



ADALOG

L'EXPERTISE ADA



La société

ADALOG est une société spécialisée dans l'expertise, le conseil et la formation dans les domaines du langage Ada, des méthodes de conception et du génie logiciel en général.

ADALOG a été fondée en 1985 par Jean-Pierre Rosen, expert du langage Ada et des techniques orientées objet. Il enseigne Ada depuis 1980 ; sa parfaite connaissance du langage alliée à un don pédagogique réputé procurent aux formations ADALOG une efficacité inégalée.

ADALOG intervient dans tous les domaines liés au langage Ada:

- **Expertise:** revues de code, analyse de problème, aide à la certification (DO178-B/C, EN50128), études d'outils.
- **Conseil:** Aide au développement, règles de codage, contrôle qualité.
- **Formation:** depuis le cours général du langage jusqu'aux domaines spécialisés : temps-réel, numérique, AWS, ASIS...
- **Validation:** ADALOG est un ACAL (Ada Compiler Assertion Laboratory), laboratoire officiellement agréé pour la validation de compilateurs Ada.

ADALOG n'est lié à aucun fabricant ni vendeur de compilateurs ou d'outils : c'est l'assurance d'une expertise et de conseils totalement libres et indépendants des fabricants.

Le langage Ada

Ada est un langage de programmation qui représente l'aboutissement de la lignée des langages "classiques", impératifs et procéduraux. Il constitue essentiellement un effort de synthèse des meilleurs éléments figurant dans les langages qui l'ont précédé, intégrés dans un ensemble cohérent. Ses normes successives, jusqu'à la toute récente version **Ada 2012**, suivent l'évolution de l'informatique en incorporant les notions les plus récentes :

programmation orientée objet, interfaces, **programmation par contrat**, support des architectures **multi-cœurs**... Il est utilisé avec succès dans des domaines aussi variés que le temps-réel, la gestion, la CAO, le médical, le traitement linguistique... Ada a été conçu d'après un **cahier des charges**, dont l'idée directrice est de **diminuer le coût** des logiciels, en tenant compte de tous les aspects du cycle de vie. Le langage est donc bâti autour de quelques idées-force :

- **Privilégier la facilité de maintenance sur la facilité d'écriture**, car la maintenance représente près des 2/3 des coûts d'un logiciel.
- **Fournir un contrôle de type extrêmement rigoureux**, permettant de diagnostiquer les erreurs le plus tôt possible.
- **Permettre une programmation intrinsèquement sûre**, en laissant le logiciel traiter **toutes** les situations anormales.
- **Etre portable entre machines d'architecture différentes**, afin de ne plus lier les logiciels à un constructeur.
- **Autoriser des implémentations efficaces et donner accès à des interfaces de bas niveau**, indispensables à la réalisation de systèmes embarqués, temps-réel, ou sécurisés.

Ada
2012

Ada cours complet

Le **cours de référence** pour ceux qui ne connaissent pas le langage, et souhaitent une présentation complète leur permettant de devenir opérationnels rapidement.

Ce cours d'une durée de 6 jours est constitué de deux sessions indissociables de 3 jours, réparties sur deux semaines. Cette formule originale permet une plus grande efficacité pédagogique en évitant la saturation de la formule traditionnelle sur 5 jours. Des travaux pratiques fréquents permettent de mettre en œuvre les notions dès qu'elles ont été présentées.

Le cours couvre la totalité d'Ada 83 et 95 (y compris les annexes) et présente les principales améliorations apportées par Ada 2005 et 2012.

Plan du cours:

Jour 1: Historique. Syntaxe. Déclarations, Instructions. Expressions. Entrées/Sorties simples.

Compilation. Types discrets. Types dérivés. Contrôle de la représentation. Sous-programmes.

Jour 2: Tableaux. Articles simples. Représentation des articles. Paquetages. Types privés et limités. **TP.** Unités hiérarchiques. Bibliothèque standard. Types à discriminants. Sous-unités. **TP.**

Jour 3: Exceptions. Traitement d'exceptions. Génériques et instanciations. Paramètres génériques. **TP.** Entrées/Sorties. Types accès. **TP.**

Jour 4: Programmation orientée objets. Types étiquetés. Héritage. Classes. **TP.** Facettes. Complément sur les types étiquetés. Surnommage. Règles de visibilité. **TP.**

Jour 5: Parallélisme. Tâches, activation et terminaison. Rendez-vous. **TP.** Familles d'entrées. Avortement. Parallélisme et exceptions. Types protégés. **TP.**

Jour 6: Interfaçages. SP assembleur. Autres paquetages standard. Annexes. Présentation des nouveautés Ada 2005/2012. Conclusion. **TP.**

Ada 2005/2012

Le **complément indispensable** pour tous ceux qui connaissent déjà Ada 95 et veulent suivre l'évolution récente du langage.

Ce cours d'une durée de trois jours présente toutes les nouvelles fonctionnalités apportées par la version la plus récente du langage. Il comporte des travaux pratiques utilisant le compilateur libre GNAT qui possède toutes les fonctions d'Ada 2012.

Plan du cours

Jour 1 : Historique. Wide_Wide_Character. Nouveaux pragmas. Clauses limited with, private with. Instantiations partielles. Notation objet, interfaces, extensions locales, contrôle de la redéfinition. Types limités initialisables. Types accès généralisés, non nuls. **TP.**

Jour 2 : Nouvelles bibliothèques: conteneurs, algèbre linéaire... Ordonnement et gestion du temps. Notification de terminaison. Interfaces synchronisées. Profil Ravenscar. **TP.**

Jour 3 : Aspects. Programmation par contrat. Itérateurs. Expressions conditionnelles, prédicats. Nouvelles fonctionnalités temps-réel, multi-coeurs. Extension de la bibliothèque. **TP.**

Formations à la demande

ADALOG peut organiser **toute formation** à la demande portant sur Ada, les méthodes du génie logiciel, et les techniques de développement (par ex: méthode **HOOD**, **AADL...**).

Les formations standard peuvent aussi être données en intra-entreprise, éventuellement adaptées pour répondre aux **besoins spécifiques** du client.

Toutes nos formations peuvent être données en français ou en **anglais**.