

E6 - Compétence

C#



Introduction	3
Présentation du concept	3
Historique	3
Microsoft .NET	5
Caractéristiques	8
L'évènementiel dans le C#	9
Les outils de travail	9
Avantages et Inconvénients	11
Avantages :	11
Inconvénients :	11
Conclusion	11
Bibliographie :	12

Introduction

La première version du C# à été commercialisé par Microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft .NET.

Il est dérivé du C++ et très proche du Java dont il reprend la syntaxe générale ainsi que les concepts, y ajoutant des notions telles que la surcharge des opérateurs, les indexeurs et les délégués. Il est utilisé notamment pour développer des applications web sur la plateforme ASP.NET.

Présentation du concept

Le C# ou C Sharp est un langage de programmation événementiel créé par Microsoft, il est dérivé du C et du C++, il ressemble au langage Java. Il est utilisé pour le développement des applications de bureaux et des applications Web. Le C# est destiné à développer sur la plateforme.NET.

Historique

En septembre 2002 la plateforme .NET et le C# devinrent disponibles au public. Le C# devient le langage de cette plateforme.

La version 2.0 du C#.

Ajouter en 2005, avec la version 2.0 du Framework .NET (2.0). Cette version s'accompagne de mise à jour de presque tous les outils. Le C# version 2.0 ajoute de nouvelles fonctionnalités avec les ajouts suivants :

- Des classes partielles, cela permet de créer des classes sur plusieurs fichiers.
- L'utilisation de co-routines avec le mot-clé yield, équivalent du yield en Python.
- L'ajout des types null, ainsi cela permet de spécifier qu'un type de valeur peut être nul. On les déclare avec un point d'interrogation, exemple : `int? i = null;`
- L'ajout d'un nouvel opérateur le double point d'interrogation. Il utilise deux données, retourne la première si elle n'est pas nulle. Il a été introduit pour spécifier une valeur par défaut pour les types null.

La version 3.0 du C#.

Ajouter en 2008, avec une nouvelle version du Framework .NET (3.5). Elle introduit des nouveaux outils et fonctionnalités :

- L'ajout des mots-clefs SELECT, FROM et WHERE pour exécuter des requêtes SQL et XML.
- Possibilité d'initialiser un objet.

Exemple :

```
Client c = new Client();
```

```
c.Nom = "Dupont";
```

On peut utiliser

```
Client c = new Client{ Nom = "Dupont" };
```

- Inférence du type des variables locales
Exemple : `string s = "Dupont"` peut-être utiliser comme ceci `var s = "Dupont"`
- Méthodes étendues, elle permette d'ajouter des méthodes à une classe en y ajoutant un premier paramètre `this`.

La version 4.0 du C#.

La version 4 a été ajoutée en 2010 accompagné de la version .NET (4.0). On y ajoute également la version 2010 de Visual Studio. Les nouvelles fonctionnalités sont :

- Le typage dynamique des variables à l'aide du mot clé `dynamic`. (L'interpréteur attribue aux variables un type lors de l'exécution en fonction de la valeur qu'elles possèdent à ce moment.).
- Les arguments nommés et facultatifs.

Visual Studio 2010 ajoute aussi des nouveautés au C# comme :

- La bibliothèque Task Parallel Library qui sert à obtenir un ensemble d'API et de type publics.
- Une optimisation de la plateforme d'accès aux bases de données (Entity Framework).

La version 5.0 du C#.

La version 5 de C# a été ajoutée en 2012, il utilise la version 5.0 de .NET. On y retrouve des nouvelles fonctionnalités tel que :

- L'option de programmer des programmes asynchrones, le fonctionnement ne nous bloquera pas ce qui va nous permettre de lancer une tâche et d'avancer dans le programme le temps que cette tâche s'exécute. On utilise `async` et `await`.

- La possibilité de Break un foreach a été modifiée. Il n'est plus nécessaire d'utiliser une variable locale dans une boucle foreach cela évite les problèmes de closure.

La version 6.0 du C#.

La sixième version du C# ajoutée en 2015 ainsi que la version 4.6 du Framework .NET apporte une modification majeure :

- Les propriétés implémentées automatiquement (ou propriétés automatiques), elle simplifie la déclaration de propriétés. Elle permet d'encapsuler l'accès à des champs cela ajoute un atout tel que rendre le code plus concis, mais ajoute un inconvénient, il faut obligatoirement le faire dans le constructeur.

La dernière version du C# existante et la version 8.0.

