

Gérer les dépendances entre composants logiciels en systèmes

Description de la formation

Mieux cibler les principes orientés objet nécessaire pour intégrer de manière adéquate des techniques avancées afin de réduire les dépendances entre composants logiciels en systèmes embarqués. Ce cours débute par un bref rafraîchissement sur la compréhension essentielle des principes orientés objet de base nécessaire pour adéquatement intégrer des techniques de pointe pour briser les dépendances entre les composants logiciels dans les systèmes embarqués. Ces techniques sont particulièrement importantes quand une dépendance externe tel qu'un composant matériel, un composant logiciel, ou même un sous-système, ne sont pas encore prêts pour l'intégration (par exemple, pas encore entièrement testé). Ce cours présente en profondeur l'utilisation de refactoring, de stubs, et de design patterns les plus importants pour être en mesure de les appliquer rapidement dans vos projets en systèmes embarqués.

Objectifs pédagogiques

1. Appliquer rapidement les meilleures techniques de gestion des évolutions et des dépendances entre composants (matériel ou logiciel) qui ne sont pas encore prêts ou pas entièrement testés pour une intégration.
2. Appliquer des tests unitaires de base et des techniques de refactoring nécessaires pour réduire les dépendances.
3. Améliorer la structure, l'organisation et la stabilité dans vos projets en systèmes embarqués.
4. Améliorer la conception de ses applications pour faciliter leur maintenance et leur évolution.

Contenu

- Énumération et description des symptômes d'une mauvaise conception
- Comment les dépendances entravent les efforts de maintenance, rendent difficile l'extensibilité, empêchent la réutilisation et les tests
- Clé du problème : la gestion des dépendances
- L'apport de la restructuration (refactoring) vs les design patterns
- L'importance de l'émulation et des stubs en systèmes embarqués
- L'importance et utilisation des fabriques (Factories), des registraires (Registries), de l'injection de dépendance (Dependency Injection) et de l'inversion de contrôle (Inversion of Control)
- Mise en oeuvre incremental de tests unitaires à l'aide d'un outil de tests léger et indépendant de tout langage de programmation applicable pour systèmes embarqués à faibles ressources

Méthodologie

Exposé interactif (40%), laboratoires (60%)

Clientèle visée

Professionnels et techniciens spécialisés en systèmes embarqués (portion développement logiciel)

Particularités

Le participant doit avoir de l'expérience en C/C++, ou avec un langage orienté-objet. Les laboratoires seront effectués en C/C++.

Formateur(s)

Gérer les dépendances entre composants logiciels en systèmes

Michel de Champlain

Michel possède 36 ans d'expérience dans la programmation, dont 29 ans avec le langage C++, 20 ans avec Java et 14 ans avec C#. Depuis 25 ans, il est un conférencier pour les systèmes embarqués (embedded systems) et il enseigne les technologies objet depuis 1988. Il a déjà formé plusieurs milliers de personnes au Québec, au Canada, aux États-Unis, en Europe et en Nouvelle-Zélande. Michel est responsable du créneau Développement logiciel embarqué chez ÉTS Formation et a conçu le langage objet B# (BSharpLanguage.org) pour faciliter le développement dans ce domaine.

Durée

1 jour(s)

Coût par participant en formation publique

477 \$

Prochaines dates en formation publique

22 novembre 2019 (Montréal)

26 février 2020 (Montréal)

ÉTS FORMATION est le leader universitaire en formation continue avec plus de 7 000 participants formés annuellement et une offre de plus de 300 différentes formations. Nos formations sont pratiques et pragmatiques et affichent un taux de satisfaction supérieur à 90 %. Consultez notre programmation complète au <http://www.etsformation.ca/>