Catalogue des formations

Leuville Objects

Architectures Informatiques Innovantes

2019

www.leuville.com

Leuville Objects

Architectures Informatiques Innovantes

Sommaire

Filière Architecture logicielle **page 3**
Développer des flux avec Apache Nifi page 4
Développer des flux avec Apache Camel page 5

Filière DevOps **page 6**
Essentiel GIT, mettre en oeuvre le contrôle de versions page 7
Essentiel Docker, mise-en-oeuvre page 8

Filière BigData **page 9**
Réaliser des applications Apache Spark pour le Big Data page 10
Maîtriser les bases du langage Python. page 11
Développer avec Scala. page 12

Filière C++ **page 13**
Développer en C page 14
Développer en C++ - Niveau 1 page 15
Développer en C++ 11/14/17 page 16
Développer en C++ - Niveau 2 page 17
Développer avec Qt page 18

Filière Framework Spring **page 19**
Développer des applications Java avec Spring page 20
Automatiser le déploiement avec Spring Boot page 21
Développer des micro-services avec Spring et Docker page 22

Filière Internet des Objets **page 23**
Essentiel IoT, panorama Internet des Objets. page 24
Essentiel LoRaWAN, émettre et traiter des messages IoT. page 25
Développer des applications IoT/LoraWAN en Java et C++. page 26
Développer des applications IoT/LoraWAN en Python. page 27

Filière JavaEE **page 28**
Java Server Faces 2.X page 29
Composants métiers JavaEE : les EJB 3.0 page 30
Développer des Web Services SOAP en Java page 31
Développer des services RESTful en Java page 32
Exploiter les spécificités JavaEE 6 page 33
Java Persistence API page 34

Filière Java, langage et outils **page 35**
Essentiel Java 8, lambdas et streams page 36
Les bases du langage Java page 37
Mettre en place des tests en Java page 38
Gérer le cycle de vie d'un projet Java avec Maven page 39

Filière .Net CORE **page 40**
Programmer avec C# page 41

Filière Android et iOS **page 42**
Développer des applications Android avec Kotlin page 43
Moderniser vos apps Android avec Kotlin et les meilleures toolkits page 44
Développer sous Android page 45
Développer pour iOS avec Swift page 46

Filière Modelisation **page 47**
Essentiel Objet, les concepts et la pratique page 48
Savoir modéliser avec UML page 49
Modéliser des processus avec BPMN 2.0 page 50

Filière --> Essentiels 1 jour **page 51**

Filière --> Sur demande **page 52**
Les concepts Objet page 53
Design Patterns et éléments de conception page 54
Mapping Objet/Relationnel avec Hibernate page 55
Essentiel Web Java, Servlets et JSP page 56
Développement JavaScript avec AngularJS page 57
Développement Web avec HTML5 page 58
Mise en forme HTML avec CSS page 59
Développement JavaScript page 60
Développement d'applications Web 2.0 avec JQuery page 61

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Architecture logicielle

Descriptif de la filière

Cette filière propose les formations sur les pratiques et outils permettant de mettre en place une architectures logicielle d'entreprise moderne et efficace.

Liste des formations de cette filière

Intitulé	Durée	Description
Développer des flux avec Apache Nifi	2 jours	Cours pratique de prise en main du routeur logiciel et moteur de workflow Apache Nifi
Développer des flux avec Apache Camel	2 jours	Cours pratique de prise en main du routeur logiciel et moteur de workflow Apache Camel

Formations rattachées

Intitulé	Durée	Description
Essentiel Docker, mise-en-oeuvre	1 jour	Cours pratique de présentation des notions de "conteneurisation" et leur utilisation dans l'environnement Docker
Développer des micro-services avec Spring et Docker	3 jours	Cette formation permet de mettre en place des micro-services avec Spring et Docker.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des flux avec Apache Nifi

Code
CP-NIFI
Durée
2 jours
Tarif
1290 € HT
Sessions
28/03/2019 au 29/03/2019
11/06/2019 au 12/06/2019
07/11/2019 au 08/11/2019

Description

Le routage des données et le traitement par flux est au coeur de l'intégration d'applications d'entreprise (EAI). On le retrouve également dans les disciplines telles que le BigData, l'IoT et les architectures à base de services. Cette formation permet d'en comprendre les principes, à travers la réalisation d'applications concrètes.

Objectifs

- Comprendre le concept de programmation "flow-based"
- Pouvoir installer et configurer Apache Nifi
- Etre en mesure de réaliser des flux impliquant différents composants logiciels

Participants

Architecte, tech lead, développeur souhaitant utiliser Apache Nifi.

Pré-requis

- Notions en intégration d'applications et architecture logicielle de SI
- Pratique du développement Java

Programme

Apache Nifi et la programmation "flow-based"

- Les concepts de programmation "flow-based" et les workflows
- Le projet open-source Apache Nifi et ses concurrents (Apache Camel, Node-RED, ...)
- L'éco-système Nifi et ses distributions (Hortonworks DataFlow, ...)
- Les fonctionnalités essentielles
- Les pré-requis d'installation et les modes de fonctionnement

TP:installation du produit Apache Nifi et vérification de son fonctionnement.

Concepts de base Apache Nifi

- Les cas d'usage d'Apache Nifi: EAI, BigData, IoT, ...
- Les concepts de FlowFile, de Processor et de Connector
- La prise en main de l'interface Web
- La création d'un flux et son fonctionnement
- Le monitoring

TP:création et exécution d'un flux de manipulation de fichiers avec GetFile, PutFile, ...

Utilisation avancée

- Le format des FlowFiles
- Le langage EL et la gestion de conditions de routage
- Les principaux processeurs et leurs paramètres
- L'utilisation de scripts (Groovy, Python, ...)
- La gestion d'erreurs
- Controller Service, Processor Group et Reporting task
- L'utilisation de templates

TP:réalisation d'un flux conditionnel avec différents systèmes (brokers JMS ou MQTT, BD, ...).

Cluster et BigData

- Le fonctionnement en cluster avec Zookeeper
- L'intégration dans un environnement BigData (Hadoop HDFS, Spark)

TP:configuration d'un cluster avec Zookeeper.

Extensions

- Les principes de fonctionnement d'un Processor
- Le paramétrage et l'accès aux données d'un flux
- Le packaging d'un Processor dans un NAR

TP:intégration d'une Processor tiers.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des flux avec Apache Camel

Code
CP-CAML
Durée
2 jours
Tarif
1290 € HT
Sessions
28/03/2019 au 29/03/2019
11/06/2019 au 12/06/2019
07/11/2019 au 08/11/2019

Description

Le routage des données et le traitement par flux est au coeur de l'intégration d'applications d'entreprise (EAI). On le retrouve également dans les disciplines telles que le BigData, l'IoT et les architectures à base de services. Cette formation permet d'en comprendre les principes, à travers la réalisation d'applications concrètes.

Objectifs

- Comprendre le concept de programmation "flow-based"
- Pouvoir installer et configurer Apache Camel
- Etre en mesure de réaliser des flux impliquant différents composants logiciels

Participants

Architecte, tech lead, développeur souhaitant utiliser Apache Camel.

Pré-requis

- Notions en intégration d'applications et architecture logicielle de SI
- Pratique du développement Java
- Idéalement connaissance des mécanismes de base Spring

Programme

Apache Camel et la programmation "flow-based"

- Les concepts de programmation "flow-based" et les workflows
- Le projet open-source Apache Camel et ses concurrents (Apache Nifi, Node-RED, ...)
- L'éco-système Camel et ses distributions (JBoss, ...)
- Les fonctionnalités essentielles
- Les pré-requis d'installation et les modes de fonctionnement

TP:installation du produit Apache Camel et vérification de son fonctionnement.

Concepts de base Apache Camel

- Les cas d'usage d'Apache Camel: EAI, BigData, IoT, ...
- Les concepts de base: route, component, data format, endpoint, ...
- Les principaux composants et leurs paramètres
- La création de routes par DSL ou XML
- Le monitoring

TP:création et exécution d'un flux de manipulation de fichiers.

Utilisation avancée

- Les patterns d'architecture EIP
- Le lien avec les beans Spring
- Le développement avec Maven
- La gestion de conditions de routage
- La gestion d'erreurs
- L'utilisation de templates
- La mise en oeuvre avec OSGi (Apache Karaf)

TP:réalisation d'un flux conditionnel avec différents systèmes (brokers JMS ou MQTT, BD, web services, ...).

Montée en charge

- Les différents modes de démarrage et d'arrêt
- Le fonctionnement en cluster, le failover
- La configuration master/slave
- L'intégration dans un environnement BigData (Hadoop HDFS, Spark)

TP:configuration d'un cluster.

Extensions

- Les principes de fonctionnement d'un Component spécifique
- Le paramétrage et l'accès aux données d'un flux
- La réalisation d'un nouveau composant

TP:réalisation d'un nouveau composant

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

DevOps

Descriptif de la filière

Cette filière présente des formations liées au mouvement DevOps.

Liste des formations de cette filière

Intitulé	Durée	Description
Essentiel GIT, mettre en oeuvre le contrôle de versions	1 jour	Cours pratique de présentation des notions de base du logiciel de gestion de versions GIT.
Essentiel Docker, mise-en-oeuvre	1 jour	Cours pratique de présentation des notions de "conteneurisation" et leur utilisation dans l'environnement Docker

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel GIT, mettre en oeuvre le contrôle de versions

Code
CP-GIT
Durée
1 jour
Tarif
690 € HT
Sessions
06/02/2019
27/03/2019
29/05/2019
13/09/2019
08/11/2019

Description

Git est le système de contrôle de version decentralisé dont on entend parler en ce moment. Créé par Linus Torvald, il est léger et efficace. Il permet de gérer des petits projets ainsi que de très gros projets.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Toutes personnes souhaitant découvrir GIT.

Pré-requis

- Connaissances de base d'un système d'exploitation
- Pratique des opérations d'installation et de lancement d'applications

Programme

Présentation de GIT
- Concepts de base du contrôle de version
- La gestion centralisée ou distribuée
- Principe de fonctionnement
- Les différentes solutions de gestion de versions

Installation et configuration de Git
- Installation sous différents systèmes
- Configuration de l'environnement de travail
- Déclaration d'outils graphiques de comparaison/fusion
- Présentation d'outils graphiques

Utilisation de Git, les fondamentaux
- Le modèle objet Git
- Le répertoire de travail et le répertoire Git
- La zone d'index ou staging area
- Les concepts de branche, tag et dépôt
- Création et initialisation d'un dépôt

Gestion locale des fichiers
- Consultation de l'état du répertoire de travail
- Ajout, ignorance, modification, suppression et recherche de fichiers
- Annulation et visualisation des modifications
- Parcours de l'historique des révisions

Gestion des branches
- La branche master
- Création de branches
- Changement de branche
- Fusion de branches
- Gestion des conflits

Partage de travail et collaboration
- Mise en place d'un dépôt distant
- Les branches distantes
- Récupération des modifications
- Publier ses modifications

Mise en oeuvre des outils Git
- Git-Gui/Gitk et SourceTree : clients graphiques Git pour Windows
- TortoiseGit : l'extension Git pour l'explorateur Windows
- GitWeb : l'interface Web de navigation au sein de dépôts Git
- GitHub : service Web d'hébergement de dépôts Git
- GitLab et Gogs : alternatives auto-hébergées à GitHub
- Gerrit : application de revue de code

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel Docker, mise-en-oeuvre

Code
CP-DOCK
Durée
1 jour
Tarif
690 € HT
Sessions
07/02/2019
28/03/2019
30/05/2019
12/09/2019
07/11/2019

Description

Docker est le leader mondial des plates-formes de "containerisation". Les conteneurs Docker permettent de contenir un morceau de logiciel dans un système de fichiers complet qui contient tout le nécessaire pour exécuter son code: outils système, les bibliothèques système etc... Tout ce qui peut être installé sur un serveur. Cela garantit que le logiciel fonctionnera toujours, quel que soit son environnement .

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Toute personne souhaitant découvrir Docker.

Pré-requis

- Notions en Linux (commandes shell de base) et virtualisation

Programme

Présentation de Docker
- Concepts de base de la "containerisation"
- Les conteneurs et les machines virtuelles
- La notion d'images
- Installation sous différents systèmes

Utilisation de Docker, les fondamentaux
- Récupération d'une image Docker
- Lancement d'un conteneur à partir d'une image
- Gestion du filesystem
- Gestion du réseau
- Utilisation du Dockerfile
- Architecture multi-containers

TP:configuration et lancement d'un conteneur à partir d'une image existante.

Cycle de vie logiciel avec Docker
- Développement
- Packaging et déploiement
- Partage de travail et collaboration

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

BigData

Descriptif de la filière

Cette filière présente des formations liées au Big Data.

Liste des formations de cette filière

Intitulé	Durée	Description
Réaliser des applications Apache Spark pour le Big Data	3 jours	Formation pratique de développement d'applications Java ou Scala pour traiter en temps réel des données Big Data.
Maîtriser les bases du langage Python.	3 jours	Stage pratique de découverte des bases de la programmation Python.
Développer avec Scala.	3 jours	Stage pratique de découverte des bases de la programmation Scala.

Formations rattachées

Intitulé	Durée	Description
Développer des flux avec Apache Nifi	2 jours	Cours pratique de prise en main du routeur logiciel et moteur de workflow Apache Nifi
Développer des flux avec Apache Camel	2 jours	Cours pratique de prise en main du routeur logiciel et moteur de workflow Apache Camel

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Réaliser des applications Apache Spark pour le Big Data

Code
CP-SPK
Durée
3 jours
Tarif
1590 € HT
Sessions
15/05/2019 au 17/05/2019
14/10/2019 au 16/10/2019

Description

Vous développerez des applications Java ou Scala pour traiter en temps réel des données issues du Big Data. Vous collecterez, stockerez et traiterez avec Spark des données de formats hétérogènes afin de mettre en place des chaînes de traitement intégrées à votre système d'information. Les travaux pratiques sont réalisés en Java ou Scala au choix du participant.

Objectifs

- Maîtriser les concepts fondamentaux de Spark
- Développer des applications avec Spark Streaming
- Faire de la programmation parallèle avec Spark sur un cluster
- Exploiter des données avec Spark SQL
- Avoir une première approche du Machine Learning

Participants

Ce cours s'adresse à des développeurs Java ou Scala souhaitant découvrir et mettre en place Apache Spark.

Pré-requis

- La pratique de la programmation Java ou Scala est nécessaire.

Programme

Présentation d'Apache Spark

Présentation des objectifs et concepts Apache Spark.
- Historique du Framework.
- Les différentes versions de Spark (Scala, Python et Java).
- Comparaison avec Apache Hadoop.
- Les différents modules de Spark.

TP: installation et configuration de Spark, exécution d'un premier exemple avec le comptage de mots.

Les Resilient Distributed Dataset (RDD)

Programmer avec les Resilient Distributed Dataset (RDD).
- Créer, manipuler et réutiliser des RDD.
- Accumulateurs et variables broadcastées.
- Utiliser des partitions.

