Glossaire

Programmation

algorithme
: Un algorithme énonce une résolution d'un problème sous la forme d'une série d'opérations à effectuer. Un algorithme est une suite finie d'instructions élémentaires écrites dans un langage universel et exécutées de manière séquentielle. La mise en œuvre de l'algorithme consiste en l'écriture de ces opérations dans un langage de programmation et constitue alors la brique de base d'un programme informatique. Un algorithme doit être suffisamment général pour permettre de traiter une classe de problèmes. Pour un problème donné, il peut y avoir un ou plusieurs algorithmes ou aucun.

bibliothèque logicielle
: Une bibliothèque logicielle est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Les bibliothèques logicielles ne sont pas complètement des « exécutables » car elles ne possèdent pas de programme principal et par conséquent ne peuvent pas être exécutées directement. Les bibliothèques logicielles sont : une interface de programmation (*Application Programming Interface* (API)), les composants d'un kit de développement logiciel (*Software Development Kit* (SDK)) parfois regroupées en un *framework*, de façon à constituer un ensemble cohérent et complémentaire de bibliothèques. Elles sont des fichiers : `.DLL` (*Dynamic Link Library*) ou Bibliothèque de liens dynamiques sous Windows et `.so` (*Shared Object*) ou Bibliothèque dynamique sous GNU/Linux. On retrouve parfois le terme impropre de « librairie » issu du faux ami anglais *library*.

fonction
: Une fonction est un sous-programme (sous-routine) qui retourne une et une seule valeur. Elle ne permet de ne récupérer qu'un résultat. Par convention, ce type de sous-programme ne devrait pas interagir avec l'environnement (écran, utilisateur). Dans certains langages de programmation, les fonctions permettent d'écrire aussi des procédures.

framework
: Un *framework* est constitué d'un ensemble cohérent et complémentaire de bibliothèques. Il fournit un cadre de développement. Les *frameworks* sont conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web, des composants logiciels.

implémentation (codage)
: Les informaticiens utilisent fréquemment l'anglicisme implémentation pour désigner l'écriture d'un algorithme dans un langage de programmation. L'écriture en langage informatique est aussi fréquemment désignée par le terme « codage », qui n'a ici aucun rapport avec la cryptographie, mais qui se réfère au terme « code source » pour désigner le texte, en langage de programmation, constituant le programme.

instruction
: Les langages de programmation offrent le concept d'instructions dans lesquelles on pourra utiliser des opérateurs. On rappelle que les instructions que l'ordinateur peut comprendre ne sont pas celles du langage humain. Le processeur sait juste exécuter un nombre limité d'instructions bien définies (son jeu d'instructions). La première des instructions est la possibilité d'affecter une valeur à une variable. La plupart des langages proposent l'opérateur d'affectation `=` pour cela.

paradigme
: Un paradigme est un style fondamental de programmation, définissant la manière dont les programmes doivent être formulés. Les plus connus sont : programmation impérative, programmation

orientée objet, programmation procédurale, programmation structurée, programmation concurrente, ...

paramètre

Un paramètre est une donnée manipulée par une fonction (ou méthode) et connue du code appelant cette fonction. On distingue deux types de paramètres : les paramètres d'entrée et les paramètres de sortie. Un paramètre d'entrée est une donnée fournie par le code appelant au code appelé. Cette donnée peut être transmise de deux façons : passage par copie (aussi appelé par valeur) : le code appelé dispose d'une copie de la valeur qu'il peut modifier sans affecter l'information initiale dans le code appelant et passage par adresse ou par référence : le code appelé dispose d'une information lui permettant d'accéder en mémoire à la valeur que le code appelant cherche à lui transmettre. Il peut alors modifier cette valeur là où elle se trouve ; le code appelant aura accès aux modifications faites sur la valeur. Dans ce cas, le paramètre peut aussi être utilisé comme un paramètre de sortie. Un paramètre de sortie est une donnée fournie par le code appelé au code appelant.

procédure

Une procédure est un sous-programme (ou sous-routine) qui permet de récupérer de 0 à n résultats. Par convention, ce type de sous-programme peut interagir avec l'environnement (écran, utilisateur). La notion de procédure ne se retrouve pas dans tous les langages de programmation. En C/C++, les procédures sont implémentables par des fonctions.