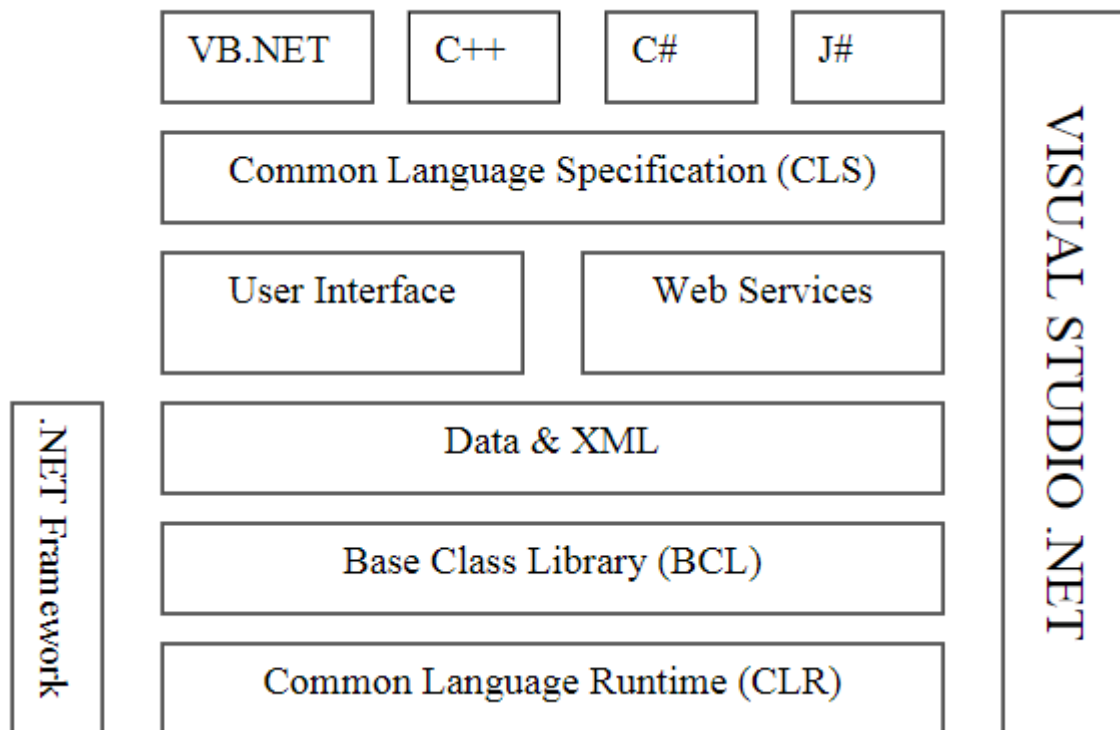
Microsoft .NET

Microsoft .NET ou Dot NET, est la base sur laquelle tous les produits et de technologie de Microsoft sont et seront développés. Le but étant de relier des systèmes, des données et des périphériques afin de rendre les applications facilement portables sur internet. Le service .NET fournit un serveur Web local permettant de gérer des services et évitant d'externaliser des données privées sur un service web de stockage ou un hébergement web tiers.

La plate-forme .NET se base sur plusieurs technologies :

- Les systèmes d'exploitation Microsoft Windows.
- Des protocoles de communication basés sur le Framework .NET.
- Une bibliothèque de logiciel compatible avec le Framework .NET.
- MSBuild qui est un outil de gestion de projet avec plusieurs compilateurs.
- Visual Studio qui est un IDE de développement.
- Le Framework .NET est un ensemble de bibliothèques.
- Une portabilité pour les systèmes d'exploitation Windows et Windows Mobile.
- Des composants facilitant le développement de services et d'applications locales ou web (ASP.NET).
- Mise en œuvre de la technologie Unicode.

Visual Studio .NET est un environnement de développement qui permet le design, le développement, le débogage et le déploiement rapide de solutions basées sur .NET Framework. Il est possible d'accéder à tout un ensemble communs d'outils à partir de n'importe quel langage de programmation (VB, C#, J#, C++).