TP: manipulation de différents Datasets à l'aide de RDD et utilisation de l'API fournie par Spark.

Spark SQL

Manipuler des données structurées avec Spark SQL.
- SQL, DataFrames et Datasets.
- Les différents types de sources de données.
- Interopérabilité avec les RDD.
- Performance de Spark SQL.
- JDBC/ODBC server et Spark SQL CLI.

TP: manipulation de Datasets via des requêtes SQL, connexion avec une base externe via JDBC.

Spark en cluster

Exploiter Apache Spark en cluster.
- Les différents types d'architecture : Standalone, Apache Mesos ou Hadoop YARN.
- Configurer un cluster en mode Standalone.
- Packager une application avec ses dépendances.
- Déployer des applications avec Spark-submit.
- Dimensionner un cluster.

TP: configuration d'un cluster Spark.

Spark Streaming pour le temps-réel

Analyser en temps réel avec Spark Streaming.
- Présentation des Discretized Streams (DStreams).
- Les différents types de sources.
- Manipulation de l'API.
- Comparaison avec Apache Storm.

TP: consommation de logs avec Spark Streaming.

GraphX

Manipuler des graphes avec GraphX.
- Les différentes opérations.
- Créer des graphes.
- Vertex and Edge RDD.
- Présentation de différents algorithmes.

TP: manipulation de l'API GraphX à travers différents exemples.

Machine Learning

Introduction au Machine Learning.
- Introduction au Machine Learning.
- Les différentes classes d'algorithmes.
- Présentation de SparkML et MLlib.
- Implémentations des différents algorithmes dans MLlib.

TP: utilisation de SparkML et MLlib.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Maîtriser les bases du langage Python.

Code
CP-PYTH
Durée
3 jours
Tarif
1490 € HT
Sessions
20/03/2019 au 22/03/2019
14/10/2019 au 16/10/2019

Description

L'objectif de cette session est de découvrir et exploiter les bases du langage Python. Ce langage connaît un succes grandissant dans de nombreuses applications (BigData, Data Science, IoT, ...).

Objectifs

- Comprendre le fonctionnement d'une application Python
- Maîtriser la syntaxe du langage
- Mettre en oeuvre les concepts de programmation Objet
- Connaître les modules de base de Python
- Réaliser une application simple sans interface graphique

Participants

Développeur, architecte ou responsable technique.

Pré-requis

- La maîtrise des bases de programmation est indispensable pour profiter des travaux pratiques.
- La connaissance des concepts Objet est également nécessaire.

Programme

Introduction

Présentation de l'éco-système Python.
- Les origines du langage et ses acteurs principaux.
- Les avantages et inconvénients de Python.
- Les cas d'usage de Python.
- Les outils disponibles (éditeurs, IDE, modules, ...).
- Le fonctionnement REPL.

TP: configurer un environnement de développement, faire fonctionner un premier programme.

Bases du langage

Présentation des concepts de base du langage Python.
- L'organisation du code et sa structuration.
- Les conventions de nommage.
- Les types de données de base.
- Les déclarations de variables et leurs portées.
- Les instructions (test, itération, ...).
- Les déclarations et appels de fonctions.

TP: réaliser un programme sans construction orientée objet.

Programmation Objet

Mise en oeuvre des concepts Objet en Python.
- Le modèle Objet de Python.
- Définir une classe, des attributs et des opérations.
- L'encapsulation.
- L'instanciation et l'envoi de messages.
- Les associations, agrégations, compositions et l'héritage.
- La mise en oeuvre du polymorphisme.

TP: réaliser un programme avec classes et polymorphisme.

Librairie standard StdLib et modules

La notion de module et la bibliothèque de modules standard de Python.
- La notion de module.
- Le dépôt central Pypi.
- Le contenu de StdLib.
- La découverte de quelques modules (re, sys, os, ...).

TP: utiliser des modules de StdLib, développer un nouveau module.

Outillage

Les principaux outils Python.
- Le plugin Eclipse Pydev.
- La mise au point avec Pychecker.
- Le respect des standards avec Pylint.
- La génération de documentation.

TP: utiliser des outils Python.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer avec Scala.

Code
CP-SCA
Durée
3 jours
Tarif
1590 € HT
Sessions
20/03/2019 au 22/03/2019
14/10/2019 au 16/10/2019

Description

L'objectif de cette session est de découvrir et exploiter les bases du langage Scala. Ce langage connaît un succes grandissant dans le domaine de la réalisation d'applications orientées Big Data. Sa compatibilité avec Java lui offre une très grande ouverture.

Objectifs

- Comprendre le fonctionnement d'une application Scala
- Maîtriser la syntaxe du langage
- Mettre en oeuvre les concepts de programmation Objet
- Utiliser des packages Java existants
- Réaliser une application simple sans interface graphique

Participants

Développeur, architecte ou responsable technique.

Pré-requis

- La maîtrise des bases de programmation est indispensable pour profiter des travaux pratiques.
- La connaissance des concepts Java est un plus.

Programme

Introduction

Présentation de l'éco-système Scala.
- Les origines du langage et ses acteurs principaux.
- Les orientations objet et fonctionnelle du langage.
- Les cas d'usage de Scala.
- Les outils disponibles (éditeurs, IDE, ...).

TP: configurer un environnement de développement, faire fonctionner un premier programme.

Bases du langage

Présentation des concepts de base du langage Scala.
- L'organisation du code et sa structuration.
- Les types de données de base.
- Les déclarations de variables et leurs portées.
- Les instructions (test, itération, ...).
- Les scripts Scala.
- Les déclarations et appels de fonctions.

TP: réaliser un programme.

Programmation Objet

Mise en oeuvre des concepts Objet avec Scala.
- Définir une classe, des attributs et des opérations.
- L'encapsulation.
- L'instanciation et l'envoi de messages.
- Les associations, agrégations, compositions et l'héritage.
- La mise en oeuvre du polymorphisme.
- Les closures.

TP: réaliser un programme avec classes et polymorphisme, utiliser les closures.

Librairies

Découverte de quelques librairies spécifiques.
- La notion de librairie en Scala.
- Le framework Play.
- Le framework akka.

TP: utiliser des frameworks Scala.

Scala avancé

L'utilisation des concepts avancés de Scala.
- L'intégration de codes Java et Scala.
- La programmation parallèle.
- Le pattern matching.

TP: utiliser des librairies Java en Scala.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

C++

Descriptif de la filière

La filière C++ propose des formations permettant de s'initier au langage C++ des bases du langage à l'utilisation de librairies permettant d'améliorer la qualité du code. Les débutants souhaitant s'initier au langage y trouveront une formation de base et les personnes expérimentées seront surement interessées par une formation plus poussée sur l'utilisation de STL ou Boost, par exemple.

Liste des formations de cette filière

Intitulé	Durée	Description
Développer en C	2 jours	Présentation des notions de base du langage C.
Développer en C++ - Niveau 1	3 jours	Présentation pratique des notions de base du langage C++ et des concepts Objet.
Développer en C++ 11/14/17	3 jours	Formation pratique de mise-en-oeuvre des innovations de C++11/14/17.
Développer en C++ - Niveau 2	3 jours	Formation pratique permettant d'améliorer ses applications en C++.
Développer avec Qt	3 jours	Présentation des notions de base de la librairie Qt.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer en C

Code
CP-C
Durée
2 jours
Tarif
1190 € HT
Sessions
sur demande

Description

Le langage C bien qu'étant un ancien langage est encore le plus utilisé à ce jour dans le domaine de l'informatique. Il permet aussi bien d'écrire des applications bas niveau (écriture de modules pour le noyau linux) que de mettre en place des IHM évoluées (en utilisant une librairie comme GTK. Sa maîtrise est indispensable aux développeurs souhaitant utiliser un langage tel que C++ ou Objective-C.

Participants

Toutes personnes souhaitant découvrir le langage C.

Pré-requis

- La connaissance d'un langage informatique est nécessaire.

Programme

Les bases du langage

Présentation des concepts essentiels du langage C.

- Les charactéristiques du langage
- Structure d'un programme C
- Les fichiers source
- Les commentaires
- Les jeux de caractères
- La déclaration de variables
- C et la compilation de programmes

Les types

Le langage C offre différents types que l'on peut utiliser dans les applications.

- Les entiers et les flottants
- Les types complexe (C99)
- Les types énumérés
- Le type void
- Les structures de bloc
- Les boucles

Les fonctions

Présentation des fonctions et de leurs utilisations possibles.

- Structure et déclaration d'une fonction
- Utilisation de pointeurs en argument ou en paramètre
- Les fonctions inline
- Les fonctions à nombre d'arguments variable

Utilisation de tableaux

Présentation des tableaux et de leur utilisation.

- Définition d'un tableau
- Initialisation d'un tableau
- Tableaux et chaînes de caractères
- Tableau à plusieurs dimensions

Les pointeurs

En C il est parfois nécessaire d'allouer soi-même l'espace mémoire (allocation dynamique). Pour cela on va utiliser les pointeurs.

- Déclaration des pointeurs
- Opération sur les pointeurs
- Pointeurs et tableaux
- Les pointeurs de fonction
- Gestion dynamique de la mémoire

Les entrées/sorties

Une application a besoin de pouvoir communiquer avec le monde extérieur. Le langage C met à disposition des méthodes pour effectuer cette communication.

- Les types de flux
- Utilisation de fichiers

Les directives du pré-processeur

Les directives du pré-processeur permettent d'enrichir le code en définissant des conditions d'exécution de certaines portions de code.

- Définition et utilisation de macros
- Compilation conditionnelle

La librairie standard

Le langage C offre un grand nombre de fonctions de base permettant de gérer aussi bien les dates que l'internationalisation.

- La gestion avancée de la mémoire
- Le débogage
- Recherche et tri

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer en C++ - Niveau 1

Code
CP-CPP1
Durée
3 jours
Tarif
1490 € HT
Sessions
12/06/2019 au 14/06/2019
23/09/2019 au 25/09/2019

Description

Le langage C++ bien qu'ayant une réputation de complexité est encore beaucoup utilisé aujourd'hui. Il est important pour l'utiliser de comprendre les concepts associés ainsi que les problématiques mémoire qu'il est possible de rencontrer.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Toutes personnes souhaitant découvrir le langage C++.

Pré-requis

- La connaissance d'un langage informatique est préférable

Programme

Les notions de base en C++

Avant de pouvoir utiliser le langage, il est nécessaire de découvrir les notions de base qui y sont associées.

- Les types de base
- Les boucles et les conditions
- Les pointeurs
- Le modificateur const
- Les références
- Les espaces de nommage

Les concepts Objet

C++ étant un langage orienté Objet, il est important de regarder quels en sont les principes et comment modéliser cela.

- Les concepts Objet
- Le langage UML
- Les diagrammes de classe

Les fonctions

Présentation des fonctions et de leurs utilisations possibles.

- Structure et déclaration d'une fonction
- Les paramètres par défaut
- La surcharge de fonctions
- La redéfinition de fonction
- Les fonctions constantes

Les classes en C++

Présentation de l'utilisation des classes, structures et unions en C++.

- La structure d'une classe
- Les structures et les unions
- Les interfaces
- Les constructeurs et le destructeur

L'héritage

Présentation de la mise en place des principes de base de l'héritage.

- Mise en place de l'héritage
- La redéfinition des fonctions
- Les fonctions virtuelles
- Les classes abstraites

La surcharge des opérateurs

Présentation du mécanisme de surcharge des opérateurs.

- Pourquoi surcharger les opérateurs ?
- Les différents types de surcharges
- Surcharges de quelques opérateurs +, - ...

Les templates

Mise en place de la généricité pour les classes et les fonctions.

- Le rôle des templates
- Les fonctions templates
- Les classes templates

La gestion des exceptions

Présentation de la gestion des erreurs en C++.

- Les blocs try/catch
- Définir des exceptions
- Lancer une exception

Les entrées/sorties

Etude des classes permettant de gérer les entrées/sorties standard ainsi que les accès aux fichiers.

- Les différentes classes
- Lecture des entrées clavier
- Affichage dans la console
- Accès aux fichiers

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer en C++ 11/14/17

Code
CP-CPP1X
Durée
3 jours
Tarif
1590 € HT
Sessions
20/03/2019 au 22/03/2019
14/10/2019 au 16/10/2019

Description

Cette formation aborde de façon concrète les avancées des versions 11/14/17 de C++: lambdas, templates variadiques, threads, mécanisme de déplacement, ... Ces nouveautés permettent à C++ d'offrir des mécanismes analogues à ceux de ces concurrents tels que Java ou C#, et améliorent de façon très importante les codes produits.

Objectifs

- Utiliser les améliorations syntaxiques introduites en C++11
- Ecrire des classes C++ de meilleure qualité basées sur les déclarations explicites
- Comprendre les nouveautés en matière de gestion des exceptions
- Maîtriser les concepts de déplacement et de pointeur intelligent
- Exploiter l'API normalisée de programmation multi-threadée
- Utiliser des templates variadiques
- Développer des lambda-expressions

Participants

Développeur, concepteur ou chef de projet MOE connaissant C++.

Pré-requis

- La pratique de C++ est indispensable pour suivre cette formation.

Programme

Les améliorations du langage

Ce module présente l'ensemble des améliorations destinées à rendre le code C++ plus lisible et maintenable.

- Les avancées des versions C++ 11, 14, 17 et 20
- Les améliorations des types: enum, unions, POD
- Le pointeur nullptr
- La boucle for sur intervalle
- L'inférence de type à la compilation avec auto

TP: première mise en pratique avec auto et la nouvelle boucle for

Les nouveautés des classes C++

Ce module présente l'ensemble des améliorations destinées à rendre les classes C++ plus lisibles.

- L'initialisation des données membres
- La délégation des constructeurs
- La déclaration de redéfinition (override) et final
- Les mécanismes de conversions de types explicites

TP: réalisation d'une arborescence de classes

Les exceptions

Ce mécanisme de gestion d'erreur subit de grosses évolutions.

- Les préconisations d'emploi des exceptions
- Le pointeur sur exception courante
- Les clauses throw et noexcept

TP: réalisation d'un code employant la clause noexcept

Le mécanisme de déplacement

Présentation du mécanisme permettant d'optimiser la gestion mémoire et les performances.

- Introduction du concept de rvalue
- Le constructeur par déplacement et l'affectation par déplacement
- La forme normale d'une classe
- L'impact du principe de "move" au niveau de la STL
- Le bon usage du mécanisme de placement (emplace_back)

TP: réalisation d'une classe dont les instances sont "move constructible" et "move assignable"

Les templates

Présentation des améliorations.

- Les templates variadiques
- La spécialisation partielle de templates
- Les arguments templates locaux et anonymes

TP: utilisation de templates variadiques

Les lambda-expressions

La programmation fonctionnelle en C++.

- Le concept de programmation fonctionnelle
- La déclaration et l'appel d'une lambda-expression
- La capture du contexte englobant
- Les lambda-expression templates

TP: programmation de différentes lambda-expressions avec paramètres et capture

Les pointeurs intelligents

Les mécanismes de gestion automatique des allocations/libérations.

