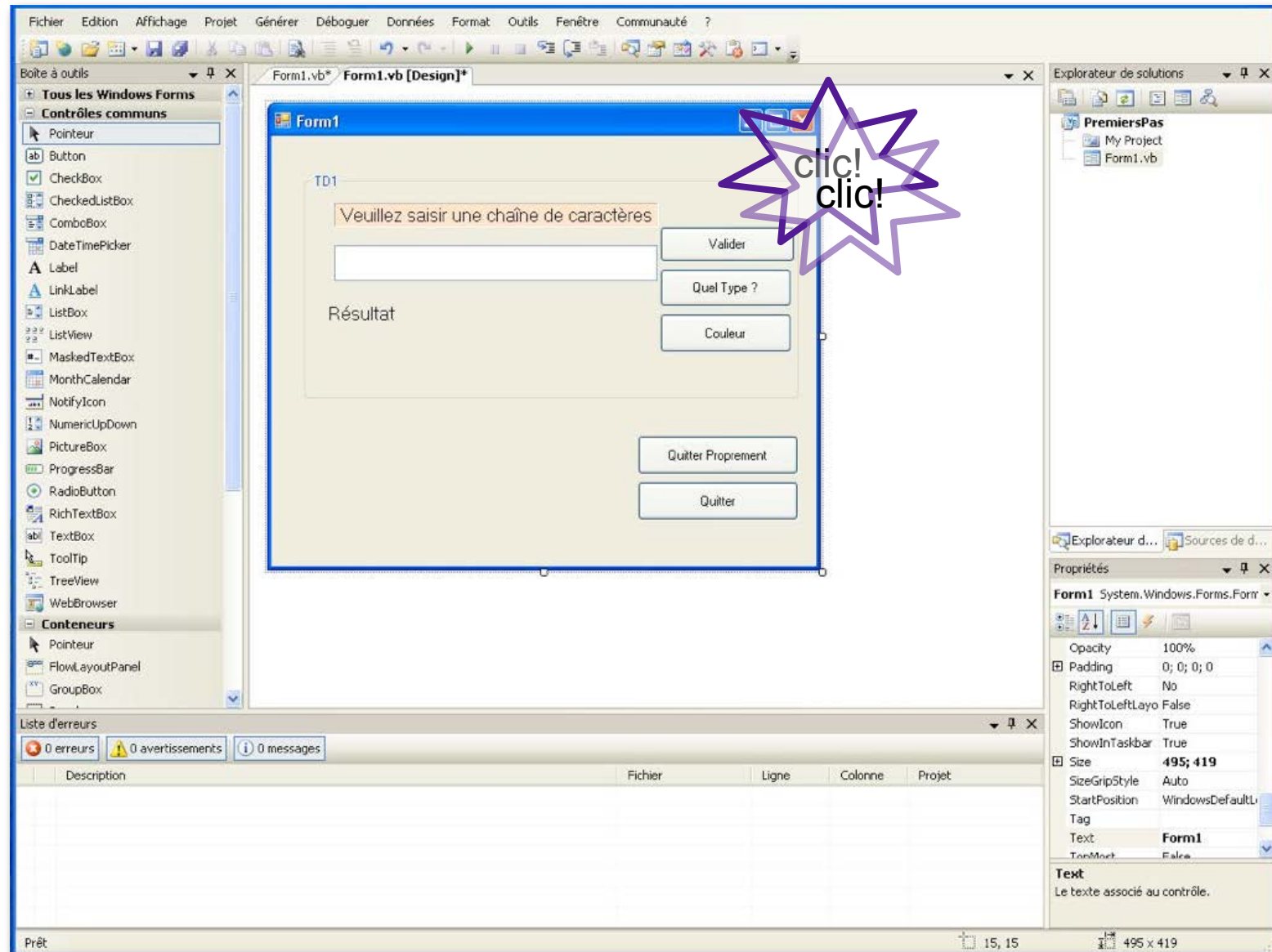
## *exemple 3: texte*

Label  
affichage de texte

TextBox  
saisie de texte



# *coder une réaction (1)*



## *coder une réaction (2)*

```
Private Sub btnValider_Click(ByVal sender  
As System.Object, ByVal e As  
System.EventArgs) Handles btnValider.Click
```

```
'Ecrire ici le code de la réaction du  
'programme au clic sur le bouton
```

```
End Sub
```

Downloaded from <http://ajph.org/> on November 10, 2015



[© Anna Lorenzetto]