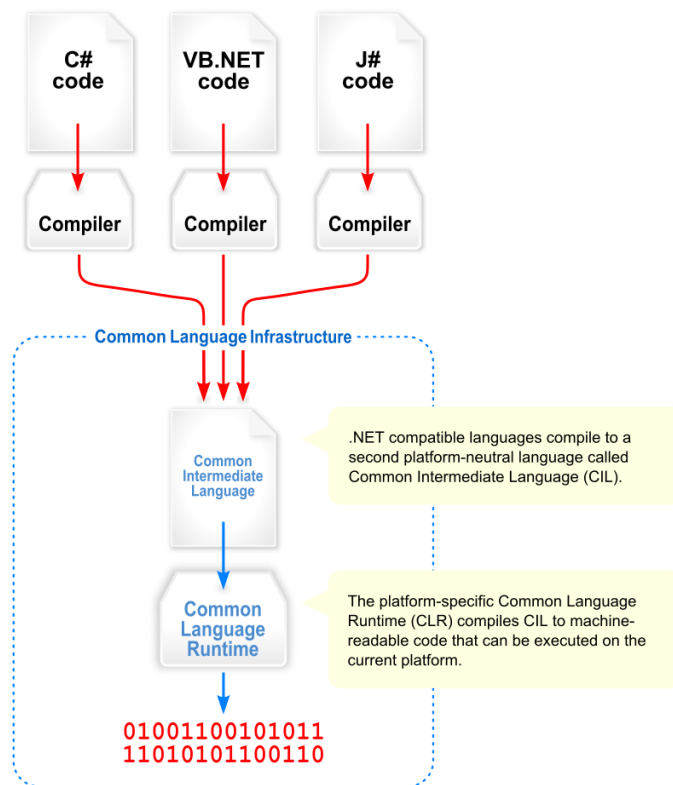
Framework .NET

Le .NET Framework, est inclut dans la plateforme Microsoft .NET. Son inclusion dans la plateforme permet de gérer divers aspects de l'exécution de l'application. Le fait que cela soit un Framework ajoute un cadre et une exécution Manager. Cette infrastructure s'appuie sur la norme CLI (Common Language Infrastructure).

Le CLI, fait partie de la stratégie .NET de Microsoft, il permet à un programme dans l'un des langages de programmation couramment utilisés de s'exécuter sur tous les systèmes d'exploitation à l'aide d'un programme d'exécution commune plutôt que d'un langage spécifique.

CIL, ou Common Intermediate Language, ce langage n'est pas interprété, mais il subit une compilation à la volée et une compilation au niveau de la Common Language Runtime CLR. La CLR est l'implémentation de la CLI.

CLR ou Common Language Runtime, il prend en charge la gestion de la mémoire (allocation et libération). Il alloue de la mémoire pour instancier les types .NET (objets). L'objet est considéré comme étant utilisé par le CLR, quand il n'y a plus de référence sur cet objet il ne peut n'y l'atteindre n'y l'utilisé.



Caractéristiques

C# apporte un typage sûr, ainsi que les possibilités d'encapsulation, d'héritage et de polymorphisme (est un concept consistant à fournir une interface unique à des entités pouvant avoir différents types) des langages orientés objet. En C# tous les types sont des objets. Le langage comporte un système de gestion d'exceptions.

Le typage sûr signifie que les opérations suivantes sont refusées : utilisation de variable non initialisée, tentative d'accéder au-delà des limites d'un tableau, conversions de type dont les résultats ne sont pas prévisibles, dépassement des limites lors d'opérations.

En C# les variables peuvent être d'un type référence ou d'un type valeur. Les types valeur sont les types primitifs, les énumérations, les struct et les types nullable. Les types référence sont les classes, les interfaces, les tableaux et les delegate.

Types primitifs, les types primitifs sont sbyte, short, int, long, byte, ushort, uint, ulong, char, float, double, decimal et bool.

Class, les constructions les plus fondamentales du langage C# sont les classes. Celles-ci peuvent contenir des constantes, des champs, des propriétés, des indexeurs, des méthodes, des événements, des opérateurs, des constructeurs, des destructeurs ou des sous-classes³. Les classes élémentaires sont string et object.

Struct, les struct sont similaires aux classes, mais ce sont des types valeurs et ils ne peuvent pas être hérités.

Delegate, un delegate est une référence à une méthode qui comporte certains paramètres. Les delegates permettent d'assigner des méthodes à des variables et les passer en paramètre.

Enum, un type énuméré est un type "valeur" qui comporte un lot de constantes. Chaque type énuméré a un type sous-jacent : un type primitif qui est déterminé en fonction des valeurs des constantes.

Type null, les type null sont des types primitifs qui peuvent en plus avoir la valeur null. Chaque type primitif T a un type nullable associé T?. Par exemple une variable de type int? peut contenir un int ou null.