- Le type shared_ptr et son fonctionnement (zone de contrôle)
- Les références faibles (weak_ptr)
- L'utilisation des smart pointers dans la STL

TP: utilisation de shared_ptr

Les threads normalisés

La programmation multi-threadée en C++.

- Le fonctionnement et les états d'un thread
- Les techniques de lancement d'un thread (callable, async, future, promise, ...)
- Les sections critiques (mutex) et la synchronisation des threads (condition_variable)
- La gestion mémoire (volatile et types atomiques)

TP: parallélisation d'un algorithme et analyse des performances, réalisation d'un producteur-consommateur (moniteur de Hoare).

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer en C++ - Niveau 2

Code
CP-CPP2
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

Le langage C++ est complexe, et offre de nombreuses possibilités. Que ce soit la gestion de la mémoire ou une meilleure utilisation du langage, il est important de s'assurer que le code mis en place offrira le plus de sécurité.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours est destiné à des personnes utilisant déjà le langage C++ et qui souhaitent approfondir leurs connaissances.

Pré-requis

- Une première expérience du langage C++

Programme

Introduction à la STL
Présentation des éléments de base de la STL.
- La classe string
- Les itérateurs
- Les conteneurs associatifs
- Les conteneurs séquentiels

Le typage dynamique
Présentation du typage dynamique avec RTTI.
- Avantages et inconvénients du typage dynamique
- La conversion de type dynamique
- L'opérateur d'identification de type

Les smart pointers
Présentation des smart pointers et de leur utilisation.
- Qu'est-ce qu'un smart pointer ?
- Avantages et inconvénients
- Conversion implicite

Compatibilité entre C et C++
Présentation des éléments de code qui ne sont pas compatibles entre C et C++.
- Le code C incompatible avec le C++

La partie utilities de la STL
Dans STL, utilities offre de nombreux outils qui permettent de simplifier la mise au point d'applications C ++.
- La classe pair
- Les pointeurs automatiques
- Les fonctions auxiliaires

Les fonctions Objet de la STL
Présentation des fonctions Objet (souvents appelées functors) de la STL.
- Présentation des fonctions Objet
- Utilisation des fonctions Objet

Les composants numériques de la STL
Présentation de l'utilisation des composants numériques (nombres complexes ...).
- Les nombres complexes
- La classe valarray
- Les fonctions numériques

Les allocators
Présentation des allocators et de leur utilité.
- Rôle des allocators
- L'allocator par défaut
- Les allocators utilisateurs

La librairie Boost
Présentation de la librairie Boost.
- Présentation de la meta-programmation
- Calcul et optimisation de code
- Génération automatique de code

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer avec Qt

Code
CP-QT1
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

Le langage C++ offre de nombreuses possibilités pour mettre en place des applications. Cependant il ne permet de mettre au point facilement des IHM, d'accèder aux bases de données ... Dans ce cas, on va devoir passer des librairies ou frameworks externes. Un des frameworks C++ les plus utilisé actuellement est Qt qui est mis au point par Nokia.

Participants

Toutes personnes souhaitant découvrir Qt.

Pré-requis

- La connaissance du langage C++ est nécessaire

Programme

Présentation de Qt

Présentation des origines du langage, des différentes façon d'utiliser Qt. Présentation des autres librairies graphiques.

- Historique du langage
- Les différentes possibilités d'utilisation
- Les autres librairies graphiques existantes

Les différents types de projets

Présentation des différents types de projets et de leurs utilisations, en passant par le logiciel Qt Creator.

- Présentation des différents types de projet avec QT
- Présentation de Qt Creator
- La structure de base d'une application a base d'IHM

Le modèle MVC avec Qt

Rappel sur le modèle MVC et s mise en place au sein de Qt.

- Présentation du modèle MVC
- Le modèle MVC dans Qt

Les différents composants graphiques

Présentation des composants graphiques de Qt et de leur utilisation.

- Les composants de base de l'IHM (QMainWindow, QFrame, QLabel …)
- La gestion du positionnement des composants
- Les boites de dialogue (QDialog)
- Les menus (QMenu)
- Modèles prédéfinis et personnalisés
- outils de conception visuelle de Qt (Qt Designer …)

La gestion des événements

Présentation de la mise en place des principes de base de l'héritage.

- Notions de signal et slot
- Déclaration de signaux et de slots
- Installer des filtres d'événement
- Accéder à l'application pendant un traitement lourd (timer et hasPendingEvents())

Les éléments importants d'un projet

Présentation des éléments constitutifs d'un projet.

- Le fichier 'pro'
- Les fichiers de conception graphique (ui)
- Les fichiers de gestion d'internationalisation (ts et qm)
- Les types de bases du langage (qint, qfloat …)
- La compilation avec qmake
- La classe QObject

Utilisation de XML avec Qt

Rappels sur le langage XML et son utilisation avec Qt.

- Un rappel sur XML
- Les parsers DOM et SAX
- Parsing de documents XML en utilisant Qt

Le système de plugin de Qt

Présentation du système de plugin.

- Comprendre ce que sont les plugins avec Qt
- différentes classes de plugin (QStylePlugin …)
- Les éléments nécessaire à la mise en place d'un plugin pour Qt
- Mise au point d'application gérant des plug-ins

L'internationalisation

Mise en place de l'internationalisation des applications.

- Rappel sur Unicode
- L'objet QTranslator
- Mise en place de l'internationalisation dans l'application
- Qt Linguist

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Framework Spring

Descriptif de la filière

Cette filière regroupe les formations Spring.

Liste des formations de cette filière

Intitulé	Durée	Description
Développer des applications Java avec Spring	3 jours	Cette formation pratique permet de réaliser des applications Java utilisant Spring.
Automatiser le déploiement avec Spring Boot	3 jours	Cette formation pratique permet de mettre en place des applications utilisant Spring Boot.
Développer des micro-services avec Spring et Docker	3 jours	Cette formation permet de mettre en place des micro-services avec Spring et Docker.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des applications Java avec Spring

Code
CP-SPRING
Durée
3 jours
Tarif
1490 € HT
Sessions
27/03/2019 au 29/03/2019
28/10/2019 au 30/10/2019

Description

Spring est un framework open source qui a été mis au point pour les applications 3-tiers. Son objectif est de faciliter le développement et le test des applications en enlevant les lourdeurs des applications Java EE. Cette formation permettra de comprendre les principes d'injection de dépendance (IoD) et d'inversion de contrôle (IoC) sur lesquels s'appuie Spring pour faciliter la création d'objets et simplifier la mise en place d'applications.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse à des développeurs Java souhaitant découvrir et mettre en place des applications utilisant Spring.

Pré-requis

- La connaissance préalable du langage Java est nécessaire.

Programme

Présentation de Spring
Présentation générale de Spring et des différentes parties qui le composent.
- Le framework Spring
- Les modules de Spring
- Injection de dépendance et inversion de contrôle
- Exemples d'utilisation

Le conteneur IoC et les beans
Présentation du conteneur d'IoC et des beans associés.
- Le conteneur d'IoC
- Les beans
- La classe BeanFactory
- Utilisation d'annotations

TP:déclaration et utilisation de beans Spring.

Accès aux ressources
Présentation des méthodes d'accès aux ressources locales et distantes de Spring.
- L'interface Resource
- Les implémentations existantes
- Le ResourceLoader

TP:gestionnaire de ressources.

Validation et conversion
Présentation des composants avancés basés sur le modèle MVC.
- Validation avec Spring
- Conversion de type
- Formatage de données

TP:réalisation de mécanismes de validation.

Spring Expression Language (SpEL)
Présentation du langage SpEL.
- Présentation du langage
- L'interface Expression

TP:écriture d'expressions EL.

L'accès aux données
Les possibilités offertes par Spring pour accéder aux données.
- La gestion des transactions
- Support de DAO
- Accès aux données avec JDBC
- Mapping Objet / Relationnel avec Spring
- Marshalling XML en utilisant O/X Mappers

TP:réalisation d'un module DAO.

Spring MVC
Présentation de Spring MVC.
- Les notions de base de Spring MVC
- Le modèle MVC
- La servlet DispatcherServlet

TP:bases d'un projet Spring MVC.

La partie vue de Spring MVC
Présentation de la partie Vue de Spring MVC.
- Les tags de SpringMVC
- Les tiles

TP:définition d'une vue.

Autres aspects
Présentation des autres parties de Spring MVC.
- Spring AOP
- Intégration des EJB
- Intégration de JMS

TP:intégration d'un module fourni.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Automatiser le déploiement avec Spring Boot

Code
CP-SPB
Durée
3 jours
Tarif
1590 € HT
Sessions
17/06/2019 au 19/06/2019
16/09/2019 au 18/09/2019
02/12/2019 au 04/12/2019

Description

Spring Boot permet de simplifier le démarrage, le développement, la configuration et le déploiement d'un projet Spring. Cette formation permet d'en appréhender et expérimenter les mécanismes essentiels. Les travaux pratiques sont réalisés avec l'outil STS.

Objectifs

- Comprendre les concepts et apports de Spring Boot
- Créer et configurer un projet Spring Boot
- Adapter la configuration aux spécificités des applications web et d'accès aux données
- Packager et déployer une application Spring Boot

Participants

Ce cours s'adresse à des développeurs Spring souhaitant découvrir et mettre en place Spring Boot.

Pré-requis

- La connaissance préalable des bases Spring est nécessaire.
- Une expérience de réalisation d'applications avec Spring Web/REST et/ou Spring Data est conseillée.

Programme

Présentation de Spring Boot

Présentation des objectifs et concepts de Spring Boot.
- Rappels sur le framework Spring
- La gestion des dépendances
- Les différents types d'applications et leur déploiement
- Les pré-requis et le support dans STS, installation
- Alternative avec Dropwizard

TP:lancement d'une application Spring Boot fournie.

Un projet Spring Boot

Les composants d'un projet Spring Boot.
- Le support des différents types d'applications
- La structure et la configuration d'un projet Spring Boot, l'auto-configuration
- L'exécution et l'accès aux logs
- Utilisation avec Maven ou Gradle

TP:création, configuration et exécution d'une application Web.

Spring Boot et les applications Web

Utiliser Spring Boot avec une application Web.
- Les conteneurs Web utilisables avec Spring Boot
- Les types d'applications compatibles
- Utilisation avec Spring MVC et Spring REST
- Impacts sur la sécurité avec Spring Security

TP:utilisation avec une application REST.

L'accès aux données avec Spring Boot

Utiliser Spring Boot avec une application d'accès aux données.
- Les concepts de Spring Data
- Spring Data JPA et Spring Data REST avec Spring Boot
- Spring Boot et les bases NOSQL

TP:utilisation avec une application de type JPA.

Déployer une application Spring Boot

Les spécificités du déploiement d'un projet Spring Boot.
- Le packaging et les starters
- Le passage en production
- Application auto-executable
- Les Profiles
- Spring Boot Actuator

TP:utiliser Spring Boot Actuator.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des micro-services avec Spring et Docker

Code
CP-MSVC
Durée
3 jours
Tarif
1590 € HT
Sessions
13/03/2019 au 15/03/2019
17/06/2019 au 19/06/2019
16/09/2019 au 18/09/2019
02/12/2019 au 04/12/2019

Description

Les architectures microservices présentent de nombreux avantages mais nécessitent une grande maturité dans les pratiques de développement, de déploiement et de maintien en conditions opérationnelles. Vous mettrez en oeuvre dans cette formation les technologies essentielles pour bâtir une telle architecture.

Objectifs

- Comprendre la démarche et l'organisation sous-jacentes à la notion de micro-service
- Pouvoir choisir un modèle d'architecture adapté à ses besoins
- Développer et tester une API REST de type Swagger/OpenAPI
- Exécuter un micro-service avec Spring Boot
- Déployer des services au sein de conteneurs Docker
- Appréhender les tâches de supervision d'une architecture basée sur les micro-services

Participants

Ce cours s'adresse à des développeurs Spring souhaitant développer des architectures basées sur les micro-services.

Pré-requis

- La connaissance préalable des bases Spring est nécessaire.
- Une pratique des commandes de base Linux est conseillée.

Programme

L'approche micro-services
Présentation de la notion de micro-service.
- Les méthodes et organisations des équipes de développement.
- Approche monolithique, avantages et inconvénients. Alternatives.
- Le lien avec le DevOps et l'Agilité.
- L'intégration et le déploiement continus.
- Les "pizza teams"

L'architecture micro-services
Caractéristiques d'une architecture micro-services.
- Caractéristiques principales des micro-services.
- Les modèles architecturaux sous-jacents.
- Les interfaces et collaborations entre services. Quid de l'orchestration ?
- La problématique de l'accès aux données.
- Les acteurs et utilisateurs principaux.

TP:comparaison de différentes propositions de modèles architecturaux.

Les échanges entre micro-services REST
Présentation des différents modèles d'échanges possibles.
- Principes de base et rappels HTTP/REST.
- Choix du style de collaboration : REST request/reply vs Publish-Subscribe Messaging.
- Développer des services REST avec Java et Spring.
- Le principe HATEOAS.
- Introduction à Swagger/OpenAPI.

TP: définir une API REST avec Swagger/OpenAPI et Spring, tester le service avec SoapUI.

Exécuter un micro-service avec Spring Boot.
Bases Spring Boot permettant de simplifier l'exécution de services.
- Principes de base Spring Boot.
- Caractéristiques d'un projet Spring Boot.
- Les spécificités des applications Web/Services.
- Le packaging de l'application.

TP: lancer les micro-services précédemment développés avec Spring Boot.

Micro-services et conteneurs Docker.
Utilisation de Docker pour déployer des micro-services.
- Le fonctionnement des conteneurs Docker.
- Configurer et démarrer des conteneurs.
- Réaliser une image Docker personnalisée.
- Utiliser plusieurs conteneurs avec Docker Compose.

TP: créer un conteneur personnalisé et mettre en place une architecture multi-conteneurs.

Infrastructure, administration et surveillance.
Les solutions d'infrastructure et de supervision.
- Solutions d'infrastructure pour les microservices (Kubernetes, Mesos, Swarm).
- Solutions Cloud (Cloud Foundry, Heroku).
- Centralisation des logs.
- Définition de métriques.
- Surveillance du bon fonctionnement (Health Check).

TP: mise en place de métriques et visualisation.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Internet des Objets

Descriptif de la filière

Cette filière rassemble les formations liées au domaine de l'Internet des Objets.

Liste des formations de cette filière

Intitulé	Durée	Description
Essentiel IoT, panorama Internet des Objets.	1 jour	Séminaire d'une journée permettant de découvrir le domaine de l'IoT à travers une présentation de ses différents aspects.
Essentiel LoRaWAN, émettre et traiter des messages IoT.	1 jour	L'objectif de cette journée pratique est de découvrir la norme LoRaWAN et de réaliser la mise en place des différents éléments.
Développer des applications IoT/LoraWAN en Java et C++.	5 jours	Stage pratique de mise en place d'un chaîne IoT/LoraWAN complète capteur - gateway - serveur en Java/C++.
Développer des applications IoT/LoraWAN en Python.	5 jours	Stage pratique de mise en place d'un chaîne IoT/LoraWAN complète capteur - gateway - serveur en Python.

