Formation Concepteur/Développeur informatique

Durée :	5 jours
Public :	Informaticiens souhaitant se réorienter vers le développement - Non informaticiens de filières scientifiques ou techniques avec des notions de programmation
Pré-requis :	Notions d'algorithmique
Objectifs :	Comprendre le cycle de développement d'une application, connaître les technologies du marché et orienter son choix, implémenter du code en C# ou Java ou C++
Sanction :	Attestation de fin de stage mentionnant le résultat des acquis
Taux de retour à l'emploi:	Aucune donnée disponible
Référence:	PRO928-F
Note de satisfaction des participants:	Pas de données disponibles

Découvrir le cycle de développement d'une application et les outils

Conception applicative, plateformes, choix du langage par type d'applications
Contraintes d'interfaces : client lourd, léger, mobile
Analyse fonctionnelle, prototypage, modélisation UML
Composants de la couche métier : composants, services web,...
Tests : types (unitaires, fonctionnels), développement piloté par les tests, outils
Processus de packaging d'une application, livraison, mise à jour
Environnement de développement : outils, suivi de versions, intégration continue

Atelier : choix d'un langage (C++, Java ou C#), mise en place de l'environnement de développement

Maîtriser les bases

Utilisation de variables, constantes, opérateurs
Types simples et types références
Transtypage, Wrappers
Expression de conditions : if/else, switch, opérateur ternaire
Utilisation de boucles : for, while, do while
Manipulation de tableaux
Factorisation de codes avec méthodes
Surcharge, arguments variables, récursivité
Commenter et documenter du code

Atelier : Multiples exemples de manipulation de structures de contrôles et de fonctions

Apprendre l'objet

Définition de classes
Déclaration des membres d'instance / de classe (static)
Constructeurs et instanciation
Cycle de vie d'un objet en mémoire
Diagramme de classes (UML)
Agrégation d'objets (association)
Encapsulation : getters et setters / propriétés
Extension de classes (Héritage)
Comparaison d'objets
Abstraction
Polymorphisme

Atelier : Modélisation et implémentation objet d'applications

Gérer les exceptions

Définition, types d'exceptions
Capturer et traiter une exception (try/catch/finally)
Lever/Remonter une exception (throw/throws)
Création d'exceptions

Atelier : Gestion des exceptions susceptibles d'être déclenchées dans une application

Utiliser des collections

Présentation des APIs disponibles, generics
Comparatif, choix d'un type de collection
Classes essentielles : listes, tables de hachage, ...
Parcours, opérations sur des collections et tris

Atelier : Manipulation de collections d'objets

Manipuler des fichiers

Lecture et écriture de fichiers
Manipulation de chemins, répertoires
Externalisation de configuration dans des .properties
Gestion des logs dans une application

Atelier : Implémentation d'exports et imports depuis des fichiers

Accèder à des bases de données

Présentation des APIs disponibles
Ecriture de requêtes SQL, exécution et traitement des résultats
Gestion des transactions
Introduction au mapping relationnel objet (pattern DAO)

Atelier : Organisation et implémentation d'une couche d'accès aux données

Construire des interfaces graphiques

Présentation des APIs disponibles
Fenêtres modales/non modales, boîtes de messages
Positionnement des contrôles
Gestion des évènements : claviers, souris

Atelier : Construction de fenêtres et implémentation d'évènements