

Formations 2018

Formations 2018



Déclaration d'activité : 93 13 12834

- **Langages de programmation**
- **Méthodes**
- **Solutions**
- **Métiers**

Des formations :

- Dédiées à la **spécification de systèmes** ou au développement de **logiciels embarqués**
- Réalisées et prodiguées par des **experts seniors** qui vous font profiter de leur savoir-faire en conseil et réalisation de **systèmes critiques**
- Adaptées à **votre besoin**, offrant ainsi une réelle **plus-value**
- Inter ou Intra-entreprises

Calendrier des formations inter-entreprises

			Durée	Déc. 17	Janv.	Fév.	Mars	Avr.	Mai	Juin	Juil.	Août	Sept.	Oct.	Nov.	Déc.
Langages	Ada 95/05	Les bases	3	11-13	15-17	05-07	12-14	9-11	14-16	04-06	02-04		10-12	08-10	12-14	10-12
		Avancé	2	14-15	18-19	08-09	15-16	12-13	17-18	07-08	05-06		13-14	11-12	15-16	13-14
	Ada 2012	Les clés	2	18-19	22-23	12-13	19-20	05-06	24-25	11-12	09-10		17-18	15-16	19-20	17-18
	C	Pour systèmes contraints	5		29/01 – 02/02		05-09		28/05-01/06				24-28		26-30	
Méthodes	B	Mise en œuvre	3		10-12		21-23			13-15			19-21			
		Techniques de preuve	3			31/01–02/02			02-04					03-05		
Solutions	S3	Systemel Smart Solver - Les bases	2				26-27			18-19				18-19		
Métiers	SDF	Sûreté de fonctionnement	3	Nous consulter												
	EN 50128	Norme EN 50128:2011 - Nouveautés	1			01			22					04		
		Norme EN 50128:2011	2				29-30			21-22					22-23	



Ada 95 et 2005



Un langage moderne conçu aussi bien pour l'implémentation de programmes à grande échelle que pour des systèmes temps réels critiques.

Au travers de nombreux exemples et exercices, cette formation montre une utilisation concrète du langage Ada en milieu industriel.

Cette formation n'est pas spécifique à une implémentation donnée d'un compilateur, elle contient néanmoins de nombreuses références sur la façon de mettre correctement en œuvre la technologie GNAT (switchs compilateur, bibliothèques spécifiques, traits du langage liés à l'implémentation GNAT, outils connexes, etc.).

Matériel fourni :

- Support de cours en français relié
- Clé USB 16 Go contenant :
 - L'ensemble des exercices avec leur corrigé
 - Un compilateur Ada (Windows, Linux)
 - Des exemples vus en séance

Ada 95 et 2005 - Les Bases

Programme

- Rappels historiques
- Les caractéristiques du langage
- Le processus de développement
- Les éléments de base
- Les types composés
- Les exceptions
- Les pointeurs
- L'encapsulation et l'abstraction
- Les génériques
- Les hiérarchies d'unités
- Objets et classes
- Héritage, polymorphisme et liaisons dynamiques
- Abstraction et types abstraits
- L'environnement prédéfini (introduction)
- L'élaboration
- 21 recommandations
- Conclusion générale
- Bibliographie

6 exercices : le nombre PI, le nombre d'OR, Lena, abstraction d'une image, pile d'images, classes d'images abstraites.

Durée : 3 jours

Ada 95 et 2005 - Avancé

Programme

- Rappels
- La programmation concurrente
- Les tâches
- Les objets protégés
- Synchronisation et communication
- Ordonnancement et gestion du temps
- Interfaces synchronisées
- Interfaçage matériel et logiciel
- Sûreté de fonctionnement et certification (introduction)
- Le profil Ravenscar
- Environnements, méthodes et outils
- Conclusion générale
- Bibliographie

4 exercices :

- Tux (le producteur et le consommateur)
- Pou (le festin de fruits)
- Bsd 2 Tux
- Convolution d'une image

Durée : 2 jours

www.systemerel.fr



SPARK 2014

Au même titre que l'a été Ada 95, Ada 2012 est une évolution majeure du langage Ada. Comme ses « prédécesseurs » Ada 2012 met particulièrement l'accent sur de nouveaux traits permettant de garantir au mieux la sécurité et l'intégrité des applications développées tout en embrassant les nouvelles technologies telles que la technologie SMP.

A l'issue de cette formation, les participants auront les clés nécessaires pour une mise en pratique rapide et correcte du langage Ada 2012.

Au travers de nombreux exemples et exercices, cette formation montre une utilisation concrète du langage Ada en milieu industriel.

Cette formation n'est pas spécifique à une implémentation donnée d'un compilateur, elle contient néanmoins de nombreuses références sur la façon de mettre correctement en œuvre la technologie GNAT (switchs compilateur, bibliothèques spécifiques, traits du langage liés à l'implémentation GNAT, outils connexes, etc.).