Formations rattachées

Intitulé	Durée	Description
Développer en C++ 11/14/17	3 jours	Formation pratique de mise-en-oeuvre des innovations de C++11/14/17.
Maîtriser les bases du langage Python.	3 jours	Stage pratique de découverte des bases de la programmation Python.
Développer des flux avec Apache Nifi	2 jours	Cours pratique de prise en main du routeur logiciel et moteur de workflow Apache Nifi

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel IoT, panorama Internet des Objets.

Code
S-IOT
Durée
1 jour
Tarif
690 € HT
Sessions
24/06/2019
09/09/2019
04/11/2019

Description

Journée-séminaire destinée à toute personne désireure de mieux comprendre le domaine de l'Internet des Objets. Les présentations sont étayées par des démonstrations d'équipements et d'objets IoT.

Objectifs

- Comprendre ce qu'on entend par Internet des Objets
- Identifier les rôles des composants d'une architecture IoT
- Connaître les principales technologies de réseau IoT

Participants

Développeur, architecte, responsable technique ou membre d'une DSI ou d'un service d'assistance MOA.

Pré-requis

- Connaissance des concepts informatiques de base souhaitable.

Programme

Internet des Objets.
Définitions de ce qu'est l'Internet des Objets.
- Les cas d'usage: smart city, santé, marketing et communication, GTC, supervision, ...
- La miniaturisation et la gestion de l'énergie.
- Les nouveaux réseaux.
- Les liens avec le cloud, le big data, l'intelligence artificielle, ...
- Les acteurs et solutions: Android Things, Win10 IOT, Amazon, ...

Les constituants de base.
Présentation des différents constituants d'une solution IoT, dans leurs grands principes.
- Les types de capteurs et leurs modalités de fonctionnement.
- Les micro-contrôleurs: ATMega, STM32, ...
- Les bus de données: UART, SPI, I2C.
- Les équipements réseau: émetteurs, concentrateurs.
- Les solutions logicielles coté serveur:MQTT, big data, ...
- La restitution/visualisation des données.

Les nouveaux réseaux pour l'IoT.
Présentation des différentes solutions disponibles.
- Les problématiques réseau liés à l'IoT
- Les solutions en présence: ZigBee, LoRaWAN, SigFox, ...
- L'arrivée de la 5G.
- Panorama d'équipements et de logiciels disponibles, écosystème IoT.
- Les modèles d'architectures sous-jacents.
- Les problèmes de sécurité.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel LoRaWAN, émettre et traiter des messages IoT.

Code
CP-LWAN
Durée
1 jour
Tarif
690 € HT
Sessions
04/02/2019
25/03/2019
27/05/2019
10/09/2019
05/11/2019

Description

Journée destinée aux informaticiens confirmés désireux de comprendre les mécanismes réseau IoT proposés par LoRaWAN. Les travaux pratiques sont réalisés sur Arduino, en liaison avec des équipements LoRaWAN spécialisés. Des modèles seront fournis afin de pouvoir se concentrer sur l'essentiel des concepts.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Développeur, architecte ou responsable technique.

Pré-requis

- La connaissance des bases de programmation est recommandée pour profiter des travaux pratiques.
- La pratique de C++ dans l'atelier Eclipse est un plus.

Programme

La spécification LoRaWAN

Présentation des concepts définis par la spécification LoRaWAN.

- Présentation de la LoRa-Alliance.
- Les modalités de communication et les classes A-B-C d'équipements.
- Les "motes", les concentrateurs et les serveurs LoRaWAN.
- La qualité de service et la sécurité.
- LoRa et LoRaWAN.
- La nature et le contenu des messages, portée et taille, débit, fréquences, ...

Serveur LoRaWAN

Le chef d'orchestre dans le modèle LoRaWAN.

- Les rôles dévolus aux serveurs LoRaWAN.
- Serveurs publics/privés et les modalités de récupération des données.
- L'activation d'un équipement auprès d'un serveur, en ABP ou OTAA.
- Panorama des serveurs disponibles: Semtech, Things Network, ...

TP:observation du bon acheminement des données d'un capteur vers un serveur.

Concentrateur LoRaWAN

L'intermédiaire entre l'émetteur (mote) et le serveur LoRaWAN.

- Le modèle de communication sous-jacent, comparaison avec d'autres modèles (GSM, ...).
- Le rôle du packet forwarder et les modalités de communication réseau.
- Les équipements et solutions logicielles associées disponibles.

TP: configuration d'un prototype de concentrateur, observation du fonctionnement d'un concentrateur spécialisé.

Mote LoRaWAN

Présentation du capteur émetteur/récepteur de messages LoRaWAN.

- L'émission et la réception de messages.
- Les équipements et solutions logicielles associées disponibles (LMIC IBM).

TP: mise en place d'un prototype de Mote sur Arduino avec un module radio LoRaWAN de type SX1276, utilisation d'une mote ABP ou OTAA.

Architecture LoRaWAN

Synthèse des modèles d'architectures offerts par LoRaWAN.

- Le choix entre opérateur public ou privé.
- La liaison avec les composants du SI: Broker MQTT, Bigdata, conteneurs ...

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des applications IoT/ LoraWAN en Java et C++.

Code
CP-IOTL
Durée
5 jours
Tarif
2390 € HT
Sessions
25/03/2019 au 29/03/2019
03/06/2019 au 07/06/2019
16/09/2019 au 20/09/2019
16/12/2019 au 20/12/2019

Description

L'objectif de cette session et de comprendre l'architecture typique d'un réseau IoT et mettre en place une chaîne IoT complète, depuis la remontée d'informations émises par des capteurs connectés LPWAN jusqu'à la DataVisualization en passant par la collecte Cloud/BigData et l'exploitation logicielle. Les travaux pratiques sont réalisés en Java et C++ avec l'IDE Eclipse, sur matériels de type Arduino et/ou Raspberry Pi.

Objectifs

- Comprendre ce qu'est l'Internet des Objets
- Maîtriser les schémas d'architecture sous-jacents
- Connaître différentes technologies réseau pour l'IoT, et en utiliser certains mécanismes de base
- Maîtriser les concepts Lora et LoraWAN, et mettre en oeuvre une mote et un gateway
- Prototyper une solution communicante LoraWAN sur base Arduino ou Raspberry
- Réaliser et utiliser des composants logiciels d'exploitation des données issues des capteurs

Participants

Développeur, architecte ou responsable technique.

Pré-requis

- La connaissance des bases de programmation est recommandée pour profiter des travaux pratiques.
- La pratique de C++ ou Java dans l'atelier Eclipse est un plus.

Programme

Introduction à l'Internet des Objets.

Présentation des concepts associés à l'IoT.

- Concepts de base IoT. Liens avec M2M. Architectures IoT.
- Problématiques de la communication et de l'énergie
- Les offres et acteurs.
- Les nouveaux réseau dédiés IoT.

Architectures IoT

Architectures matérielles et logicielles spécifiques IoT.

- Schéma général d'architecture, des capteurs au serveurs.
- Les systèmes d'exploitation adaptés à l'IoT (Linux, Raspbian, Win10 IoT...)..
- Liens avec les différents Cloud (MS Azure IoT...)..
- Sécurité et confidentialité des données.
- Les différents types de réseaux disponibles: réseaux privés, réseaux opérateurs.

Réseaux et détections de proximité.

Les moyens de communication spécifiques IoT.

- Panorama des technologies disponibles: wifi, Bluetooth, NFC, RFID...
- Echanges avec Bluetooth (BLE) et Wifi Direct.
- Fonctionnement des systèmes RFID et NFC (tag et lecteur).
- Les possibilités offertes par les Smartphones.

TP: client-serveur bluetooth, détection NFC.

Le réseau LPWAN.

Spécificités des réseaux dédiés IoT.

- Les spécificités des réseaux IoT (distance, positionnement des équipements, autonomie).
- Caractéristiques des réseaux LPWAN. Etude de SigFox et LoRaWan. Quelle programmation ?.
- Réseaux privés ou publics. Les opérateurs.
- Sécuriser un réseau IoT.

LoRa et LoRaWAN.

Présentation du standard LoRa/LoRaWAN.

- La modulation radio LoRa et la spécification LoRaWAN. L'architecture générale d'un réseau LoRaWAN.
- Les classes A-B-C d'équipements (« mote ») et les mécanismes ABP et OTAA d'inscription sur le réseau.
- Les fonctions des concentrateurs (« gateways ») et des serveurs réseaux et applicatifs.
- Sécuriser un réseau LoRaWAN.

TP: utilisation d'une mote ABP, paramétrage d'un concentrateur, configuration d'un serveur (clés).

Le prototypage.

Solutions pour prototyper un objet connecté.

- Présentation des solutions Arduino et Raspberry.
- Les possibilités Arduino Hardware (capteurs, shield, GPIO) et la programmation embarquée.
- Les environnements disponibles sur Raspberry et le lien avec Arduino.
- Le choix d'un système d'exploitation et des outils de développement.

TP: réalisation d'une mote ABP ou OTAA sur Arduino en C++, utilisation de capteurs sur bus UART ou I2C, réalisation d'un concentrateur ou d'une mote sur base Raspberry Pi en Java.

Exploitation logicielle.

Solutions pour exploiter les données des capteurs IoT.

- Les brokers de messages adaptés IoT (MQTT).
- Les APIs de type REST et outils de développement.
- Les solutions d'injection de données dans une base NoSQL BigData.
- Les différents Cloud (Google Cloud Platform, ThingPlus...).
- Les moyens logiciels permettant de sécuriser une solution IoT.
- Les techniques de Data Visualization.

TP: ajout de messages LoRaWAN dans un broker MQTT (local ou cloud), extraction et visualisation avec Apache Nifi.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des applications IoT/ LoraWAN en Python.

Code
CP-IOTP
Durée
5 jours
Tarif
2390 € HT
Sessions
03/06/2019 au 07/06/2019
16/09/2019 au 20/09/2019
16/12/2019 au 20/12/2019

Description

L'objectif de cette session et de comprendre l'architecture typique d'un réseau IoT et mettre en place une chaîne IoT complète, depuis la remontée d'informations émises par des capteurs connectés LPWAN jusqu'à la DataVisualization en passant par la collecte Cloud/BigData et l'exploitation logicielle. Les travaux pratiques sont réalisés en Python, sur matériels de type Pycom et/ou Raspberry Pi.

Objectifs

- Comprendre ce qu'est l'Internet des Objets
- Maîtriser les schémas d'architecture sous-jacents
- Connaître différents technologies réseau pour l'IoT, et en utiliser certains mécanismes de base
- Maîtriser les concepts Lora et LoraWAN, et mettre en oeuvre une mote et un gateway
- Prototyper une solution communicante LoraWAN sur base micro-contrôleur ou Raspberry
- Réaliser et utiliser des composants logiciels d'exploitation des données issues des capteurs

Participants

Développeur, architecte ou responsable technique.

Pré-requis

- La connaissance des bases de programmation est recommandée pour profiter des travaux pratiques.
- La pratique de Python est recommandée.

Programme

Introduction à l'Internet des Objets.

Présentation des concepts associés à l'IoT.

- Concepts de base IoT. Liens avec M2M. Architectures IoT.
- Problématiques de la communication et de l'énergie
- Les offres et acteurs.
- Les nouveaux réseau dédiés IoT.

Architectures IoT

Architectures matérielles et logicielles spécifiques IoT.

- Schéma général d'architecture, des capteurs au serveurs.
- Les systèmes d'exploitation adaptés à l'IoT (Linux, Raspbian, Win10 IoT...)..
- Liens avec les différents Cloud (MS Azure IoT...)..
- Sécurité et confidentialité des données, respect de la vie privée.
- Les différents types de réseaux disponibles: réseaux privés, réseaux opérateurs.

Réseaux et détections de proximité.

Les moyens de communication spécifiques IoT.

- Panorama des technologies disponibles: wifi, Bluetooth, NFC, RFID...
- Echanges avec Bluetooth (BLE) et Wifi Direct.
- Fonctionnement des systèmes RFID et NFC (tag et lecteur).
- Les possibilités offertes par les Smartphones.

TP: client-serveur bluetooth, détection NFC.

Le réseau LPWAN.

Spécificités des réseaux dédiés IoT.

- Les spécificités des réseaux IoT (distance, positionnement des équipements, autonomie).
- Caractéristiques des réseaux LPWAN. Etude de SigFox et LoRaWan. Quelle programmation ?.
- Réseaux privés ou publics. Les opérateurs.
- Sécuriser un réseau IoT.

LoRa et LoRaWAN.

Présentation du standard LoRa/LoRaWAN.

- La modulation radio LoRa et la spécification LoRaWAN. L'architecture générale d'un réseau LoRaWAN.
- Les classes A-B-C d'équipements (« mote ») et les mécanismes ABP et OTAA d'inscription sur le réseau.
- Les fonctions des concentrateurs (« gateways ») et des serveurs réseaux et applicatifs.
- Sécuriser un réseau LoRaWAN.

TP: utilisation d'une mote ABP ou OTA, paramétrage d'un concentrateur, configuration d'un serveur (clés).

Le prototypage.

Solutions pour prototyper un objet connecté.

- Présentation des solutions Arduino et Raspberry.
- Les possibilités Arduino Hardware (capteurs, shield, GPIO) et la programmation embarquée.
- Les environnements disponibles sur Raspberry et le lien avec Arduino.
- Le choix d'un système d'exploitation et des outils de développement.
- Les micro-contrôleurs sur base Python et leur intérêt.

TP: réalisation d'une mote ABP ou OTAA sur Lopy en Python, utilisation de capteurs sur bus UART, SPI ou I2C, configuration d'un concentrateur sur base Pycom Lopy en Python.

Exploitation logicielle.

Solutions pour exploiter les données des capteurs IoT.

- Les brokers de messages adaptés IoT (MQTT).
- Les APIs de type REST et outils de développement.
- Les solutions d'injection de données dans une base NoSQL BigData.
- Les différents Cloud (Google Cloud Platform, ThingPlus...).
- Les moyens logiciels permettant de sécuriser une solution IoT.
- Les techniques de Data Visualization.

TP: ajout de messages LoRaWAN dans un broker MQTT (local ou cloud), extraction et visualisation avec Apache Nifi.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

JavaEE

Descriptif de la filière

La filière JavaEE propose des formations concernant les technologies de l'édition Entreprise de Java. Il s'agit notamment du Web (Servlet, JSP, JSF, ..), des EJB ou encore des Web Services (JAX-WS, JAX-RS).