L'évènementiel dans le C#

Le C# se base sur les principes de la programmation événementielle. Les composants des applications événementielles, les objets interagissent entre eux et avec l'environnement. Ils communiquent entre eux en fonction de certain événement. Les événements peuvent correspondre à différente action réaliser par l'utilisateur sur l'application.

Un clic droit sur un bouton déclenchera une action le fait de faire un clic gauche en déclenchera une tous autres que l'on peut personnaliser. Il en est tous à fait pareil pour les zones de texte, case à cocher, image etc.... Ils peuvent aussi se déclencher par le système comme le chargement d'un form, une heure précise par l'horloge.

Les événements seront alors captés par le système, le traitement consiste à exécuter les procédures des événements lui étant associés. Le développeur doit alors prévoir tous les cas de figure qui peuvent être envisagé et leurs définir un événement.

Par exemple, le déclenchement de l'événement click sur un bouton quitter doit terminer l'exécution, le choix d'un élément dans un menu doit déclencher certaines opérations, un top d'horloge peut modifier le contenu d'une zone d'image.

Un objet se définit par une propriété, une méthode et un ou plusieurs événements. Il faut résonner avec cette manière : Si l'événement E se produit sur l'objet O alors le traitement T est exécuté.

Les outils de travail

Les environnements de développement intégrés ou IDE (Integrated Development Environment) sont un ensemble d'outils qui en plus des tâches classiques d'un langage offre des fonctionnalités étendues permettant de couvrir une grande partie de la création du logiciel. On peut distinguer différentes tâches qu'ils peuvent effectuer comme :

- La compilation et débogages.
- Déploiement d'application.
- L'intégration d'outils de test et de vérification.
- Des outils de création graphique.
- L'interface avec des SGBD.

Différent IDE existe et il ne permette pas de programmer dans tous les langages.

Exemple d'IDE : Borland Delphi, Borland Jbuilder, Oracle JDeveloper et Microsoft Visual Basic et C# ainsi que Microsoft Visual C++ qui fut remplacé par Microsoft Visual Studio.

Organisation d'une application avec Visual Studio

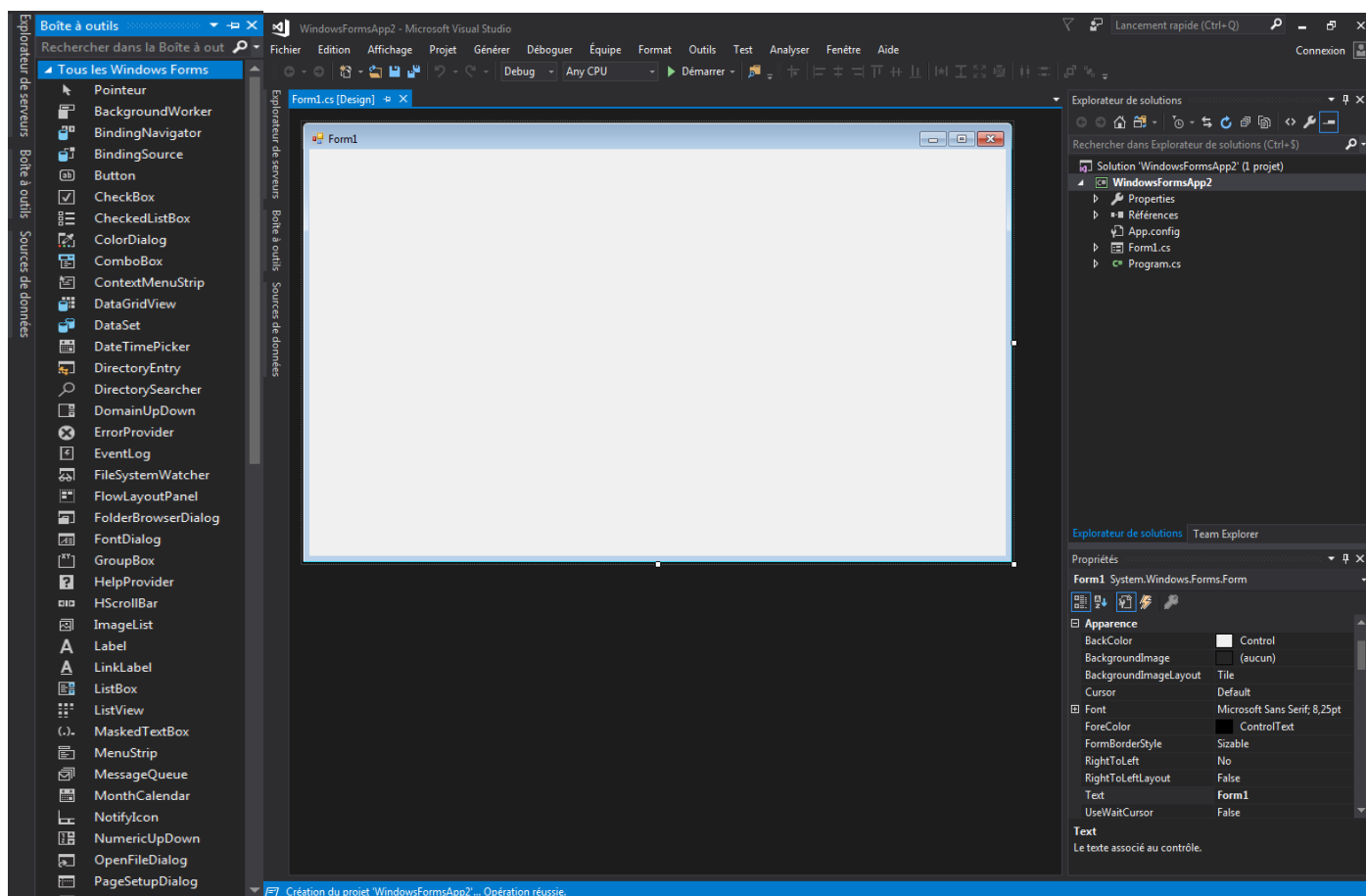
A la création d'un nouveau projet Windows Forms sur Visual studio. On voit la fenêtre de notre formulaire. En haut à droite il y a tous les fichiers composants notre solution.

