

Introduction à l'algorithmique (avec Visual Basic)

Cours Pratique de 3 jours - 21h

Réf : INP - Prix 2024 : 1 870CHF HT

Ce cours vous initiera aux bases de la programmation et de l'algorithmique. Vous mettrez en œuvre les différentes étapes clés de la construction d'un programme informatique au travers du langage Visual Basic. Vous découvrirez les éléments de lexique et de syntaxe, les outils, l'organisation du code et les tests.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Structurer des programmes selon un algorithme

Maîtriser les éléments de lexique et de syntaxe d'un langage pour écrire un programme

Compiler et exécuter un programme

Déboguer et tester un programme

Accéder à une base de données

Comprendre les grands principes de la Programmation Orientée Objet

MÉTHODES PÉDAGOGIQUES

Pédagogie active basée sur des exemples, des démonstrations, des partages d'expériences, des cas pratiques et une évaluation des acquis tout au long de la formation.

TRAVAUX PRATIQUES

Ce cours d'algorithmique contient plus de 60 % de travaux pratiques effectués avec le langage Visual Basic. Cependant, il ne s'agit pas d'un stage VB.

LE PROGRAMME

dernière mise à jour : 07/2021

1) Les fondements de la programmation

- Qu'est-ce qu'un programme ? Qu'est-ce qu'un langage ? Les différents paradigmes. Quel langage pour quelle application ?
- Les compilateurs. Les exécutoires.
- Les responsabilités d'un programmeur.
- Qu'est-ce qu'un algorithme ?
- Les besoins auxquels répond un algorithme.
- Le concept de pseudo-langage.

Travaux pratiques : Présentation de différents langages (Java, C#, Visual Basic, C, C++).

Écriture d'un premier algorithme en pseudo-langage.

2) Genèse d'un premier programme

- Écriture d'un programme simple : syntaxe et instructions.
- Compilation et exécution du programme.
- Qu'est-ce qu'une librairie ? Son rôle, son usage.

Travaux pratiques : Découverte de l'environnement de développement et d'exécution.

Écriture, compilation et exécution d'un premier programme.

3) Règles de programmation

- Convention de nommage.
- Convention syntaxique.
- Utilisation des commentaires. Pourquoi commenter les développements ?
- Améliorer la lisibilité des programmes : indentation du code, découpage du code...

PARTICIPANTS

Toute personne devant apprendre à programmer.

PRÉREQUIS

Aucune connaissance particulière.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

4) Les variables

- Qu'est-ce qu'une variable ?
- Pourquoi typer une variable ?
- Les types primitifs : entiers, chaînes de caractères, nombres réels, autres.
- Déclaration, définition et initialisation d'une variable.
- Les constantes.
- Saisie, affichage, affectation, conversion de type.
- Organiser ses données sous forme de tableaux.
- Les types évolués : enregistrement, matrice, arbre.

Travaux pratiques : Ecriture de plusieurs programmes simples manipulant les variables.

5) Opérateurs et expressions

- Les différents opérateurs (multiplicatif, additif, comparaison, égalité, logique, affectation).
- Combinaison d'opérateurs.
- Expression booléenne.

Travaux pratiques : Manipulation des opérateurs et des expressions booléennes.

6) Les structures de contrôle

- Les sélections alternatives (si, si-alors-sinon, sélection cas).
- Les blocs d'instructions (notion de début... fin).
- Les boucles itératives (tant que-répéter, répéter-jusqu'à, pour-de-à).
- Imbrication des instructions.
- Les commentaires.

Travaux pratiques : Utilisation des structures de contrôle pour implémenter un algorithme.

7) Les procédures et les fonctions

- Définitions : procédure, fonction.
- Pourquoi sont-elles incontournables en programmation (réutilisabilité, lisibilité...) ?
- Le passage de paramètres.
- Le code retour d'une fonction.
- Sensibilisation aux limites du passage de la valeur d'une variable.
- Notion de passage par adresse.
- Appel de fonctions.

Travaux pratiques : Debugging de programmes exemples.

8) Introduction à la programmation Objet

- Les concepts associés à la programmation Objet : classe, attribut, méthode, argument.
- La modélisation objet à partir des exigences fonctionnelles.
- Introduction aux bonnes pratiques d'organisation de conception et d'organisation d'un programme.

Travaux pratiques : Illustration des concepts Objets.

9) L'accès aux bases de données

- Organisation et stockage des données.
- Les traitements de base (connexion, requêtes, récupération des données).
- Application cliente et serveur de données.
- Affichage et manipulation des données dans l'application cliente.

Travaux pratiques : Création d'un formulaire de recherche d'informations dans une base de données.

10) Maintenance, débogage et test des programmes

- Savoir lire et interpréter les différents messages d'erreurs.
- Utiliser un débogueur : exécuter un programme pas à pas, points d'arrêts, inspecter les variables pendant l'exécution.
- Prévoir les tests unitaires.

Travaux pratiques : Utilisation d'un débogueur pour contrôler l'exécution des programmes.

LES DATES

CLASSE À DISTANCE

2024 : 18 sept., 16 déc.