Liste des formations de cette filière

Intitulé	Durée	Description
Java Server Faces 2.X	3 jours	Développer des applications Web Java en utilisant le framework JSF (version 2).
Composants métiers JavaEE : les EJB 3.0	3 jours	Développer des composants métiers en Java en suivant la spécification EJB 3.0.
Développer des Web Services SOAP en Java	3 jours	Développement de Web Services conformes aux standards du W3C et du WS-I avec l'API JAX-WS.
Développer des services RESTful en Java	3 jours	Présentation pratique des principes REST et mise en oeuvre en Java (JAX-RS).
Exploiter les spécificités JavaEE 6	3 jours	Cette formation pratique est focalisée sur les nouveautés introduites avec JavaEE 6.
Java Persistence API	3 jours	Formation pratique de mise en oeuvre des mécanismes de persistance des données en Java avec JPA.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Java Server Faces 2.X

Code
CP-WJSF
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Java Server Faces permet de structurer et simplifier le développement d'applications web JavaEE importantes. Cette formation en présente les aspects essentiels. Le cours est basé sur la version 2.X du framework.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant apprendre à mettre en place des applications Web Java à l'aide du Framework JSF.

Pré-requis

- Maîtrise du langage Java nécessaire
- La connaissance du langage HTML
- La connaissance des JSP et des servlets

Programme

Les concepts de JSF

Avant de commencer à utiliser JSF, il est important de comprendre les concepts qui y sont associés.

- Le modèle MVC2
- Le modèle événementiel de JSF
- La servlet FacesServlet
- Le fichier facesconfig.xml
- Les ressources bundle

Les composants standard

Présentation des différents composants fournis avec JSF.

- Les différentes familles de balises
- Utilisation du langage Unified EL avec JSF

Cycle de traitement d'une requête

Analyse du fonctionnement de JSF lors de l'envoi d'une requête.

- Les différents éléments appelés

Utilisation de JavaBean dans JSF

Présentation des différents types de Beans qui peuvent être utilisés avec JSF.

- Rappel sur les JavaBeans
- Déclaration d'un Bean pour une application (Managed bean)
- Les Backing Beans

La navigation entre pages

Présentation du modèle qui permet de définir les enchainements d'appels entre les différentes pages d'une application.

- Définition des règles de navigation
- Navigation statique et dynamique

Conversion et validation de données

Présentation des possibilités offertes pour améliorer la gestion des données saisies par l'utilisateur.

- Le processus de conversion et de validation
- Utilisation de la conversion standard
- Utilisation de la validation standard
- Créer ses propres valideurs et convertisseurs

Les événements JSF

Présentation des différents événements rencontrés avec JSF et présentation de leur utilisation.

- Cycle de vie des événements
- Les types d'événements
- Les balises d'écoute d'événements

Création de composants personnalisés

Présentation de la mise au point de composants personnalisés.

- Les classes permettant de créer des composants
- Encodage et décodage

Les librairies complémentaires

Présentation d'extensions existantes pour JSF.

- Présentation de certaines librairies
- La librairie Apache Tomahawk

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Composants métiers JavaEE : les EJB 3.0

Code
CP-EJB3
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

Les EJB ont pendant longtemps souffert d'une mauvaise image : trop complexes à développer, fastidieux à configurer, pas toujours réellement portables, ... Cependant, avec la venue de JavaEE 5 et de la spécification EJB 3.0, les EJB sont devenus beaucoup plus simples à mettre en oeuvre, tout en restant très puissants.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et la plateforme JavaEE et souhaitant apprendre à développer, utiliser et intégrer des EJB.

Pré-requis

- Maîtrise du langage Java nécessaire
- Connaissances de base sur les serveurs d'application JavaEE

Programme

Les Objet Métier

Présentation de la vision de l'Objet Métier par l'OMG et ses implémentations dans diverses technologies.

- Objet métier : présentation
- Architectures à base d'objets répartis : CORBA, DCOM, RMI
- Limite des architectures à base d'objets répartis

Caractéristiques de base de JavaEE

Rappels sur les principes de base de la plateforme JavaEE et du rôle des conteneurs.

- Les serveurs d'applications JavaEE
- Rôle des conteneurs
- Architecture canonique JavaEE
- Les API de JavaEE
- Panorama des principaux serveurs JavaEE

Introduction aux EJB

Présentation des différents types d'EJB, leurs différences et leur place dans une architecture.

- EJB Session Stateless
- EJB Session Stateful
- EJB Message Driven
- La disparition des EJB Entity
- Les services offerts par le conteneur

Développer des EJB Session Stateless

Techniques et bonnes pratiques de développement des EJB Session Stateless.

- Cycle de vie d'un EJB Session Stateless
- Développer les interfaces et l'implémentation
- Présentation des annotations utiles
- Lien entre EJB Session Stateless et Web Service

Développer des EJB Session Stateful

Techniques et bonnes pratiques de développement des EJB Session Stateful.

- Cycle de vie d'un EJB Session Stateful
- Développer les interfaces et l'implémentation
- Présentation des annotations utiles

Java Persistence API

Présentation de l'API qui vient remplacer les EJB Entity.

- Le mapping Objet / Relationnel
- Notion d'entité
- Entity Manager et Persistence Unit
- Annotations JPA

Développer des EJB Message Driven

Développer des EJB évènementiels basés sur JMS.

- Java Messaging Service
- Cycle de vie d'un EJB MDB
- Les annotations utiles

Services techniques EJB

Présentation de la façon de configurer les services techniques pris en charge par le conteneur.

- Les EJB et la gestion de la sécurité
- Les EJB et la gestion des transactions

Les Intercepteurs

Présentation des Intercepteurs EJB, utilité, développement.

- Le concept d'intercepteur EJB
- Intercepter une invocation

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des Web Services SOAP en Java

Code
CP-SOAP
Durée
3 jours
Tarif
1490 € HT
Sessions
02/04/2019 au 04/04/2019
28/10/2019 au 30/10/2019

Description

Les organismes de standardisation W3C, OASIS et WS-I ont défini un ensemble de spécifications permettant de standardiser les échanges de messages XML entre systèmes informatiques. Il s'agit des spécifications définissant les Web Services à base de messages SOAP. De tels Web Services permettent de faire communiquer entre eux des systèmes jusque là incompatibles. La mise en oeuvre de tels services en Java a été standardisée dans l'API JAX-WS. Ce cours est un tour d'horizon des spécifications SOAP, WSDL et JAX-WS.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs ou architectes souhaitant participer au développement de Web Services SOAP en Java.

Pré-requis

- La connaissance préalable du langage Java est nécessaire.
- Une expérience préalable de XML est un plus, un rappel des concepts est cependant au programme.
- Une expérience préalable de l'API JAXB est recommandée.

Programme

Introduction aux Web Services

Définition des Web Services W3C, intérêt supposé et présentation des standards sous-jacents.

- Les problématiques d'interopérabilité
- La solution apportée par les Web Services
- SOAP et WSDL

Rappels : les bases de XML

Rappels sur les principes de base des langages à balises, les règles syntaxiques, la validation.

- Introduction à XML
- Les règles syntaxiques de XML
- La validation par schéma
- Les espaces de nommage XML

SOAP

Présentation du format des messages SOAP, utilisé dans les Web Services.

- Structure de base d'un message SOAP
- Les balises Enveloppe, Header, Body et Fault
- RPC vs Document
- Soap encoded vs Literal
- Les recommandations du WS-I

WSDL

Le format WSDL est le format standard de description d'un Service Web.

- Structure d'une description WSDL
- Description abstraite d'un service
- Description concrète d'un service
- Les recommandations du WS-I

JAX-WS

Présentation de l'API JAX-WS, API standard de développement de Web Services en Java.

- Présentation de JAX-WS
- Présentation de JAXB
- Possibilités offertes par l'API
- Les annotations
- Implémentations de la spécification

Web Service en Java

Mise en oeuvre de la spécification JAX-WS pour développer des services et des clients.

- Développer un service depuis son implémentation
- Personnalisation du bindign XML / Java
- Générer un client depuis une description WSDL

Les Handlers

Un Handler est un composant coté client ou coté service chargé d'intercepter et de traiter des requêtes et/ou des réponses.

- Les différents types de Handlers avec JAX-WS
- Développer un Handler
- Configurer et déclarer une chaîne de Handlers avec JAX-WS

l'API Provider

L'API Provider, incluse dans JAX-WS permet de développer des services Web traitant directement les messages en XML.

- Présentation de l'API
- Les différents types de Provider
- Configuration et lancement d'un provider

l'API Dispatch

L'API Dispatch, incluse dans JAX-WS, permet de développer des clients de Service Web traitant directement les messages en XML.

- Présentation de l'API
- Développer un client Dispatch

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des services RESTful en Java

Code
CP-REST
Durée
3 jours
Tarif
1490 € HT
Sessions
21/03/2018 au 23/03/2018
16/05/2018 au 18/05/2018
29/10/2018 au 31/10/2018
20/03/2019 au 22/03/2019
15/05/2019 au 17/05/2019
28/10/2019 au 30/10/2019

Description

REST est aujourd'hui de plus en plus utilisé pour la réalisation d'architectures orientées services dans laquelle des systèmes hétérogènes nécessitent de communiquer entre eux par échange de messages sur HTTP. Plus qu'une alternative à SOAP, il s'agit d'une autre façon, complémentaire, de voir la communication entre applications. La spécification JAX-RS (JSR 311) décrit un standard de développement de services REST en Java.

Objectifs

- Comprendre les concepts HTTP et REST
- Effectuer des choix entre les différentes techniques présentées
- Réaliser et déployer une classe de ressource JAX-RS
- Fournir et consommer différents types de données
- Connaître les différentes propositions de publication d'API REST

Participants

Ce cours s'adresse aux responsables de projet, développeurs et architectes désirant comprendre ce qu'est REST et la spécification relative à sa mise en œuvre en Java.

Pré-requis

- Connaissances de base des applications web Java
- Maîtrise du langage Java 5

Programme

Présentation des concepts

Présentation des concepts REST, tel qu'ils ont été décrits par Roy Fielding.

- Representational State Transfer
- Intégration de REST dans une architecture informatique
- Positionnement par rapport à SOAP, gRPC, ...

Le protocole HTTP

Le fonctionnement de REST est, en partie, calqué sur les possibilités offertes par le protocole HTTP.

- Historique de HTTP
- Structure des messages HTTP
- Les méthodes HTTP: GET, POST, PUT, DELETE, ...
- Sécurisation d'une communication HTTP

JAX-RS

JAX-RS est la spécification Java qui standardise le développement de services dits RESTful.

- Java API for RESTful Web Services
- Classe ressource
- Les annotations JAX-RS: @Path, @Produces, @QueryParam, ...
- L'URI pour identifier les ressources
- Le passage de paramètres
- Le déploiement de ressources

TP:réalisation d'une classe de ressources avec JAXRS-RI (Jersey) ou Apache CXF, utilisation de types multimédia (images, ...).

Métadonnées REST et consommation d'une ressource en Java

Aperçu des différentes solutions de description de ressources.

- Propositions en présence: WADL, Swagger, OpenAPI, ...
- La génération de code client ou la découverte dynamique des possibilités ?
- L'approche hypermedia avec HATEOAS
- L'API cliente de JAX-RS

TP:réalisation d'une API REST avec Swagger/OpenAPI et d'un client Java.

JSON et XML

Aperçu des techniques de binding Java - XML - JSON.

- Rappels XML et XSD.
- Usage des flux JSON et XML dans une architecture RESTful
- Les principes essentiels de JAXB

TP:réalisation de ressources disponibles simultanément en JSON et en XML.

REST et la sécurité

Les solutions permettant de sécuriser la consommation de ressources REST.

- Application des concepts d'authentification, intégrité, confidentialité à REST.
- La sécurité des services avec Jersey: annotations, configuration
- Introduction à oAuth

TP:authentification avec Jersey, mise en place de HTTPS.

Eléments d'architecture

Place des services RESTful dans une architecture globale.

- Place des services RESTful dans une architecture multi-niveaux
- REST comme couche d'accès aux données
- REST comme couche d'objets métiers
- REST en tant que services frontaux

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Exploiter les spécificités JavaEE 6

Code
CP-JEE6
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Ce cours présente les différentes évolutions de JavaEE 6 (JPA 2.0, EJB 3.1, JSF 2.0 et servlets 3.0, JAX-RS 1.1, JSP 2.2 et EL 2.2, injection de dépendances, bean validation, ...). Les travaux pratiques sont réalisés avec GlassFish et Eclipse.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien JavaEE et souhaitant découvrir les spécificités de la version 6.

Pré-requis

- Maîtrise du langage Java indispensable
- Connaissance et pratique de JavaEE nécessaires

Programme

Introduction

Rappels sur JavaEE et présentation des ajouts et retraits par rapport à JavaEE 5.

- Les API JavaEE
- Les API devenues obsolètes
- Les nouveautés: profils et extensibilité, ...

EJB 3.1

Présentation des nouveautés concernant les EJB (JPA 2.0, EJB 3.1).

- Entités JPA 2.0 (proxy, lazy loading, cache, fetching, ...)
- EJB Session 3.1 (cycle de vie, accès concurrent, exécution programmée, invocation asynchrone, ...)
- Simplification du packaging
- EJB Lite

servlet 3.0 et JSP 2.2

Présentation des nouveautés concernant les API Web.

- L'API servlet 3.0: fragments, traitements asynchrones, ...
- Les nouvelles possibilités en termes de sécurité
- JSP 2.2 et EL 2.2

JSF 2.0

Présentation de JSF 2.0.

- Les facelets
- Le support d'AJAX
- Les templates et les composants composites

Les web services REST (JAX-RS 1.1)

Présentation de REST et JAX-RS.

- Rappels sur la problématique d'interopérabilité
- L'approche RESTful par rapport à SOAP
- JAX-RS 1.1 et ses annotations

Autres nouveautés

Injection de dépendances, validation, ...

- Injection de dépendances avec CDI 1.0
- Bean Validation
- Généralisation de l'usage des annotations

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Java Persistence API

Code
CP-JPA
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Il est difficile de concevoir des applications n'exploitant pas des données stockées dans un SGBD relationnel. JPA permet de simplifier la réalisation de "passerelles" entre objets Java et données stockées en base.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant apprendre à mettre en place des données Java persistantes.

Pré-requis

- Maîtrise du langage Java nécessaire
- La connaissance des concepts liés aux SGBDR est recommandée

Programme

Introduction
Premières définitions de la notion de persistance.
- Problématiques: "Impedence mismatch"
- Qu'est ce qu'un ORM ?
- Les solutions Java dans les architectures multicouches
- L'API JPA

Concepts de base JPA
Entity et EntityManager.
- La notion d'Entity
- Création d'un EntityManager
- Les opérations de base: création, mise-à-jour, suppression, lecture de données

Bases du mapping objet/relationnel
Mapping avec et sans annotations.
- Les classes d'entités, le mapping des tables et des colonnes
- Les clés primaires
- Les relations entre entités: 1-1, 1-N, M-N
- Le support de l'héritage

La gestion des entités persistantes
Stratégies de chargement, cascade d'opérations, ...
- Les stratégies de chargement des données
- Les opérations en cascade
- Le cycle de vie des objets persistants

Techniques avancées de mapping
Stratégies de chargement, cascade d'opérations, ...
- Relations de composition entre entités, mapping multi-tables
- Clés primaires composées
- La gestion des relations N-M, les tables de jointure
- Les différents types de collections possibles
- Le contrôle des insert et des update

Les requêtes JPQL
Java Persistence Query Language.
- Les principes des SELECT
- Opérations sur les chaînes de caractères et les dates
- Les sous-requêtes
- Les "jointures"
- Les ensembles

Les transactions
Gestion des transactions avec JPA et JTA.
- Gestion des transactions avec JPA
- Principes JTS / JTA
- Container-managed Transactions
- Bean-managed Transactions

Les concurrences d'accès
Comment gérer le compromis intégrité / performances.
- Les niveaux d'isolation d'une transaction
- Les différents modes de verrouillage
- Verrouillage optimiste et @Version

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Java, langage et outils

Descriptif de la filière

La filière JavaSE propose des formations couvrant l'ensemble des technologies basées sur le langage Java (édition standard). Les débutants souhaitant s'initier au langage y trouveront les formations de base qui leur permettront d'aller ensuite vers la pratique des APIs JavaEE ou celles d'autres frameworks (Struts, Hibernate, ...).

