

C++

1 - Introduction au C++

Le C++ est un langage de programmation de haut niveau, compilé, généraliste et orienté objet. Il a été créé par Bjarne Stroustrup au début des années 1980, à partir du langage C.

Le C++ est un langage puissant et flexible, qui offre une gamme de fonctionnalités avancées et des performances élevées. Il est souvent utilisé pour développer des applications à hautes performances, telles que les systèmes d'exploitation, les logiciels d'ingénierie et les jeux vidéo.

Voici un résumé des principales caractéristiques du C++ :

- Programmation orientée objet : Le C++ permet de programmer en utilisant la programmation orientée objet, qui est une approche plus structurée et modulable de la programmation.
- Fonctionnalités avancées : Le C++ offre une gamme de fonctionnalités avancées, telles que la surcharge d'opérateurs, les templates et la gestion d'exceptions.
- Performances : Le C++ est un langage performant, qui peut être utilisé pour développer des applications à hautes performances.

2 - Introduction au C++

Voici un résumé plus détaillé de chaque point :

1. Programmation orientée objet

La programmation orientée objet (POO) est une approche de la programmation qui permet de modéliser le monde réel en utilisant des objets. Chaque objet a son propre état et comportement.

Le C++ prend en charge la POO en fournissant des classes, des objets, l'héritage, le polymorphisme et l'encapsulation.

- Les classes sont des modèles qui définissent les propriétés et les comportements d'un objet.
- Les objets sont des instances de classes.
- L'héritage permet aux classes de partager des propriétés et des comportements.
- Le polymorphisme permet aux objets de différentes classes d'être traités de la même manière.
- L'encapsulation permet de cacher les détails de mise en œuvre des objets.

3 - Introduction au C++

2. Fonctionnalités avancées

Outre la POO, le C++ offre une gamme de fonctionnalités avancées, telles que :

- La surcharge d'opérateurs permet de définir des comportements personnalisés pour les opérateurs.
- Les templates permettent de créer des fonctions et des classes génériques qui peuvent être utilisées avec différents types de données.
- La gestion des exceptions permet de traiter les erreurs de manière structurée.

3. Performances

Le C++ est un langage performant, qui peut être utilisé pour développer des applications à hautes performances. Cela est dû au fait que le C++ est un langage compilé, ce qui signifie que le code C++ est converti en code machine avant d'être exécuté.

Conclusion

Le C++ est un langage de programmation puissant et flexible, qui offre une gamme de fonctionnalités avancées et des performances élevées. Il est souvent utilisé pour développer des applications à hautes performances, telles que les systèmes d'exploitation, les logiciels d'ingénierie et les jeux vidéo.

4 - Introduction au C++

Installation et configuration de l'environnement de développement C++

Pour développer des applications C++, il est nécessaire d'installer un compilateur et un IDE.

Compilateurs

Un compilateur est un logiciel qui convertit le code source C++ en code machine, qui peut ensuite être exécuté par l'ordinateur.

- GCC est un compilateur open source largement utilisé.
- Clang est un compilateur open source qui est souvent considéré comme plus rapide que GCC.
- MSVC est un compilateur fourni par Microsoft et est la norme pour le développement C++ sur Windows.

5 - Introduction au C++

IDEs

Un IDE est un environnement de développement intégré qui fournit des outils pour écrire, compiler et déboguer du code.

- Visual Studio est un IDE commercial de Microsoft qui est la norme pour le développement C++ sur Windows.
- CLion est un IDE open source de JetBrains qui est spécialisé pour le C++.
- Code::Blocks est un IDE open source léger pour le C++.
- Eclipse CDT est une version de l'IDE Eclipse axée sur le C/C++.

Installation

Le processus d'installation varie selon le système d'exploitation et l'outil choisi.

- Sur les systèmes Linux, les packages sont généralement disponibles via les gestionnaires de paquets.
- Sur Windows, des installateurs sont souvent fournis.
- Sur MacOS, des outils comme Homebrew peuvent faciliter l'installation.

6 - Introduction au C++

Pour installer un environnement de développement C++, il suffit de télécharger et d'installer un compilateur et un IDE. Le choix du compilateur et de l'IDE dépend des besoins et des préférences de l'utilisateur.

Résumé plus concis

Pour développer des applications C++, il faut installer un compilateur et un IDE.

- Les compilateurs convertissent le code source C++ en code machine.
- Les IDEs fournissent des outils pour écrire, compiler et déboguer du code.

Les compilateurs et IDEs les plus populaires sont :

- Compilateurs : GCC, Clang, MSVC
- IDEs : Visual Studio, CLion, Code::Blocks, Eclipse CDT

L'installation varie selon le système d'exploitation. Notre choix est basé sur le Code::Blocks

1 - Les bases du C++

Structure d'un programme C++:

- Organisation:
 - Un programme C++ est constitué d'une ou plusieurs fonctions. La fonction main() est le point d'entrée de l'exécution.
 - Les programmes sont structurés en en-têtes (headers) et fichiers sources.
- Exemple de structure:

```
#include <iostream>    // Inclusion d'un en-tête
```

```
int main() {
```

```
    cout << "Hello, World!" << endl;
```

```
    return 0;
```

```
}
```


2 - Les bases du C++

Espaces de noms (namespace):

- Définition: Un espace de noms est une région qui permet de grouper des identificateurs (noms de variables, types, fonctions, etc.) afin d'éviter les conflits de nommage.
- Utilisation: L'espace de noms **std** est utilisé pour la STL (Standard Template Library). Il englobe des fonctionnalités telles que les opérations d'entrée/sortie et les conteneurs.
- Exemple:

```
using namespace std;    // permet d'utiliser les éléments de l'espace de noms sans préfixe ""
```

Entrée/sortie en C++ avec <iostream>:

- Composants principaux:
 - cin: pour l'entrée standard (clavier généralement).
 - cout: pour la sortie standard (écran généralement).
 - cerr: pour la sortie d'erreurs.
 - clog: pour les logs.
- Exemple d'utilisation:

```
int age;  
cout << "Entrer votre age: ";  
cin >> age;  
cout << "Vous avez " << age << " ans." << endl;
```