Solti

Formations 2012

Table des matières

1 Introduction 3
1.1 Formations SOLTI 3
1.2 Prérequis 3
1.3 Agenda 3
2 Module 1 : Bases 4
2.1 Technologies 4
2.2 Méthodes 4
3 Module 2 : Langage C 5
3.1 Déclarations 5
3.2 Expressions 5
3.3 Chaînes de caractère 5
3.4 Pièges et astuces 5
3.5 Règles de programmation 5
3.6 Architecture 5
3.7 MISRA 5
3.8 Optimisation de code 5
3.9 Gestion mémoire 5
4 Module 3 : Approche Objet et UML 6
4.1 Approche Objet 6
4.2 Le langage UML 6
4.3 Exercices 6
4.4 Star UML 6
5 Module 4 : Langage C++ 7
5.1 Spécificités C++ 7
5.2 Les entrées sorties 7
5.3 Les classes 7
5.4 Héritages simples et multiples 7
5.5 Fonctions virtuelles 7
5.6 Gestion des exceptions 7
5.7 Templates 7
6 Module 5 : Langage C# 8
6.1 Introduction 8
6.2 Visual 8
6.3 Préprocesseur 8
6.4 Déclarations 8
6.5 Flux 8
6.6 Objet 8

Solti

Formations 2012

7 Module 6 : Applications C# 9
7.1 Forms 9
7.2 Architecture Vue-Données 9
7.3 Communication TCP/IP 9
7.4 Accès aux données 9
8 Module 7 : Gestion de projets 10
8.1 Bases 10
8.1.1 Introduction 10
8.1.2 Etapes 10
8.1.3 Management 10
8.1.4 Outils 10
8.2 Projets logiciels 10
8.2.1 Introduction 10
8.2.2 Méthodes et outils 10
8.2.3 Tests 10
9 Module 8 : Méthodes agiles 11
9.1 Introduction 11
9.2 Agilité 11
9.3 Le développeur 11
9.4 Scrum 11
9.5 Extreme Programming 11
9.6 Recettes 12
9.7 Gestion du changement 11
9.8 Règles 11
9.9 Mise en pratique 12
9.9.1 Plannification 12
9.9.2 Gestion des sources 12
9.9.3 Les tests 12
9.9.4 Coding dojo 12
10 Module 9 : Logiciel embarqué 13
10.1 Architecture de RTOS 13
10.2 Ethernet 13
10.3 TCP/IP 13
10.4 USB 13

Solti

Formations 2012

1 Introduction

1.1 Formations SOLTI

Dans un esprit d'efficacité, les formations SOLTI ne sont pas des produits standards et normalisés. Les objectifs d'une formation sont variés, les motivations et les compétences des stagiaires également.

La formation doit aider le stagiaire, qui est dans une situation particulière (capacités, expérience, compétence, disponibilité, motivation) à évoluer vers une nouvelle situation qui doit être définie aussi précisément que possible avant la formation.

Aussi, le programme sera adapté, dans le contenu et dans la forme, aux conditions réelles de formation et à l'attente des stagiaires. Le contenu de la formation est donné ci dessous à titre indicatif et ne constitue pas au chapitre près à un engagement de SOLTI. Le totalité du programme représente environ 15 jours de formation. Nous avons réalisé depuis 2009 plus de 300 jours stagiaire.

1.2 Prérequis

Le client fournit

- Un PC par élève avec Visual Studio ou Eclipse installé
- Une salle aux dimensions adaptée au nombre de participants
- Un vidéo projecteur
- Un tableau blanc avec ses stylos
- Une connexion Internet pour le formateur

1.3 Agenda

La durée effective des cours est de 6 heures quotidiennes minimum.
Pour le confort et la flexibilité, il est bon de ne pas faire plus de 2 journées à la suite suite.
Le contenu précis des formations dépend des objectifs fixés, du temps disponible et du temps passé sur les exercices. On essaie toutefois de tenir les horaires suivants, en alternant théorie et exercices :

Matin:	9h30 - 11h	pause	11h15 - 12h45
Après midi :	13h30 -15h15	pause	15h30 - 17h

Solti

Formations 2012

2 Module 1 : Bases

2.1 Technologies

Architecture d'un ordinateur
Architecture d'un processeur
Code machine
Langage assembleur
Macros
Procédures
Langage C
Compilateur
Editeur de lien
Runtime
Debugger
Entrées sorties
Fichiers
Liaison série
Ethernet

2.2 Méthodes

Ingénierie: Du besoin à la validation
Les modèles
Les méthodes
Cycle de vie d'une application
Méthodes traditionnelles: en cascade
Méthodes traditionnelles: en V
Le cycle de vie théorique
Les cycles de vie réels
La méthode itérative
Les outils UML
L'intérêt d'UML

Solti

Formations 2012

3 Module 2 : Langage C

3.1 Déclarations

Structure d'un programme
Les fonctions
Les variables
Les types simples
Caractères
Entiers
Flottants
Void
Tableaux
Structures
Enumérations