Matériel fourni :

- Support de cours en français relié
- Clé USB 16 Go contenant :
 - L'ensemble des exercices avec leur corrigé
 - Un compilateur Ada (Windows, Linux)
 - Des exemples vus en séance

Ada 2012 - Les Clés

Programme

- Aperçu
- La programmation par contrat
- Expressions et prédicats
- Programmation concurrente et TR
- Itérateurs
- Extension de la bibliothèque prédéfinie
- Évolutions diverses
- SPARK2014 (introduction)
- Conclusion générale
- Bibliographie

4 exercices :

- Jeu d'échec
- Fractale de Mandelbrot
- Queue synchronisée
- Itérateurs
- Compression de Huffman

Durée : 2 jours



Le langage C



Créé par Brian Kernighan et Dennis Ritchie au début des années 70 pour l'écriture du système UNIX, ce langage se veut simple, efficace et portable.

C'est pour ces mêmes qualités que le langage C est toujours abondamment utilisé de nos jours dans l'industrie.

Lors de cette formation, l'ensemble des traits du langage ainsi que leur mise en œuvre dans des systèmes contraints seront présentés au travers de nombreux exemples et exercices.

Matériel fourni :

- Support de cours en français relié
- Clé USB 16 Go contenant :
 - L'ensemble des exercices avec leur corrigé
 - Un compilateur C (Windows, Linux)
 - Des exemples vus en séance

C pour systèmes contraints

Programme

- Les caractéristiques du langage
- Le processus de développement
- Éléments de base
- Les pointeurs
- Le passage de paramètres
- Les pointeurs de fonctions
- Les macros
- L'architecture d'une application
- La bibliothèque standard
- Les pièges du langage
- Les optimisations
- Les outils associés

Durée : 5 jours



Méthode B



La méthode B est une méthode formelle développée par J-R. Abrial et utilisée dans le domaine industriel pour le développement de systèmes complexes.

La méthode B permet un développement formel structuré en étapes jusqu'à l'obtention d'un code exécutable. Des mécanismes de preuve garantissent mathématiquement que le logiciel produit sera in fine conforme à sa spécification.

Les formations sont étayées par de nombreux exemples et exercices permettant une mise en œuvre concrète de la méthode B et des techniques de preuve dans un projet industriel.

Méthode B - Mise en œuvre

Programme

- Principes de la méthode B
- Principes d'un développement logiciel en B
- Modélisation des données, des exigences et des traitements en B
- Structurer un développement logiciel en B
- Appliquer les techniques de raffinement
- Principes de validation par la preuve

Durée : 3 jours

Méthode B - Techniques de Preuve

Programme

- Principes de la méthode B et applications dans le cadre d'un développement logiciel
- Preuve d'un projet B logiciel
- Preuve des lemmes au travers d'un prouveur interactif
- Ecriture et vérification des règles utilisateurs
- Enchaînement des activités liées à la preuve et à sa vérification
- Environnements et outils

Durée : 3 jours



Systemerel propose une gamme de solutions éprouvées industriellement pour l'analyse statique ou la preuve formelle de systèmes ou logiciels à l'aide de technologies à base de Model Checking.

Systemerel Smart Solver est un moteur d'analyse basé sur les technologies SAT. Ce moteur est notamment utilisé dans le cadre des vérifications de sécurité pour de nombreux systèmes, notamment dans les domaines ferroviaire ou aéronautique.

A l'issue de cette formation, les participants auront acquis des notions de Model Checking, sauront bâtir un modèle formel du système à vérifier, mettre en œuvre S3 pour vérifier le comportement de ce système, et auront compris comment S3 s'inscrit dans un processus certifiable.

Retrouvez les fonctionnalités offertes par Systemerel Smart Solver sur www.systemerel-smart-solver.fr

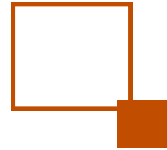
La formation est étayée d'exemples et d'exercices permettant une mise œuvre concrète de Systemerel Smart Solver en milieu industriel.

Systemerel Smart Solver - Les bases

Programme

- Introduction au Model Checking
- Principes du langage HLL
- Principes de modélisation avec HLL
- Les outils de Systemerel Smart Solver et leurs utilisations
- Présentation d'un processus certifiable avec Systemerel Smart Solver

Durée : 2 jours



La sûreté de fonctionnement

La Sûreté de Fonctionnement (SdF) a pour objectif de répondre aux exigences de FDMS (fiabilité, disponibilité, maintenabilité, sûreté ou sécurité-innocuité) contraignant les systèmes, logiciels et/ou matériels dits critiques et souvent soumis à certification de différents domaines sectoriels (ex. transport ferroviaire, aéronautique...), et ce en fonctionnement normal ou en cas de défaillance.

