

# 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.



Adalog, membre du groupe Pacte-Novation



### Conseil

#### Utilisation du langage

ADALOG intervient pour **aider** des équipes confrontées à un problème particulier dans l'utilisation d'Ada, ou pour **conseiller** et **orienter** les choix quant à l'utilisation des nombreuses fonctionnalités du langage.

#### Règles de codage

Grâce au succès de **l'outil AdaControl**, ADALOG a développé une grande expertise dans l'établissement de règles de codage et la rédaction des manuels associés; cette expérience est proposée à ses clients pour leur permettre d'obtenir un ensemble de règles efficace et qui contribue effectivement à l'amélioration de la qualité des logiciels

#### Conception et suivi de projet

ADALOG est en mesure de fournir une **assistance** et des **conseils** lors de la mise en place de la conception préliminaire, ainsi qu'un **suivi du projet** pour assurer une base de départ correcte, vérifier la bonne mise en œuvre des méthodes orientées objet (COO, HOOD) et garantir une utilisation optimale du langage Ada lors de la réalisation.

### Tutorat

Lorsqu'une équipe de programmation a suivi un stage d'une ou deux semaines de formation à Ada, elle n'est pas encore capable de prendre seule la direction d'un projet. A mi-chemin entre la formation et le conseil, le tutorat permet de combler ce "trou" entre la formation initiale et l'obtention d'une équipe opérationnelle et autonome.

Concrètement, il s'agit d'une assistance technique au démarrage de projet couplée à une assistance pédagogique pour les points du langage insuffisamment assimilés. En expliquant les raisons qui l'ont amené à sa solution, le tuteur apporte le "fond culturel" qui permettra ensuite à l'équipe de prendre son autonomie. Il est toujours prêt à faire un cours technique lorsqu'il diagnostique une insuffisance de compréhension d'un point précis du langage.

### Développements

ADALOG peut se charger de tous **développements logiciels** au forfait en Ada, ou utilisant les techniques connexes.

ADALOG peut se charger du développement de **composants logiciels** spécifiques sur demande.

### Adassistance

Vous avez besoin d'un "coup de pouce" dans vos développements Ada ? Pour un montant forfaitaire, ADALOG vous propose le **contrat Adassistance** :

- Deux jours de **conseil sur place** pour bien connaître votre projet et faire une revue de conception.
- Un an de **hot-line Ada** pour répondre à toutes questions.
- Des **tarifs réduits** sur les journées de conseil supplémentaires.

Et si vous voulez passer à Ada 2005/2012, c'est le meilleur moyen de constituer rapidement votre nouveau savoir-faire.

### Assistance technique

En partenariat avec son associé NOVASYST Ingénierie, ADALOG peut mettre à disposition de ses clients des **ingénieurs spécialistes d'Ada** en régie.

Les personnes proposées par ADALOG ont toutes été **validées** techniquement par ADALOG, et ont pour la plupart suivi les formations ADALOG. En cas de difficulté technique dans l'exécution de leur mission, elles bénéficient du soutien et des conseils de J-P. Rosen.

### Études et expertises

Des missions d'**études**, d'**expertise**, d'**audit** et de **conseil** peuvent être traitées dans les domaines suivants:

- Langage Ada (conseil sur l'utilisation des fonctionnalités du langage, revues de code, assurance qualité).
- Règles de codage (développement de référentiel, contrôle automatique).
- Conception orientée objets (revues de conception, audit).
- Contrôle qualité (conception, développement, codage).
- Evaluation d'outils.
- Mise en place de plans de formation et de stratégies d'implantation d'Ada.
- Mise en place de politique de réutilisation et organisation de base de composants logiciels réutilisables.
- Autres langages orientés méthodologie.
- Systèmes d'exploitation, temps réel, techniques de compilation.

## 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... Ordonnancement 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**.



## L'outil de tous les contrôles

**AdaControl** est un outil libre (licence **GMGPL**) développé par **ADALOG** qui détecte la présence de toutes sortes de constructions dans les programmes Ada.