Liste des formations de cette filière

Intitulé	Durée	Description
Essentiel Java 8, lambdas et streams	1 jour	Cours pratique de présentation des nouveautés Java 8, notamment les lambda-expressions et les streams.
Les bases du langage Java	3 jours	Formation pratique d'initiation au langage Java, ainsi qu'à quelques API. Destinée aux débutants.
Mettre en place des tests en Java	3 jours	Présentation des APIs, frameworks et outils de tests ainsi que les différents test réalisables dans les applications.
Gérer le cycle de vie d'un projet Java avec Maven	2 jours	Cours pratique de présentation des notions de base de Maven.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel Java 8, lambdas et streams

Code
CP-JAV8
Durée
1 jour
Tarif
690 € HT
Sessions
05/02/2019
26/03/2019
28/05/2019
11/09/2019
06/11/2019

Description

Journée destinée aux développeurs Java confirmés désireux de comprendre les nouveautés essentielles de Java 8 constituées par les lambdas et les streams. Des travaux pratiques sont réalisés.

Objectifs

- Comprendre les concepts de programmation fonctionnelle et de lambda
- Découvrir la puissance des streams
- Utiliser lambda et streams pour rendre les codes Java plus maintenables

Participants

Toute personne désireuse de comprendre la puissance de Java 8.

Pré-requis

- La maitrise de Java 5/6 est requise pour profiter des travaux pratiques.

Programme

Java 8.
Présentation des nouveautés de Java 8.
- La nouvelle philosophie du langage: se concentrer sur ce que l'on veut obtenir, et pas sur la façon de le faire.
- Le contenu détaillé de cette version de Java.

La programmation fonctionnelle.
Présentation des concepts associés à la programmation fonctionnelle.
- Caractéristiques et bénéfices de l'approche fonctionnelle.
- La réification d'une fonction, où comment marier Objets et Fonctions.
- La transparence référentielle et l'optimisation par mémoisation.

Les lambda-expressions en Java 8.
La transcription du concept de programmation fonctionnelle dans le monde Java.
- Liens entre lambda, interface et classe anonyme.
- Le typage d'une lambda-expression, Function, BiFunction, Predicate, Supplier, Consumer...
- L'intérêt des références aux méthodes et aux constructeurs
- Les APIs modifiées pour exploiter les lambda-expressions.

TP:création de lambda-expressions.

Map-reduce et streams.
Présentation des concepts popularisés par Google en 2004.
- L'intérêt du modèle.
- Les opérations map et reduce, les quatre étapes du processus.
- Exemples de mises-en-oeuvre.

Les streams Java 8.
L'API "stream" de Java 8 et ses cas d'utilisation.
- Les principes de base du streaming.
- L'usage conjoint des streams et des lambda-expressions.
- La parallélisation des traitements avec parallelStream().
- La bonne utilisation du pool de threads Java.
- Points d'attention et bonnes pratiques.

TP:utilisation de parallelStream().

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Les bases du langage Java

Code
CP-JAVA1
Durée
3 jours
Tarif
1490 € HT
Sessions
24/04/2019 au 26/04/2019
28/10/2019 au 30/10/2019

Description

Ce cours est une première approche du langage Java. Après présentation des concepts Objet et de la syntaxe du langage, le stagiaire apprend à développer des applications interactives en Java. Des exercices pratiques permettent d'illustrer et de mettre en oeuvre les concepts abordés.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse à toute personne amenée à travailler avec le langage Java et / ou souhaitant s'initier à ce langage.

Pré-requis

- La connaissance et la pratique d'un langage de programmation structurée sont fortement recommandées.
- La connaissance préalable des concepts Objet est nécessaire.

Programme

Syntaxe de base du langage Java

Présentation des éléments syntaxiques de base du langage : la structure en blocs, la déclaration des variables, la déclaration d'une méthode, ...

- Déclaration et initialisation d'une variable
- Manipulation des variables, opérations arithmétiques
- Signature et déclaration d'une méthode
- La structure en blocs

Contrôle du flot d'exécution

Apprentissage des boucles (for, foreach, while, do while) et des conditions (if, switch, opérateur ternaire, assertions).

- Itérer avec les boucles for / while
- Tests à base de if / else if / else
- Les expressions booléennes en Java

Classes et interfaces

Déclaration de classes et d'interfaces en Java, contrôle de la visibilité des membres, classes abstraites, ...

- Déclaration d'une classe / interface en Java
- Contrôle de la visibilité d'une classe et de ses membres
- Héritage, polymorphisme

Les exceptions

Présentation du système de gestion des erreurs par exception. Présentation des exceptions courantes. Différence en erreurs, exceptions et RuntimeException.

- Exceptions et erreurs, exceptions gérées / non gérées
- Gérer ou propager une exception
- Les blocs try / catch / finally

Entrées / Sorties en Java

Apprentissage des classes de base du package java.io pour lire, écrire (du texte, des objets, ...) dans des flux (fichiers, flux réseaux, ...).

- Le package java.io
- Accès au système de fichier
- Les classes Reader / Writer
- Lecture / Ecriture bufferisée
- Sérialisation

Les énumérations

Apprentissage des énumérations, types apparus avec la version 5 de Java.

- Déclaration d'une énumération
- Manipulation des membres d'une énumération
- Déclaration de membres dans un type énumération

Collections et généricité

Présentation des structures de données dites "Collection" en Java (listes, piles, queues, ...). Présentation de la généricité en Java.

- Les méthodes equals et hashcode
- Collection, List, Set et Map
- Les implémentations, leurs différents usages
- La généricité en Java

Les annotations

Présentation des annotations, nouveau type de données apparus dans Java 5. Utilisation des annotations standard, création et utilisation d'une annotation.

- Présentation et utilité des annotations
- Présentation des annotations communes du langage Java
- Créer et utiliser ses propres annotations

Le packaging d'une application Java

Description du format d'archive jar et des métadonnées associées.

- Le format jar
- Métadonnées d'une archive jar

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Mettre en place des tests en Java

Code
CP-TEST
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

La mise en place de tests est toujours un point compliqué dans les applications. Que ce soit le choix de ce que l'on doit effectivement tester, mais également quels sont les outils les plus adaptés à ce besoin. Cette formation a pour but de présenter quoi et comment tester ?

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Toutes personnes devant mettre en place différents types de tests.

Pré-requis

- Une bonne connaissance de Java est nécessaire. La connaissance du développement Web en Java est un plus.

Programme

Pourquoi tester ?

Avant de de commencer à réfléchir sur la façon de mettre au point des tests, il est important de regarder pourquoi tester et ce que cela implique dans un prjet informatique.

- L'utilité du test
- L'impact des tests sur un projet
- Les méthodes de test

JUnit

JUnit est l'API de test la plus connus dans le monde Java. Elle permet de mettre en place des tests unitaires

- Présentation de JUnit
- Les assertions
- Les classes de test
- Les fakes et mocks
- Les frameworks de génération de mock

Le framework Mockito

Présentation de Mockito et utilisation avec Eclipse.

- Présentation de Mockito
- L'architecture de Mockito
- Utilisation de Mockito

La couverture de tests

Une chose difficile a déterminer lors de la mise en place de tests unitaires est de s'assurer de ce qui a été testé. On va pour cela utiliser des outils qui analiserons la couverture effective de ces test.

- La couverture de tests
- Les outils de couverture de test

TestNG

Présentation de TestNG qui est un framework de test qui se veut une évolution et une amélioration de JUnit.

- Présentation de TestNG
- Les différents moyen d'utiliser TestNG
- Présentation de JMockit
- Utilisation de TestNG pour faire des tests unitaires

Le test d'applications Java EE

Les applications Java EE sont de par leur architecture difficiles à tester. Cependant, JUnit ne permet pas de faire cela. Pour réaliser les tests, on fera appel à des frameworks tel que cactus et JUnitEE.

- La problématique des tests pour les applications Java EE
- Le framework cactus
- Le framework JUnitEE

Les tests fonctionnels et les tests de charge

Afin de s'assurer de la validité (fonctionnelle) d'une application et de sa résistance face à l'augmentation du nombre d'utilisateurs, on va devoir également de mettre en place des tests. On va pour cela utiliser des outils tels que JMeter et eValid.

- Les tests fonctionnels et tests de charge
- Les outils de test fonctionnel
- Présentation et utilisation de JMeter

Tests d'intégration et de recette

Les tests d'intégration et de recette sont extrémement important pour assurer d'un bon fonctionnement et la disponibilité de l'application mais également qu'aucune interaction ne s'effectue avec les autres applications.

- Qu'est-ce qu'un test d'intégration ?
- Mise en place de tests d'intégration
- Le rôle du test de recette et sa mise en place

Conclusion sur les test et les outils associés

Une conclusion sur la façon d'utiliser

- Quelle conclusion apporter sur la mise en place de tests ?
- Les serveurs d'intégration continue

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Gérer le cycle de vie d'un projet Java avec Maven

Code
CP-MAV
Durée
2 jours
Tarif
1290 € HT
Sessions
01/04/2019 au 02/04/2019
09/09/2019 au 10/09/2019

Description

Maven est un outil de build très utilisé dans le monde Java. Cette formation permet d'en appréhender toutes les facettes.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Développeur, techlead, architecte Java / JavaEE.

Pré-requis

- La pratique du développement Java est indispensable.

Programme

Présentation de Maven
Présentation du projet Maven.
- Le projet Apache Maven
- L'historique et la communauté maven
- Les modes d'utilisation de Maven (autonome, intégré dans un IDE)
- Le fichier POM

TP: fabrication d'un projet fourni.

Concepts essentiels Maven
Les notions de cycle de vie, dépendances, modules, ...
- Les différentes phases du cycle de vie
- La gestion des dépendances
- Les référentiels et plugins
- La notion d'héritage entre projets
- Le travail avec plusieurs modules
- L'exécution d'une tâche et la gestion des erreurs

TP: création et configuration d'un projet Java Maven.

Utilisation avancée
Les concepts avancés de Maven.
- Les tests
- Les propriétés
- Le filtrage des ressources
- Les profils
- Les archetypes et assembly

TP: exploiter les concepts Maven avancés.

Livraison
La livraison avec Maven.
- La livraison avec un gestionnaire de code source
- Le processus de Release
- La production de rapport et la mesure de qualité
- L'utilisation dans un cadre d'intégration continue

TP: livrer avec Maven.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

.Net CORE

Descriptif de la filière

.NET Core est la version ouverte et normalisée du framework .NET Microsoft. Ce framework permet de construire des applications portables en environnement Windows, Linux et MacOS. Cette filière propose des formations permettant d'y parvenir.

Liste des formations de cette filière

Intitulé	Durée	Description
Programmer avec C#	3 jours	Introduction au langage de programmation C# et aperçu de quelques technologies associées.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Programmer avec C#

Code
CP-CS01
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Le langage C# est le langage de programmation orienté Objet le plus complet de la plateforme .NET. Simple et élégant, il est à la fois efficace et facile d'accès et peut donc être assimilé par tous.

Objectifs

- Utiliser les types de données et instructions C#
- Créer des classes et mettre en oeuvre le polymophisme
- Compiler, exécuter et mettre au point une application

Participants

Ce cours s'adresse aux développeurs désireux de s'initier au langage C#.

Pré-requis

- La connaissance des concepts Objet et la pratique de la programmation sont nécessaires.
- La connaissance d'un langage Objet (C++, Java, ...) est un plus, mais n'est pas obligatoire.

Programme

Introduction
- Motivations ayant présidé à la création du langage
- Normalisation et aspects propriétaires
- Concepts de base: machine virtuelle, gestion mémoire, ...
- Positionnement par rapport aux langages comparables et cas d'usage

TP: analyse d'un premier programme C# fourni.

Bases du langage C#
Aspects syntaxiques, types de données et instructions du langage.
- Expressions, types et variables
- Typage fort et inférence de type
- Boucles et conditions
- Espaces de nommage

TP: réalisation d'applications procédurales mettant en oeuvre les concepts présentés.

Principes Objet appliqués en C#
Ce module aborde la programmation Objet avec C#.
- Définition d'une classe avec variables d'instances et opérations
- Propriétés, Indexeurs, Structures
- Héritage et polymorphisme
- Interfaces

TP: utilisation des propriétés avec génération des variables d'instances, réalisation d'applications basées sur le polymorphisme.

Notions avancées du langage C#
Mécanismes avancés et spécifiques du langage.
- Evènements, Delegates
- Enumérations, Attributs
- Contrôle fin du polymorphisme
- Cycle de vie des objets et allocation sur le tas ou sur la pile

TP: utilisation du mécanisme de delegate.

Généricité
Présentation des mécanismes de programmation générique C#
- Concept de généricité
- Utilisation d'une classe générique
- Définition d'une classe générique
- Contrôles sur les types à l'instanciation (where)

TP: création d'une classe générique.

Gestion des exceptions en C#
La gestion d'erreurs par exceptions.
- Principe des exceptions
- Le bloc try / catch
- Le bloc finally

TP: utilisation des exceptions.

Lambda-expressions
La programmation fonctionnelle en C#.
- Principes de programmation fonctionnelle
- Définition et utilisation d'une lambda-expression

TP: déclaration et appel de lambda-expressions.

Introduction au framework de base (Unified BCL)
Présentation de quelques mécanismes et classes de base (Unified BCL)
- Tour d'horizon sur l'ensemble des classes
- Structures de données (Collections)

TP: utilisation de structures de données (Collections).

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Android et iOS

Descriptif de la filière

Liste des formations de cette filière

Intitulé	Durée	Description
Développer des applications Android avec Kotlin	5 jours	Apprendre à développer des applications Android en Kotlin.
Moderniser vos apps Android avec Kotlin et les meilleures toolkits	3 jours	Développer des apps Android en Kotlin et exploiter les outils récents.
Développer sous Android	4 jours	Apprendre à développer des applications Android en Java.
Développer pour iOS avec Swift	3 jours	Présentation des bases de programmation d'applications Swift pour iOS (iPhone, iPad, ...).