La SdF recouvre des notions et grandeurs caractéristiques qu'il est important de connaître afin de comprendre un besoin de sûreté et pouvoir y répondre de manière adaptée (i.e. en évitant sous-protection et sur-protection, toutes deux coûteuses). Elle repose sur des méthodes et des outils, qui peuvent être avantageusement utilisés pour évaluer les risques pour l'intégrité des personnes et des biens et déterminer les actions préventives et/ou correctives à mettre en œuvre afin de maîtriser ces risques, mais dont il faut également être conscients des limites.

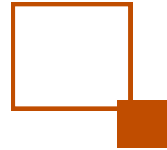
La formation est ponctuée d'exercices et propose un cas d'étude permettant une mise en pratique tutorée des connaissances théoriques présentées et facilitant ainsi leur assimilation.

La sûreté de fonctionnement

Programme

- Principales définitions et notions générales de la SdF
- Analyse de risque : identification du risque, caractérisation du risque, acceptation du risque et traitement du risque
- Approche SdF au niveau système / logiciel / matériel
- Méthodes et outils pour la SdF : analyse fonctionnelle, APR / APD, AMDEC / AEEL, arbre de défaillances...
- Dossier de sûreté (safety case)
- Modes communs et indépendance
- Prévention des fautes, élimination des fautes, tolérance aux fautes
- Définitions probabilistes de fiabilité (taux de défaillances, MTTF, MTBF), maintenabilité et disponibilité
- Composants non réparables / réparables

Durée : 3 jours



Norme EN 50128



Les logiciels pour systèmes de commande et de protection ferroviaire doivent répondre aux exigences de la norme CENELEC EN 50128. Toute personne impliquée dans la réalisation de ces logiciels du domaine ferroviaire est ainsi concernée par la compréhension de cette norme et sa mise en application, au travers de méthodes et techniques adaptées.

La norme EN 50128 régit la totalité du cycle de vie logiciel. On notera en particulier les fondamentaux significatifs suivants :

- Clauses relatives au développement du logiciel,
- Contraintes liées à l'assurance du logiciel (vérification, évaluation, qualité, ...),
- Exigences quant à l'organisation, la définition des rôles et des compétences des acteurs du logiciel.

A l'issue de cette formation, vous :

- Appréhendez la notion de SIL, concept - clé de la norme,
- Connaitrez les étapes et requis associés au cycle de vie,
- Maîtriserez les aspects organisationnels imposés par la norme.

La formation est ponctuée d'exercices ou d'exemples concrets d'applications logicielles permettant d'en appréhender au mieux la mise en œuvre.

Norme EN 50128

Programme

- Introduction générale de la norme EN 50128 et de sa finalité
- Présentation des étapes du cycle de vie du logiciel
- Analyse des aspects organisationnels
- Cas particulier des données et algorithmes d'application
- Challenges autour de l'application de la norme

Durée : 2 jours



Nouveautés de la norme EN 50128:2011

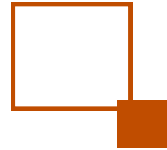
Les logiciels pour systèmes de commande et de protection ferroviaire doivent répondre aux exigences de la norme CENELEC EN 50128. Toute personne impliquée dans la réalisation de ces logiciels du domaine ferroviaire est ainsi concernée par la compréhension de cette norme et sa mise en application, au travers de méthodes et techniques adaptées.

La norme EN 50128 a évolué en 2011.

On notera en particulier les changements significatifs suivants :

- nouvelles exigences quant à l'organisation, la définition des rôles et des compétences des acteurs du logiciel,
- nouvelles clauses relatives aux outils, basées sur la norme CEI 61508-2:2010,
- mise à jour des tables de l'annexe A...

L'objectif de la formation est de présenter, clarifier et expliquer toutes les nouveautés introduites dans la nouvelle version de la norme. Elle est ponctuée d'exercices ou d'exemples concrets d'applications logicielles permettant d'en appréhender au mieux la mise en œuvre.



Nouveautés de la norme EN 50128:2011

Programme

- Présentation générale de la norme EN 50128:2011 et de sa finalité
- Principales évolutions entre l'ancienne version de la norme (EN 50128:2001) et la nouvelle
- Analyse de leur impact sur l'ensemble du cycle de vie du logiciel
- Moyens pour répondre aux nouvelles exigences et travailler efficacement avec la norme
- Challenges autour de l'application de la norme

Durée : 1 jour

Contact

Devis personnalisé

Informations

Réservation

Robert CHOLAY
Responsable Formations

formation@systemerel.fr

(+33) 4 42 90 41 20

FRANCE



PARIS
3 rue Danton
92240 Malakoff
+33 (0)1 76 60 40 26



AIX EN PROVENCE
1090 rue René Descartes
13100 Aix en Provence
+33 (0)4 42 90 41 20



TOULOUSE
2 imp. Michel Labrousse
31036 Toulouse
+33 (0)9 84 09 04 38

ALLEMAGNE



BERLIN
Laubstrasse 3
12045 Berlin
+49 (0)151 41 21 46 41