3.2 Expressions

Expressions
Les pointeurs
Opérateurs simples
Opérateurs complexes
Les boucles if
Les switches
Les boucles for
Les boucles while
Les autres boucles
Break, return continue

3.3 Chaînes de caractère

Principe
Utilisation
Librairies
Usages et abus
Les solutions
Exercice : codage d'une bibliothèque string

3.4 Pièges et astuces

Les fautes de frappe
Récursivité
Conversions de type
Promotion des entiers
Manipulation de pointeurs
Réentrance
Opérateur &
Complexité du code
Allocation dynamique
Tableaux et pointeurs
Les unions
Les champs de bits
Tests unitaires
Exercice : comparaison des codes générés

3.5 Règles de programmation

Gestion des sources
Versions
Sauvegardes
Codage
Structuration
Nommage
Commentaires
Divers

3.6 Architecture

Systèmes asynchrones
Systèmes synchrones
Bonnes pratiques
Architecture 3 tiers
Modèle Vue Contrôleur
Threads
Tâches
Traçabilité
Portage
HAL
Norme Posix
Définition de la cible
Définition du processeur
Librairie portable
Kernel

3.7 MISRA

Inventaire
Les règles simples
Les règles utiles
Les règles pénibles
Les règles peu critiques

3.8 Optimisation de code

Introduction
Métriques de performance
Spécifications
Criticité

3.9 Gestion mémoire

Vitesse
Génération de code
Design patterns
Bonnes pratiques
Exercice : optimisation de performance

Solti

Formations 2012

4 Module 3 : Approche Objet et UML

4.1 Approche Objet

Problématique: Taille et complexité des logiciels
Solutions: Descartes
Solutions: Le découpage
Concepts
Terminologie
Manipulations d'objets
Les messages
L'encapsulation
L'abstraction
L'héritage : Spécialisation
L'héritage : Généralisation
Classes abstraites
Conclusion

4.2 Le langage UML

Les use cases
Fonctionnalités du système
A quoi ca sert
Les acteurs
Description
Relation extend
Relation include
Les scénarios
Séquences d'interactions
Diagramme de séquence
Diagramme de collaboration
Schémas
Objets
Classes
Association
Rôle
Agrégation
Cardinalité
Qualificateur
Généralisation et spécialisation
Diagrammes d'états
Définitions

4.3 Exercices

Parcours d'une motrice
Trajets
Géométrie
Gestion de stock
Distributeur de billets

4.4 Star UML

Une IDE style Visual
Créer un projet
Choisir un modèle
Ajouter un diagramme de classes
Editer un diagramme de classes
Ajouter une classe
Spécialiser la classe
Ajouter des attributs
Modifier un attribut
Afficher les propriétés
Diagramme d'état
Générer le code
D'autres outils

Solti

Formations 2012

5 Module 4 : Langage C++

5.1 Spécificités C++

Commentaire fin de ligne
Déclarations et initialisations
Notion de référence
Arguments par défauts
Surcharge de fonctions
Opérateurs New et Delete
Incompatibilité entre C et C++
Exemples

5.2 Les entrées sorties

Les entrées sorties standards
Affichage à l'écran
Lecture au clavier
Les flots
Exemples

5.3 Les classes

Propriétés des fonctions membres
Construction, destruction et initialisations d'objets
Fonctions amies
Surcharge d'opérateurs
Conversions de types
Exemples

5.4 Héritages simples et multiples

Notion d'héritage
Utilisation, surcharge des membres
Appel des constructeurs et destructeurs
Mise en œuvre

5.5 Fonctions virtuelles

Mécanisme
Propriétés
Fonctions virtuelles pures
Exemples

5.6 Gestion des exceptions

Mécanisme de gestion des exceptions
Choix du gestionnaire
Les exceptions standards
Exemples d'exceptions

5.7 Templates

Création et utilisation
Les paramètres de type
Les paramètres expressions
Spécialisation
Exemples

Solti

Formations 2012

6 Module 5 : Langage C#

6.1 Introduction

Historique
Avant
.Net
.Net 3.0
.Net 3.5
.Net 4.0
En bref
Proche de java
Proche de C++
Les nouveautés
Les plus
Les moins
Références Web
Références papier

6.2 Visual

Application console
Désassembleur
Librairie : Création
Librairie : Utilisation
Application forms : Dessiner
Application forms : Coder
Application forms : Debugger
Conclusion

6.3 Préprocesseur

Définitions
Régions
Pragma
Commentaires
Doc XML
Balises

6.4 Déclarations

Les espaces de noms
Les variables
Les constantes
Les types valeurs
Les types nullables
Les types référence
Les types de base
Les énumérations
Les conversions
Les tableaux unidimensionnels
Tableaux à 2 dimensions
Structures
Sécurité du code

6.5 Flux

Les opérateurs
Tous les opérateurs
Branchements if
Branchements switch
Boucles simples
Boucles moins simples
Les exceptions : émettre
Les exceptions : capturer

