

LabVIEW Fondamental 3

Description

LabVIEW Fondamental 3 vous présente des pratiques structurées pour concevoir, implémenter, documenter et tester des applications LabVIEW. Ce cours met l'accent sur le développement d'applications hiérarchiques extensibles, claires et faciles à entretenir. Les processus et techniques abordés dans ce cours vous aident à réduire la durée du développement et à améliorer la stabilité de votre application. En intégrant ces pratiques de conception dans les phases initiales du développement, vous évitez de reconstruire inutilement votre application, augmentez la réutilisation de votre VI et minimisez les coûts de maintenance.

Durée

En salle : 3 jours

En ligne : 4 sessions de 4 heures et travail personnel

Public

- Utilisateurs LabVIEW et NI Developer Suite qui ont besoin d'améliorer les performances, l'extensibilité ou la réutilisation de leur code et de réduire les coûts de maintenance de leurs applications
- Utilisateurs LabVIEW souhaitant obtenir la certification CLD (Certified LabVIEW Developer)

Pré-requis

- LabVIEW Fondamental 1 et 2, ou expérience équivalente

Produits NI utilisés pendant le cours

- Système de développement professionnel de LabVIEW

Après avoir suivi ce cours, vous pourrez :

- Développer une application avec agilité.
- Utiliser la fenêtre de l'Explorateur de projet pour améliorer l'organisation de vos fichiers et résoudre les conflits de fichiers.
- Utiliser le modèle de projet Gestionnaire de messages dans une file d'attente pour développer une application à plusieurs boucles.
- Utiliser des notificateurs pour communiquer d'une boucle à plusieurs entre des boucles parallèles.
- Développer un prototype d'interface utilisateur personnalisé pour les tests d'utilisabilité initiaux.
- Gérer les erreurs localement ou globalement en fonction de leur gravité.
- Développer, intégrer et tester des modules de code extensibles, clairs et faciles à entretenir.

Inscription

Inscrivez-vous en ligne sur ni.com/training ou appelez le 01 57 66 24 24 (Fax : 01 57 66 24 14) france.formation@ni.com

Hors de France, contactez votre filiale NI locale.
Informations sur les filiales : ni.com/global

Numéros de référence

910797-xx

- 01 Siège social de NI ou filiale
- 11 Régional
- 21 Sur site (dans vos locaux)
- 69 En ligne

Suggestions de cours à suivre par la suite

- Managing Software Engineering in LabVIEW
- Advanced Architectures in LabVIEW
- Object-Oriented Design and Programming in LabVIEW
- LabVIEW Performance

Certification recommandée

- Certification CLD (Certified LabVIEW Developer)

Programme du cours LabVIEW Fondamental 3

1er jour

Développement d'applications réussies

Cette leçon décrit différentes techniques de développement permettant de créer des VIs extensibles, clairs et faciles à entretenir. Vous comprendrez combien il est important de suivre un processus de développement logiciel et apprendrez à développer du code qui répond à toutes les exigences d'un projet.

Les points traités comprennent :

- VIs extensibles, clairs et faciles à entretenir
- Présentation générale des pratiques de développement d'un logiciel
- Exigences
- Analyse des tâches

Organisation du projet

Cette leçon décrit les outils et techniques d'organisation et de gestion des fichiers dans un projet LabVIEW. Vous apprendrez à créer et utiliser des bibliothèques de projet, à utiliser divers outils pour en savoir davantage sur chaque fichier de projet LabVIEW, et à identifier, résoudre et éviter les erreurs de références croisées et de conflits de fichiers.

Les points traités comprennent :

- Bibliothèques de projet
- Outils et organisation de l'Explorateur de projet
- Conflits dans un projet

2ème jour

Création d'une architecture d'application

Cette leçon présente des techniques et des pratiques de programmation permettant de créer des architectures intuitives et robustes pour de grandes applications. Vous comprendrez combien il est important de tester l'architecture de haut-niveau et de suivre des directives de style établies, et vous apprendrez à implémenter des événements utilisateur et des notificateurs et à utiliser des modèles de projet Gestionnaire de messages dans une file d'attente pour développer une application à plusieurs boucles.

Les points traités comprennent :

- Test de l'architecture
- Directives de style LabVIEW
- Événements utilisateur
- Gestionnaire de messages dans une file d'attente
- Types de données d'application
- Notificateurs

Personnalisation de l'interface utilisateur

Cette leçon présente des techniques permettant d'améliorer l'implémentation des faces-avant dans LabVIEW. Vous apprendrez à utiliser des menus d'exécution, barres de division, faces-avant secondaires et commandes onglets pour personnaliser et étendre votre interface utilisateur. Vous comprendrez l'importance de créer un prototype d'interface utilisateur et apprendrez des techniques pour améliorer l'utilisabilité de votre application.

Les points traités comprennent :

- Directives de style d'une interface utilisateur
- Prototypes d'interface utilisateur
- Personnalisation d'une interface utilisateur
- Extensibilité d'une interface utilisateur
- Apparence de la fenêtre
- Documentation utilisateur
- Initialisation de l'application
- Test de l'interface utilisateur

3ème jour

Gestion et enregistrement des erreurs dans un fichier

Cette leçon décrit plusieurs approches pour développer des logiciels répondant de manière contrôlée à différents types d'erreurs. Vous apprendrez à déterminer si une erreur doit être gérée localement ou globalement et quand enregistrer les données d'erreur sur disque pour analyse ultérieure.

Les points traités comprennent :

- Test des erreurs
- Gestion d'erreur locale
- Gestion d'erreur globale
- Enregistrement des erreurs

Création de code modulaire

Cette leçon décrit comment utiliser du code modulaire dans une grande application et présente des règles pour faciliter la maintenance des grandes applications. Vous découvrirez plusieurs approches pour tester des modules de code et les intégrer à l'architecture de haut niveau de votre application.

Les points traités comprennent :

- Conception d'applications modulaires
- Test des modules de code
- Tests d'intégration