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer des applications Android avec Kotlin

Code
CP-ANDK
Durée
5 jours
Tarif
2390 € HT
Sessions
20/05/2019 au 24/05/2019
21/10/2019 au 25/10/2019

Description

Ce cours vous présente les fondamentaux du développement d'applications à destination des appareils tournant sous le système d'exploitation Android. Il couvre le cycle de développement depuis la conception jusqu'au déploiement de l'application. Les travaux pratiques sont réalisés en Kotlin avec Android Studio.

Objectifs

- Maîtriser les bases de Kotlin et d'Android Studio
- Concevoir, réaliser et déployer une application Android
- Appréhender les APIs essentielles d'Android

Participants

Architecte, Techlead, Développeur désirant aborder la plateforme Android à travers le langage Kotlin.

Pré-requis

- La pratique d'un langage de programmation (idéalement Java) est indispensable.
- Les bases du langage Kotlin sont étudiées pendant cette formation.

Programme

Introduction à la plateforme Android

Présentation du système d'exploitation et de ses possibilités.

- Positionnement d'Android
- Versions d'Android et versions d'API
- Ecosystème Android avec le projet AOSP et Google
- Langages de développement et NDK/SDK

Langage Kotlin

Les concepts de base de Kotlin et la mise en application avec Android Studio.

- Positionnement de Kotlin par rapport à Java
- Syntaxe et concepts de base (types, instructions, classes, ...)
- Délégation et généricité
- Lambda-expressions, collections, exceptions, ...
- Prise en main de l'environnement de développement Android Studio

TP: analyse d'un code Kotlin, création d'un projet dans Android Studio.

Cycle de vie d'une application

Cycle de développement d'une application Android.

- Structure d'un projet Android
- Test et debug avec l'émulateur et un périphérique réel
- Signature et déploiement d'une application Android

TP: réalisation et déploiement d'une première application.

Bases du développement Android

Concepts et classes de base.

- Notions d'Activity, Fragment, Intent
- Le manifeste de l'application
- La gestion de l'application par le système (processus, threads, ...)

TP: conception d'une application avec plusieurs écrans.

Gestion de l'interface graphique

Eléments graphiques de base, positionnement des éléments sur un écran.

- ViewGroup et view, les Layout
- Composants graphiques de base
- La gestion des événements
- Créer des vues en XML ou par code
- ActionBar et menus

TP: mise en oeuvre d'éléments graphiques complexes.

Persistance de données

Accès au système de fichiers. Accès à une base de données.

- Les différents types de stockages (interne, externe, sd, ...)
- Accès en lecture et/ou écriture au système de fichiers
- La gestion des préférences
- Utilisation de SQLite

TP: persistance de données depuis une application.

Communications avec des systèmes externes

Utilisation de sockets réseau, consommation de services web, communication HTTP.

- Communications HTTP
- Communications par sockets
- Accès à des ressources REST et exploitation de données JSON

TP: consommation de ressources REST depuis une application Android.

Multimédia

Lecture audio et video.

- Utiliser le lecteur multimedia dans ses applications
- Afficher des graphiques
- Jouer des animations

TP: réalisation d'une application multimédia.

Aspects avancés

Les mécanismes complexes de la plateforme.

- Maîtriser les threads
- La communication à base d'Intents
- La notion de service
- Utiliser les APIs Google (Maps, ...)
- Gestion des notifications

TP: réalisation d'un service, utilisation d'API Google.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Moderniser vos apps Android avec Kotlin et les meilleures toolkits

Code
CP-ANDK2
Durée
3 jours
Tarif
1590 € HT
Sessions
27/05/2019 au 29/05/2019
28/10/2019 au 30/10/2019

Description

Ce cours s'adresse aux développeurs Android désireux de migrer vers Kotlin. Il présente également un ensemble d'outils et de pratiques permettant d'améliorer la qualité des développements. Les travaux pratiques sont réalisés en Kotlin avec Android Studio.

Objectifs

- Connaître le langage Kotlin
- Découvrir des outils et librairies Android
- Exploiter des APIs avancés d'Android

Participants

Architecte, Techlead, Développeur désirant optimiser leurs pratiques de développement Android à travers le langage Kotlin.

Pré-requis

- La pratique courante du développement Android est indispensable.
- Les bases du langage Kotlin sont étudiées pendant cette formation.

Programme

Introduction

Rappels Android.
- Les dernières évolutions de l'éco-système Android
- Versions d'Android et versions d'API
- Langages de développement et NDK/SDK

Langage Kotlin

Les concepts de base de Kotlin et la mise en application avec Android Studio.
- Positionnement de Kotlin par rapport à Java
- Syntaxe et concepts de base (types, instructions, classes, ...)
- Délégation et généricité
- Lambda-expressions, collections, exceptions, ...
- Prise en main de l'environnement de développement Android Studio

TP: analyse d'un code Kotlin, création d'un projet dans Android Studio.

Outils avancés pour le développement

Cycle de développement d'une application Android.
- Paramétrer le build Gradle et maîtriser les build variants
- Analyser et améliorer le code avec Lint
- Optimiser une application avec ProGuard
- Mettre au point et profiler une application

TP: paramétrer Gradle, utiliser les différents outils.

Automatiser le test

Mettre en place l'automatisation des tests applicatifs Android.
- Présentation des outils adaptés aux différents types de tests
- Simulation d'interactions utilisateur avec Espresso
- Paramétrage d'Android Studio pour les tests
- Utilisation de Cloud Test Lab

TP: automatiser les tests.

Toolkits et services indispensables

Découvrir des outils, services et librairies incontournables.
- Utiliser les Google Play Services
- Gérer les dépendances par injection avec Dagger
- Optimiser les chargements d'images avec Picasso
- Simplifier la consommation de ressources REST avec Retrofit

TP: déployer et utiliser des librairies parmi l'ensemble présenté.

APIs avancées

Utiliser des APIs et mécanismes Android avancés.
- Utiliser Material Design dans les IHM
- Créer des widgets
- Obtenir les informations de capteurs (accéléromètre, température, ...)
- Géolocaliser (wifi, gps, ...)

TP: créer une IHM élaborée, géolocaliser.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer sous Android

Code
CP-AND1
Durée
4 jours
Tarif
1890 € HT
Sessions
13/03/2019 au 15/03/2019
28/10/2019 au 30/10/2019

Description

Ce cours vous présente les fondamentaux du développement d'applications à destination des appareils tournant sous le système d'exploitation Android. Il couvre le cycle de développement depuis la conception jusqu'au déploiement de l'application. Les travaux pratiques sont exécutés avec Android Studio.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Développeur désirant aborder la plateforme Android à travers le langage Java.

Pré-requis

- La maîtrise du langage Java et de XML sont nécessaires.

Programme

Introduction à la plateforme Android
Présentation du système d'exploitation et de ses possibilités.
- Positionnement d'Android
- Versions d'Android et versions d'API
- Possibilités offertes par l'API

Cycle de vie d'une application
Cycle de développement d'une application Android. Android Studio.
- Prise en main de l'environnement de développement Android Studio
- Structure d'un projet Android
- Test et debug avec l'émulateur et un périphérique réel
- Signature et déploiement d'une application Android

Bases du développement Android
Concepts et classes de base.
- Notions d'Activity, Fragment, Intent
- Le manifeste de l'application
- La gestion de l'application par le système (processus, threads, ...)

Gestion de l'interface graphique
Eléments graphiques de base, positionnement des éléments sur un écran.
- ViewGroup et view, les Layout
- Composants graphiques de base
- La gestion des événements
- Créer des vues en XML ou par code
- ActionBar et menus

Persistance de données
Accès au système de fichiers. Accès à une base de données.
- Les différents types de stockages (interne, externe, sd, ...)
- Accès en lecture et/ou écriture au système de fichiers
- La gestion des préférences
- Utilisation de SQLite

Communications avec des systèmes externes
Utilisation de sockets réseau, consommation de services web, communication HTTP.
- Communications HTTP
- Communications par sockets
- Accès à des ressources REST et exploitation de données JSON

Multimédia
Lecture audio et video.
- Utiliser le lecteur multimedia dans ses applications
- Afficher des graphiques
- Jouer des animations

Aspects avancés
Les mécanismes complexes de la plateforme.
- Maîtriser les threads
- La communication à base d'Intents
- La notion de service
- Utiliser les APIs Google (Maps, ...)

Nouveautés
Nouveautés Android.
- La charte graphique Material Design
- Les notifications
- Les modifications en matière de stockage, réseau et multimedia

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développer pour iOS avec Swift

Code
CP-IPH1
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

Ce cours vous présente les fondamentaux du développement d'applications à destination des appareils tournant sous le système d'exploitation iOS (iPhone, iPod Touch, iPad). Cette formation pratique en aborde les aspects essentiels, depuis la réalisation jusqu'au déploiement sur l'AppStore.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Développeur désirant aborder la plateforme iOS avec Swift.

Pré-requis

- De bonnes connaissances Swift sont nécessaires pour profiter de cette formation.

Programme

Introduction à la plateforme iOS

Présentation de la plateforme sous ses aspects économiques et techniques.

- Les différentes versions du système et des modèles l'utilisant
- Les spécificités de la plateforme
- Les langages de programmation Objective-C et Swift
- L'AppStore, le programme de développement Apple

Outils et environnement de développement

Démarrage d'un premier projet à l'aide des outils disponibles.

- La configuration matérielle nécessaire au développement
- La liste des librairies et frameworks
- Les outils et IDE disponibles (XCode, InterfaceBuilder, simulateur, ...)
- La préparation d'un projet à destination d'une cible (iPhone, iPod, iPad)
- Le mécanisme de signature

Introduction à l'API Cocoa Touch

Présentation des fonctionnalités essentielles de la version Cocoa dédiée à iOS.

- Les frameworks constitutifs (Foundation, UIKit)
- La hiérarchie des classes et design-patterns de base (Delegate, Observer, MVC)
- Les principes du pattern Model-View-Controller (MVC)

L'interface utilisateur avec UIKit

Principes de base de l'IHM sous iOS.

- Les concepts de base (fenêtres et vues)
- La hiérarchie et l'affichage des Views
- La gestion des événements (UIResponder)
- Quelques types de vues avancées (UITableView, UICollectionView)

Le stockage des données

Mécanismes permettant l'accès aux données persistantes (fichiers, bases, ...).

- L'emploi de CoreData et NSCoding
- La gestion des préférences utilisateur

Communications avec des systèmes externes

Utilisation de sockets réseau, consommation de services web, communication HTTP.

- La communications Sockets et HTTP
- L'invocation de services Web (REST) et le parsing JSON
- Les autres protocoles réseau

Multimédia

Lecture audio et video.

- Afficher des vidéos et jouer des sons et musiques
- Introduction à Quartz, Core Graphics, Open GL, Core Animation, Core Audio, Video

Périphériques et composants spéciaux

Exploiter les ressources "spéciales" (GPS, capteur, ...).

- La géo-localisation au sein des applications
- Les capteurs: proximité, accéléromètre
- Le capteur photo/vidéo

Distribuer ses applications

Les mécanismes prévus par Apple.

- Les différentes modalités de distribution (AppStore, ...)
- Le programme de développement Apple

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Modelisation

Descriptif de la filière

Cette filière présente quelques standards de modélisation parmi les plus utilisés, tels qu'UML et BPMN.

Liste des formations de cette filière

Intitulé	Durée	Description
Essentiel Objet, les concepts et la pratique	1 jour	Introduction pratique aux concepts Objet.
Savoir modéliser avec UML	3 jours	Cours pratique de présentation du formalisme et de l'usage des divers diagrammes UML 2.
Modéliser des processus avec BPMN 2.0	2 jours	Cours pratique de modélisation de processus à l'aide du formalisme BPMN 2.0.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel Objet, les concepts et la pratique

Code
CP-OBJ1
Durée
1 jour
Tarif
590 € HT
Sessions
sur demande

Description

Cette formation se donne pour objectif de clarifier les concepts tels que l'Objet, la classe, l'encapsulation, l'héritage ou le polymorphisme. Les exercice proposés se basent sur le langage Java.

Participants

Ce cours s'adresse aux développeurs, architectes, chefs de projet ou tout autre personne ayant à participer à un projet manipulant des concepts Objet (en phase d'analyse, de conception ou de développement).

Pré-requis

- Connaissances de base en conception d'applications et en développement logiciels

Programme

Les concepts de base du modèle Objet
Présentation des concepts Objet de base: classe, héritage, ...
- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme
- Principes fondamentaux en conception Objet
- Erreurs à éviter
- Sensibilisation aux concepts avancés: généricité, interfaces, ...

L'objet en programmation
Exemples de mise en pratique avec un langage de programmation Objet.
- La définition de classes
- La réalisation de méthodes
- L'instanciation et la gestion mémoire

Introduction à la modélisation Objet
Présentation des bases du diagramme de classes UML.
- Présentation générale d'UML 2: les différents types de diagrammes, différences entre diagrammes statiques et dynamiques
- La place d'UML dans le processus de développement
- La représentation de classes à l'aide du diagramme de classes

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Savoir modéliser avec UML

Code
CP-UML1
Durée
3 jours
Tarif
1490 € HT
Sessions
22/05/2019 au 24/05/2019

Description

Ce cours permet de préciser la place de la modélisation UML, et plus particulièrement ses divers diagrammes, dans le processus de développement d'une application. Ce cours s'adresse aussi bien aux concepteurs désireux de formaliser leur modèle Objet en UML, qu'aux responsables MOA souhaitant communiquer avec leur Maîtrise d'Oeuvre en utilisant un langage commun.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Développeurs, concepteurs, analystes, architectes, chefs de projet, responsables MOA désireux de découvrir ou d'approfondir leurs connaissances de UML.

Pré-requis

- Aucun pré-requis pour ce cours. La connaissance préalable des concepts Objet est cependant un plus.

Programme

Rappels : les concepts Objet

Rappel des concepts Objet, base de toute conception efficace.

- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme

Introduction à UML

Historique et philosophie de UML. Différence entre le formalisme et la méthode.

- La formalisation, l'apport du modèle, différence entre la modélisation fonctionnelle et Objet, les langages de modélisation
- L'historique d'UML, modes d'utilisation, outillage
- Présentation générale de UML 2 : les diagrammes, modélisation statique et dynamique
- Stéréotypes et l'extension d'UML

Processus de développement

La place d'UML dans le processus de développement.

- Processus de développement et cycle de vie logiciel
- Le Processus Unifié : RUP, 2TUP, ...
- Relations entre 2TUP et UML

Diagramme de cas d'utilisation

Formalisme et utilité des diagrammes de cas d'utilisation.

- Acteurs, Cas d'utilisation, frontières du système
- Relations entre entités dans le Use Case Diagram

Diagramme de classes, d'objets et de packages

Formalisme du diagramme de classes. Représentation des relations entre classes.

- Représentation des classes, attributs, opérations
- Relations entre classes : association, agrégation et héritage
- Visibilité des membres d'une classe
- Classes abstraites et interfaces
- Diagrammes d'objets
- Diagramme de packages

Diagramme de séquences et diagramme de communication

Formalisme de présentation d'une séquence d'interaction entre objets.