WindowsFormsApp est le dossier principal qui contiendra toutes nos références.

Properties est un fichier dans lequel on peut stocker des informations de paramétrage pour pouvoir les modifier sans avoir à recompiler. Il a comme format clé=valeur, exemple.

References qui permet d'ajouter des références comme mysql.data.dll pour se connecter à la base de données.

Form1.cs est notre fenêtre principale. C'est là où nous pouvons ajouter button, label, picturebox etc....



Les fichiers C# ont plusieurs types d'extensions possibles :

- .sln : Visual Studio Solution : Organise les projets, les éléments de projet et les éléments de solution dans la solution (conteneur de projets)
- .suo : Solution User Options : Contient toutes les options associées à la solution.
- .cs : Fichiers source c sharp qui appartiennent au projet.
- .csproj : Projet c sharp: regroupe plusieurs formulaires et fichiers de modules.

Avantages et Inconvénients

Avantages :

- Les performances la gestion de la mémoire ou le fait que la mémoire est allouée automatiquement. Ce qui apporte une certaine stabilité.
- L'implémentation du ramasse miette.
- Visual Studio (suivie de mise à jour, personnalisation vaste)
- La communauté présente.
- Documentation de Microsoft.
- Moins typé que le C++.

Inconvénients :

- La portabilité sur des machines autres que Windows.
- Lenteur au niveau de la compilation.
- Gourment en ressources

Conclusion

En conclusion, c'est un excellent langage généraliste. Il nous permet de mettre l'impasse sur des concepts assez techniques tels que la gestion de la mémoire. La gestion des erreurs est assez robuste pour ne pas faire crasher les processus en cours d'exécution. Mais il reste développer par Microsoft donc très limité dans d'autre environnement que celui de Windows. Même si le Core du .Net est disponible sur toutes les autres OS.

Bibliographie :

<https://docs.microsoft.com/fr-fr/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
https://fr.wikipedia.org/wiki/C_sharp
<https://docs.microsoft.com/fr-fr/dotnet/standard/parallel-programming/task-parallel-library-tpl>
<https://docs.microsoft.com/fr-fr/dotnet/csharp/whats-new/csharp-7>
https://fr.wikipedia.org/wiki/Microsoft_.NET
<https://searchmicroservices.techtarget.com/definition/Common-Language-Infrastructure-CLI>
<http://www.technologuepro.com/cours-programmation-evenementielle/chapitre-1-introduction-a-la-programmation-evenementielle.html>
<https://www.developpez.net/forums/d834150/java/plateformes-java-ee-jakarta-ee-spring-serveurs/serveurs-d-applications/tomcat-tomee/fichier-properties-quoi-ca-sert/>
https://forum.hardware.fr/hfr/Programmation/CNET-managed/avantages-rapport-java-sujet_104786_1.htm