

Introduction

L'inventeur de C++, Bjarne Stroustrup, a travaillé sur des projets de développement très importants. Les premières applications de C++ ont été réalisées dans le domaine des télécommunications, domaine qui réclame des outils de haut niveau. Certes les classes favorisent l'abstraction, mais un programme n'est pas composé uniquement d'interfaces et d'implémentations.

À la conception du logiciel, le développeur doit déterminer la limite de réutilisation des réalisations précédentes. Derrière le terme réutilisation, on entend souvent le fait de copier/coller certaines parties de code source. Quels sont les éléments qui se prêtent le mieux à cette opération ? Les classes, naturellement, le modèle orienté objet étant construit autour du concept de réutilisation. Mais on trouve également des structures de données, des fonctions spécialisées, des pièces algorithmiques de natures diverses, n'étant pas encore parvenues à la maturité nécessaire à la formation de classes.

Les modules décrivent un découpage assez physique des programmes. Ainsi, un fichier de code source `.h` ou `.cpp` peut être considéré comme module. Toutefois, l'assemblage de modules n'est pas une chose aisée, surtout lorsqu'ils proviennent de réalisations précédentes. Il peut survenir des conflits de noms, de fonctions, des différences de représentation de types, ainsi qu'une déstructuration du programme, tous les membres d'un module apparaissant au même plan que les autres.

C'est pour ces raisons que C++ propose les espaces de noms ; il s'agit d'ensembles de variables, de fonctions, de classes et de sous-espaces de noms, membres obéissant à des règles de visibilité. Ainsi, le programmeur pourra organiser et ordonner son développement, fruit d'un travail nouveau et d'une agrégation de codes sources anciens.

Langage de haut niveau, C++ incite aussi à prendre du recul sur les développements achevés. On ne peut que s'étonner que les mêmes algorithmes se retrouvent dans tous les logiciels. Un algorithme, c'est un élément caractérisé, au comportement déterminé. Pourquoi ne pas le formaliser à l'aide d'une ou de plusieurs classes ? C'est précisément un des rôles tenus par la bibliothèque standard, la Standard Template Library.

À dire vrai, la STL constitue tout aussi bien un ensemble d'outils pour le programmeur qu'une preuve du fonctionnement du modèle orienté objet.