**C#** et C++ sont des langages de programmation où C++ est le descendant du C#. Cependant, C# est dérivé du langage C et possède de nombreuses fonctionnalités de C et C++, mais certaines fonctionnalités sont également supprimées dans C#. Et en ce qui concerne la productivité du programmeur, le C# est bien plus avancé que le C/C++.

La principale différence entre C# et C++ réside dans ses applications où C# peut être utilisé pour développer des applications Web professionnelles, tandis que le C++ est utile lorsque le programmeur souhaite créer quelque chose qui nécessite une interaction étroite avec le matériel.

|  | **C++** | **C#** |
| --- | --- | --- |
| **Type de langage** | C++ est un langage de programmation de bas niveau. | C# est un langage de haut niveau. |
| **Compilation** | C++ est compilé en code machine | C# est compilé jusqu’à CLR (Common Language Runtime), et interprété par JIT dans ASP.NET |
| **Gestion de la mémoire** | En C++, vous devez gérer la mémoire manuellement. | C# contrôle automatiquement la gestion de la mémoire |
| **Héritages multiples** | C++ supporte l’héritage multiples | C# ne prend pas en charge l’héritage multiples. |
| **Niveau de difficulté** | C++ inclut des fonctionnalités plus complexes. | C# n’a pas de fonctionnalités complexes. Il a une hiérarchie simple et assez facile à comprendre. |
| **Modificateurs d’accès par défaut** | Public en C++. | Private en C#. |
| **Plate-forme** | C++ est un langage qui fonctionne sur toutes sortes de plates-formes. Il est également populaire sur les systèmes Unix et Linux. | C#, bien que normalisé, est rarement vu en dehors de Windows. |
| **Applications autonomes** | C++ peut créer des applications autonomes. | C# ne peut pas créer des applications autonomes. |
| **Orienté Objet** | C++ un langage qui n’est pas orienté objet pure. | C# est un langage purement orienté objet. |
| **Instruction Switch** | Dans l’instruction Switch/Case, la variable de test ne peut pas être une chaîne de caractères. | Dans l’instruction Switch/Case, la variable de test peut être une chaîne de caractères. |
| **Résultat de la compilation** | Après la compilation, le code C++ est converti en code machine. | Après la compilation, le code C# est converti en langage intermédiaire. |
| **Utilisation** | Il vous permet de développer uniquement des applications en console. | La programmation C# peut être utilisée pour créer des applications Windows, mobiles et console. |
| **Utilisation de pointeurs** | Vous pouvez utiliser des pointeurs n’importe où dans le programme. | Vous pouvez utiliser des pointeurs uniquement en mode non sécurisé. |
| **Boucle Foreach** | C++ ne prend pas en charge la boucle for each. | C# prend en charge la boucle for each. |
| **Garbage Collection** | C++ ne prend pas en charge le garbage collector. | C# support le garbage collector. |

Table de comparaison

Conclusion

Le C++ est principalement utilisé lorsque le programme nécessite plus de performances, tandis qu’en C#, même les débutants peuvent développer facilement n’importe quelle application Web, mobile ou console. Cet article ne peut aborder que quelques points subtiles. Le CLR et le Framework .NET fournissent un support étendu pour le threading, le marshaling, le développement d’applications Web, le développement d’applications basées sur Windows, etc.