programmation

La programmation est l'ensemble des activités qui permettent l'écriture des programmes informatiques. C'est une étape importante du développement de logiciels. La programmation représente la rédaction du code source d'un logiciel. On utilise plutôt le terme développement pour dénoter l'ensemble des activités liées à la création d'un logiciel : cela inclut la spécification du logiciel, sa conception, puis son implémentation proprement dite au sens de l'écriture des programmes dans un langage de programmation bien défini et aussi la vérification de sa correction ....

programmation concurrente

La programmation concurrente est un paradigme de programmation tenant compte, dans un programme, de l'existence de plusieurs fils d'exécution qui peuvent être appelés threads, processus ou tâches. La programmation concurrente est plus complexe et difficile que la programmation impérative.

programmation impérative

La programmation impérative est un paradigme de programmation qui décrit les opérations en séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme. Ce type de programmation est le plus répandu parmi l'ensemble des langages de programmation existants. La quasi-totalité des processeurs qui équipent les ordinateurs sont de nature impérative : ils sont faits pour exécuter une suite d'instructions élémentaires. Les langages de plus haut niveau utilisent des variables et des opérations plus complexes, mais suivent le même paradigme. La grande majorité des langages de programmation est impérative. La plupart des langages de haut niveau comporte cinq types d'instructions principales : la séquence d'instructions, l'assignation ou affectation, l'instruction conditionnelle, la boucle et les branchements.

programmation orientée objet

La programmation orientée objet (POO) consiste à définir des objets logiciels et à les faire interagir entre eux. Il existe actuellement deux grandes catégories de langages à objets : les langages à classes (C++, C#, Java, Python, ...) et les langages à prototypes (JavaScript).

programmation procédurale

La programmation procédurale est un paradigme qui se fonde sur le concept d'appel procédural. Une procédure, aussi appelée routine, sous-routine ou fonction, contient simplement une série d'étapes à réaliser. N'importe quelle procédure peut être appelée à n'importe quelle étape de l'exécution du programme, y compris à l'intérieur d'autres procédures, voire dans la procédure elle-même (récursivité). Un programme est donc composé de procédures (et de fonctions) interagissant entre

elles.

programmation structurée

La programmation structurée, apparue en 1970, constitue un sous-ensemble de la programmation impérative. Elle est en effet célèbre pour son essai de suppression de l'instruction `goto` ou du moins pour la limitation de son usage à des cas exceptionnels. La programmation structurée recommande une organisation hiérarchique simple du code. Les programmeurs décomposent leur code en modules (appelés fonctions et procédures dans certains langages) ne dépassant guère 60 lignes, afin d'être présente en entier sous les yeux. On recommande aux programmes d'éviter l'usage des variables globales afin de prévenir les effets de bord (*side effects*). La plupart des programmeurs ont adopté la programmation structurée. Dijkstra rappelait qu'un programme devait d'abord être compris par le programmeur et ses collègues chargés de la maintenance, et que si cette tâche était accomplie, le reste (le faire exécuter à la machine) n'était plus que formalité.

routine

Une routine est une entité informatique qui encapsule une portion de code (une séquence d'instructions) effectuant un traitement spécifique bien identifié (asservissement, tâche, calcul, etc.) relativement indépendant du reste du programme, et qui peut être réutilisé dans le même programme, ou dans un autre. Dans ce cas, on range souvent la routine dans une bibliothèque pour la rendre disponible à d'autres projets de programmation, tout en préservant l'intégrité de son implémentation. En programmation informatique, on retrouve les routines sous deux formes principales : la procédure et la fonction.

variable

Pour manipuler des données, les langages de programmation offre le concept de variable. Une variable est un espace de stockage pour un résultat. Dans un programme, une variable est associé à un symbole (habituellement un nom qui sert d'identifiant) qui renvoie à une position de la mémoire (une adresse) dont le contenu peut prendre successivement différentes valeurs pendant l'exécution d'un programme. De manière générale, les variables ont un type : c'est la convention d'interprétation de la séquence de bits qui constitue la variable. Le type de la variable spécifie aussi sa taille (la longueur de cette séquence) soit habituellement 8 bits, 32 bits, 64 bits, ... Cela implique qu'il y a un domaine de valeurs (ensemble des valeurs possibles). Et elles ont aussi une valeur : c'est la séquence de bits elle même. Cette séquence peut être codée de différrentes façons suivant son type.

 © 2018 <tvaira@free.fr> v.1.0