6.6 Objet

Les classes
Exemple de classe
Les méthodes
Appel de méthode
Passage de paramètres
Surcharge
Liste de paramètres
Propriétés
Indexeurs
Héritage
Classe abstraite : méthode abstraite
Classe abstraite : propriété abstraite
Surcharge des opérateurs
Interfaces : définir
Interfaces : utiliser

Solti

Formations 2012

7 Module 6 : Applications C#

7.1 Forms

Architecture forms
Buttons
EditBox
ListBox
ComboBox
TabPanels
Control forms
Class libraries
Intégration de DLLs
Intégration d'objets COM

7.2 Architecture Vue-Données

Bonnes pratiques Windows
Paramètres
Timers
Delegates
Threads
Traçabilité
Gestion de fichiers
Fichiers de log

7.3 Communication TCP/IP

Client server
Principe
Mise en œuvre
Exemple

7.4 Accès aux données

Classe DataTable
Classe DataGridView
Bibliothèque Office et fichier Excel
ODBC et fichier Excel
ODBC et accès SQL

Formations 2012

8 Module 7 : Gestion de projets

8.1 Bases

8.1.1 Introduction

Industrie
Disciplines transverses
Produire quoi ?
Critères de qualité
Coûts directs et indirects
Exigences qualité

8.1.2 Etapes

Analyse du besoin
Analyse fonctionnelle
Cahier des charges
Mise en œuvre

8.1.3 Management

Qualité d'un projet
Management
Planification
Contrôle
Chef de projet
Maître d'ouvrage
Maître d'œuvre

8.1.4 Outils

PERT
Gantt
Kamban
Réunions
Documentation

8.2 Projets logiciels

8.2.1 Introduction

Sensibilisation
La réalité en chiffres
Personne n'est parfait
Projet logiciel
Triangle des Bermudes
Le cycle de vie
Cycle en V, idéal
Cycle en V, réel
Autres cycles
Modèle en spirale
Comparaison des cycles

8.2.2 Méthodes et outils

Pathologies
Les algorithmes
Complexité du langage
Les tests
Quels tests ?
Gestion de code
La documentation
Critères de qualité
Recettes
Modularité
Conclusion 1
Conclusion 2

8.2.3 Tests

Cycle de vie et documentation
Encadrer les travaux
La fiche de test est éternelle
Par exemple…
Gantt

Solti

Formations 2012

9 Module 8 : Méthodes agiles

9.1 Principes

1.1.1 Introduction

Historique
Gestion de projet
Qualité
Le Lean
L'agilité

1.1.2 Agilité

Problématique: Pourquoi ? Comment ?
Effet tunnel : Méthode incrémentale
Erreurs de specs: Méthode itérative
Gestion commerciale: Méthode collaborative
Les critiques !
Historique
Structuration ou adaptation
Le manifeste agile : les 12 principes
Agilité, Lean et amélioration continue
La documentation

1.1.3 Gestion du changement

Problématique
Solution lourde
Solution lean
Documentation
FAE
Conclusion

1.1.4 Le développeur

La réalité en chiffres
Personne n'est parfait
Triangle QCD des Bermudes
Le cycle en V Virtuel
Le cycle en R Réel
Pathologies du développeur
Qu'est ce qu'un bel algorithme ?
Complexité du langage
Les tests
Quels tests ?
Documentation
Critères de qualité
Recettes
Modularité
Conclusions

9.2 Méthodes

1.1.5 Scrum

Mêlée
Rugby
Principes
Origines
Utilisateurs
Utilisations
Caractéristiques
Manifeste agile
Cycle
Sprints
Activités
Stabilité
Rôles
Cérémonial
Artifacts
Références
Ce qu'il faut retenir

1.1.6 Extreme Programming

Historique
Principes
Méthode
Intégration continue
Pair programming
Test driven
Pair testing

1.1.7 Règles

Gestion des sources
Versions
Sauvegardes

Codage
Structuration
Nommage
Commentaires
Divers

1.1.8 Recettes

Planning Poker
Non régression
Psychologie du programmeur
Google Programming
Gestion des conflits
La journée d'un XP

9.3 Pratique

9.3.1 Plannification

Les sprints
Les réunions
La documentation
Les livrables

9.3.2 Gestion des sources

Incrémental
Repository
Méthodologie
Le build
La livraison

9.3.3 Les tests

Test unitaire
Framework
Test fonctionnel
Non régression

9.3.4 Coding dojo

Gestion d'un micro projet XP
Formation des binômes
Codage alterné
Bilan.
Discussions.

Solti

Formations 2012

10 Module 9 : Logiciel embarqué

10.1 Architecture de RTOS

Tâches
Co routines
Queues
Sémaphores
Compteurs Sémaphores
Mutexs
Conclusion

10.2 Ethernet

Principe
Les différentes normes
La topologie
Les performances
Communication UDP
Mise en œuvre avec Visual C

10.3 TCP/IP

Principe
Client serveur
Réalisation d'un serveur avec Visual C
Réalisation d'un client avec Visual C
Gestion des déconnexions

10.4 USB

Principes
USB 1.0
USB 2.0
Implémentations