Son but premier est le contrôle de règles de programmation, mais c'est également un puissant outil pour rechercher l'utilisation (ou la non-utilisation) de nombreux modèles de programmation, *patterns*, etc.

Les éléments recherchés peuvent être très simples, comme l'occurrence de certaines entités, déclarations ou instructions, ou très sophistiqués, comme la vérification du respect de véritables "design patterns".

AdaControl est un **outil industriel**, utilisé notamment par Eurocontrol (test de plus de 2 millions de lignes de code), BelgoControl, Ansaldo, Alstom, Sagem, Thales... Il y contrôle le respect de règles de codage, participe à la certification des logiciels, et aide à la détection de problèmes complexes.

## Quelques exemples de règles vérifiées par AdaControl

- Les lignes sont limitées à 120 caractères de long.
- Les noms de types commencent par "T\_", sauf les noms de types accès qui commencent par "TA\_".
- L'instruction "goto" est interdite.
- L'instruction "case" doit être préférée à l'instruction "if" partout où cela est possible.
- Il est interdit de déclarer des tableaux dont la taille n'est pas déterminable à la compilation.
- Aucune exception ne doit se propager hors d'une tâche.
- Une instruction "accept" ne contient que des instructions dépendant directement de ses paramètres.
- Une même variable ne doit pas figurer en tant que paramètre réel correspondant à plusieurs paramètres formels de mode "out" ou "in out".
- Les sémaphores respectent un modèle qui garantit qu'aucun ne reste bloqué, même en cas d'exception.
- ...

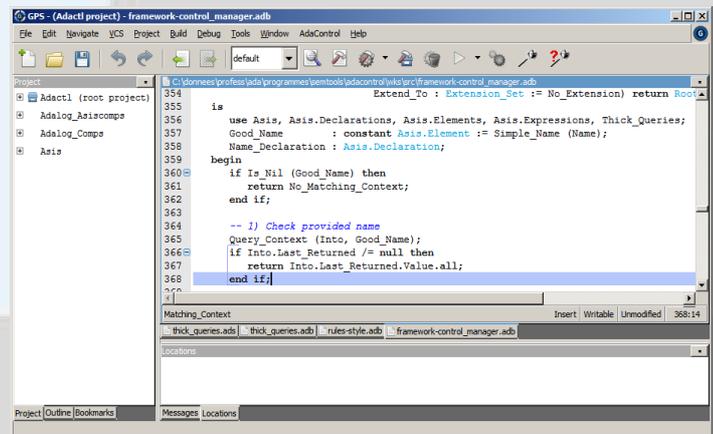
## AdaControl : l'offre commerciale d'ADALOG

→ **Contrat de maintenance**: Pour les utilisateurs réguliers d'AdaControl, ce contrat offre les avantages suivants:

- **Aide à l'installation** et à l'utilisation du produit.
- **Corrections des problèmes** rencontrés dans l'utilisation d'AdaControl, avec mise à disposition de versions "wavefront" pour chaque problème corrigé; compte dédié dans notre système de suivi de tickets pour un traitement rapide des problèmes.
- **Accès aux versions beta** avant qu'elles soient publiées.
- **Tarif préférentiel** pour le développement de règles à façon
- **Traitement prioritaire** des demandes d'amélioration.

→ **Développement de règles à façon**: ADALOG propose le développement au forfait de règles nouvelles, et en particulier de règles "métier" spécifiques du contexte du client. L'architecture d'AdaControl garantit la pérennité des règles ainsi développées à travers toutes les évolutions futures du produit.

→ **Aide à la définition de règles de codage**: Notre expérience de la problématique des règles de codage et de la vérification du code nous permet d'offrir des services de conseil et d'assistance pour la définition de standards de codage et de règles de programmation, avec mise en œuvre de la vérification avec AdaControl.



AdaControl s'intègre dans l'environnement GPS d'AdaCore