- Scénario, ligne de vie d'un objet
- Echange de messages synchrones / asynchrones
- Les cadres UML 2
- Diagramme de communication

Diagramme d'activités

Formalisme de présentation d'un workflow ou processus en UML.

- Notion d'action, d'activité
- Les couloirs d'activité

Diagramme d'états transitions

Formalisme de présentation du changement de l'état d'un objet.

- Etats et transitions
- Gardes, actions

Diagramme de structure composite

Formalisme de présentation de l'intérieur d'un système en UML.

- Notions fondamentales du formalisme

Diagramme de composants / déploiement

Représentation d'une architecture en UML.

- Diagramme de composants
- Diagramme de déploiement

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Modéliser des processus avec BPMN 2.0

Code
CP-BPMN
Durée
2 jours
Tarif
1290 € HT
Sessions
11/03/2019 au 12/03/2019
21/10/2019 au 22/10/2019

Description

Ce cours permet d'appréhender les possibilités de la notation BPMN dans le domaine de la modélisation des processus métier. Les notions sont mises en pratique à l'aide d'un outil BPMN open-source.

Objectifs

- Comprendre les besoins liés à la modélisation de processus
- Appréhender les possibilités du standard BPMN et les différents types de diagrammes disponibles
- Réaliser des modélisations à l'aide d'un outil BPMN

Participants

Développeurs, concepteurs, analystes MOE ou MOA, architectes, chefs de projet, responsables MOA désireux de mettre en pratique BPMN.

Pré-requis

- Sensibilisation à la notion de modélisation.
- Pratique des outils informatiques usuels.

Programme

Introduction

Présentation de BPMN
- La place de BPMN dans l'offre de l'OMG et son utilité
- Les types de processus modélisables avec BPMN (stratégique, opérationnels, exécutables)
- Les liens avec le BPM, les workflows, les diagrammes d'activités UML, le concept d'orchestration
- Les outils et l'Ingénierie dirigée par les modèles

La notation BPMN

Présentation des entités BPMN
- Les éléments de base: processus, tâche, gateway, lane, évènements, ...
- La modélisation des règles métier
- La structuration: sous-processus, pool, flows, ...
- Les chorégraphies et les conversations

TP: réalisation de différents types de diagrammes, à difficulté croissante, à l'aide d'un outil graphique

Les processus stratégiques

Les éléments de modélisation spécifiques aux processus stratégiques
- Définition d'un processus stratégique
- Les élements BPMN correspondant et leurs règles d'usage
- Les cas nominaux, les scénarios alternatifs, les exceptions
- Les flux de messages liés aux participants externes

TP: modélisation de processus stratégiques

Les processus opérationnels

Les éléments de modélisation spécifiques aux processus opérationnels
- Définition d'un processus opérationnel
- Les élements BPMN correspondant et leurs règles d'usage
- Les processus des participants
- Les exceptions et transitions

TP: modélisation de processus opérationnels

Les processus exécutables

Les spécificités des processus exécutables
- Définition d'un processus exécutable
- La représentation XML et le lien avec BPEL
- Les moteurs d'exécution

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Essentiels 1 jour

Descriptif de la filière

Cette filière présente des formations condensées en une journée, de façon à acquérir rapidement l'essentiel des concepts liés à une thématique.

Formations rattachées

Intitulé	Durée	Description
Essentiel IoT, panorama Internet des Objets.	1 jour	Séminaire d'une journée permettant de découvrir le domaine de l'IoT à travers une présentation de ses différents aspects.
Essentiel LoRaWAN, émettre et traiter des messages IoT.	1 jour	L'objectif de cette journée pratique est de découvrir la norme LoRaWAN et de réaliser la mise en place des différents éléments.
Essentiel GIT, mettre en oeuvre le contrôle de versions	1 jour	Cours pratique de présentation des notions de base du logiciel de gestion de versions GIT.
Essentiel Docker, mise-en-oeuvre	1 jour	Cours pratique de présentation des notions de "conteneurisation" et leur utilisation dans l'environnement Docker
Essentiel Java 8, lambdas et streams	1 jour	Cours pratique de présentation des nouveautés Java 8, notamment les lambda-expressions et les streams.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Descriptif de la filière

Cette filière regroupe les formations disponibles en cours spécifiques sur demande. N'hésitez-pas à contacter notre équipe pour obtenir une proposition adaptée à vos besoins.

Liste des formations de cette filière

Intitulé	Durée	Description
Les concepts Objet	3 jours	Introduction aux concepts Objet.
Design Patterns et éléments de conception	3 jours	Présentation des Design Patterns et des bonnes pratiques de conception.
Mapping Objet/Relationnel avec Hibernate	3 jours	Mise en place d'un mapping Objet/ Relationnel en utilisant Hibernate.
Essentiel Web Java, Servlets et JSP	1 jour	Développement d'applications web en JavaEE à base de Servlets et de JSP.
Développement JavaScript avec AngularJS	4 jours	Développement d'applications Web en utilisant le framework Javascript de Google AngularJS.
Développement Web avec HTML5	3 jours	Développement d'applications Web en utilisant le langage HTML5.
Mise en forme HTML avec CSS	2 jours	Utilisation de CSS pour pour améliorer les aspects visuels des applications Web.
Développement JavaScript	3 jours	Développement d'applications Web en utilisant le langage JavaScript.
Développement d'applications Web 2.0 avec JQuery	3 jours	Développement d'applications Web 2.0 JavaScript avec JQuery.

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Les concepts Objet

Code
CP-OBJT
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

L'usage des langages Objet et de la modélisation avec UML est de plus en plus répandu dans les entreprises. Si la syntaxe d'un langage ou le formalisme UML s'apprennent relativement facilement, les concepts sous-jacents (les concepts de l'orienté Objet) sont peut être plus difficiles à appréhender. Cette formation se donne pour objectif de clarifier les concepts tels que l'Objet, la classe, l'encapsulation, l'héritage ou le polymorphisme. Les langages Objet ainsi que UML y sont également présentés.

Participants

Ce cours s'adresse aux développeurs, architectes, chefs de projet ou tout autre personne ayant à participer à un projet manipulant des concepts Objet (en phase d'analyse, de conception ou de développement).

Pré-requis

- Connaissances de base en conception d'applications et en développement logiciels

Programme

Paradigme de programmation orientée Objet

Rappel des différentes approches de programmation

- Une taxonomie des principaux paradigmes de programmation
- L'historique des langages orientés Objet

Les concepts de base du modèle Objet

Les notions de base et la présentation des principes qui, si ils sont suivis, permettent d'aboutir à une "conception efficace". Présentation des critères d'efficacité d'une conception.

- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme
- Principes fondamentaux en conception Objet
- Erreurs à éviter

Les concepts avancés du modèle Objet

Notions plus avancées en programmation orientée Objet.

- Classes et méthodes abstraites
- Interfaces
- Généricité

Méthodologie orientée Objet

Différences entre le formalisme et la méthode.

- La modélisation, l'apport de modèles, différences entre la modélisation fonctionnelle et Objet
- Processus de développement et cycles de vie de logiciels (processus unifié: RUP, 2TUP, ...)
- Langages de modélisation

Modélisation Objet

Historique et philosophie d'UML. UML et le processus de développement Objet.

- L'historique d'UML, modes d'utilisation, outillage
- Présentation générale d'UML 2: les différents types de diagrammes, différences entre diagrammes statiques et dynamiques
- La place d'UML dans le processus de développement
- Stéréotypes et extensions d'UML

UML (Unified Modeling Language)

Présentation du langage

- Diagrammes principaux
- De la modélisation à l'implémentation: comment passer de modèle UML à code

Langages de développement Objet

Initiation aux langages Java et C#

- Caractéristiques des langages de programmation Objet
- Java et C#: présentation générale, les concepts Objet avec ces langages et quelques exemples

Design Patterns

Qu'appelle-t-on Design Patterns ? A quoi servent ils ? Comment se présentent ils ?

- Objectifs des Design Patterns
- Formalisme
- Catégories des patterns GOF
- Les autres patterns

Les plates-formes Objet actuelles et les composants

Présentation des plates-formes Objet et les technologies .NET et J2EE

- Définitions: plate-forme, composant, librairies de classes
- Technologies .NET de Microsoft et J2EE de SUN. Comparaison de leurs points forts et leurs points faibles

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Design Patterns et éléments de conception

Code
CP-PATT
Durée
3 jours
Tarif
1590 € HT
Sessions
sur demande

Description

La pratique de la conception Objet n'est pas chose aisée. Une fois compris les principes fondateurs de l'Objet, on réalise que leur connaissance n'est pas suffisante et que quelques règles de bonnes pratiques seraient les bienvenues lors d'une conception. Le but de ce cours est de présenter ces bonnes pratiques. Il s'agit aussi bien de Design Patterns (plus particulièrement les patterns GOF) que des principes généraux de conception.

Participants

Ce cours s'adresse aux développeurs, concepteurs ou architectes désireux d'approfondir leurs connaissances de la conception Objet.

Pré-requis

- Une bonne maîtrise des concepts Objet est nécessaire
- Une expérience de la conception Objet est fortement recommandée

Programme

Rappels : les concepts Objet
Rappel des concepts Objet, base de toute conception efficace.
- Notions d'Objet, abstraction, classe, instance, encapsulation
- L'héritage, le polymorphisme

Principes fondamentaux de conception
Présentation des principes qui, si ils sont suivis, permettent d'aboutir à une "conception efficace". Présentation des critères d'efficacité d'une conception.
- Objectifs de la conception efficace
- Principe d'ouverture / fermeture
- Principe de substitution de Liskov
- Principe de connaissance minimale

Principe d'organisation en package
L'élément de base de la réutilisabilité est le package. De cela découle un ensemble de principes.
- Principe d'équivalence / réutilisation
- Principe de réutilisation commune
- Principe de fermeture commune
- Principe des dépendances acycliques
- Principe de relation dépendance / stabilité
- Principe de stabilité des abstractions

Principes de construction des classes
Ensemble de principes permettant de déterminer quelles classes créer et les relations à créer entre elles pour obtenir une conception efficace.
- Principe d'inversion de dépendance
- Principe de séparation des interfaces
- Principes de GRASP

Design Patterns
Qu'appelle-t-on Design Patterns ? A quoi servent ils ? Comment se présentent ils ?
- Objectifs des Design Patterns
- Formalisme
- Catégories des patterns GOF
- Les autres patterns

Patterns de création
Revue des Design Pattern GOF de type "Création".
- Factory, Singleton
- Prototype, Builder, AbstractFactory

Patterns structurels
Revue des Design Pattern GOF de type "Structure".
- Bridge, Composite, Facade, Decorator
- Proxy, Adapter, Flyweight

Patterns comportementaux
Revue des Design Pattern GOF de type "Comportement".
- Iterator, Observer, State
- Template Method, Strategy, Mediator, Chain of responsability
- Commande, Visitor, Interpreter

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Mapping Objet/Relationnel avec Hibernate

Code
CP-HIB
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Il est devenu courant aujourd'hui d'avoir besoin de mettre en place une couche d'accès aux données persistantes. Celle-ci se révèle rapidement complexe à développer et on va préférer se tourner vers une API existante. Hibernate est à ce jour l'API de persistance de référence en Java.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant apprendre à mettre en place des applications sur la persistance des données.

Pré-requis

- Maîtrise du langage Java nécessaire
- La connaîssance de SQL et d'un SGBD

Programme

Le mapping Objet/Relationnel

Présentation des concepts relatifs au mapping objet relationnel.

- Qu'est-ce que le mapping Objet / Relationnel
- Les API de persistance en Java

Présentation d'hibernate

Présentation d'Hibernate et de sa mise en place.

- Présentation d'Hibernate
- La mise en place d'Hibernate dans une application
- Exemple d'utilisation

Les fichiers de mapping

Présentation des fichiers de mapping (hbm) et de leur configuration.

- Les fichiers de mapping Hibernate (hbm)
- Mapping des associations
- Le mapping de composants
- Filtrer les données

Le mapping de collection

Présentation de la mise en place du mapping de collections.

- Les collections persistantes
- Le mapping de collections

Utilisation de SQL natif

Présentation de l'utilisation de requêtes SQL avec Hibernate.

- Les SQLQuery
- Les requêtes SQL nommées
- SQL personnalisé

Le langage HQL

Présentation du langage de requêtage d'Hibernate (HQL).

- Le langage HQL
- Les clauses (from, select ...)
- Associations et jointures
- Les sous-requêtes

Les intercepteurs et les événements

Présentation des intercepteurs et événements d'Hibernate.

- Les intercepteurs
- Le système d'événements
- La sécurité déclarative

Les annotations d'Hibernate

Présentation des annotations Hibernate et exemples d'utilisation.

- JPA
- Exemple d'utilisation

Les outils

Présentation des outils permettant de simplifier l'utilisation d'Hibernate.

- Personnalisation du schéma
- Utilisation d'outils
- Utilisation de Ant

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Essentiel Web Java, Servlets et JSP

Code
CP-WJSP
Durée
1 jour
Tarif
690 € HT
Sessions
sur demande

Description

Le développement d'applications Web dynamiques que ce soit pour Internet ou pour des Intranet a pris une place importante dans le monde de l'informatique. Cette formation permettra de découvrir les mécanismes de base servlet/JSP.

Objectifs

- Comprendre les concepts présentés dans les différents modules de la formation
- Pouvoir effectuer des choix entre les différentes techniques présentées
- Réussir à les mettre en oeuvre de façon autonome lorsque le module est accompagné de travaux pratiques

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà bien Java et souhaitant assimiler les bases des applications Web JavaEE.

Pré-requis

- Maîtrise du langage Java et d'un IDE de type Eclipse nécessaires
- La connaîssance du langage HTML est recommandée

Programme

Rappels HTTP et application Web
Rappels sur les notions de base du Web et la structuration normalisée des applications Web Java.
- Rappel sur HTTP
- La structure normalisée d'une URL
- La structure d'une application Web

Les pages JSP
Présentation des principes de base associés aux pages JSP.
- Structure d'une page JSP
- Syntaxe des pages JSP
- Le cycle de vie d'une page JSP
- scripts, directives, expressions et actions
- Les objets pré définis
- La gestion des erreurs

L'API Servlet
Présentation des principes de base associés aux servlets.
- Structure d'une servlet
- Le cycle de vie d'une servlet
- Points-clé de l'API

L'utilisation de sessions
Présentation de la gestion de sessions utilisateurs dans les applications.
- Pourquoi utiliser des sessions ?
- Principe de fonctionnement des sessions
- Les méthodes de suivi des sessions
- Le cycle de vie des sessions
- Le modèle MVC
- Utilisation des JavaBeans

La sécurité des applications Web Java
En dehors de l'utilisation de protocoles sécurisés, il est important de regarder comment réduire l'accès aux différentes parties d'une application Web.
- La sécurité déclarative
- Utilisateur, rôle et realm
- Sécurité par programmation

JavaServer Pages Standard Tag Library
Présentation de la librairie de balises ansi que du langage unified EL.
- Le langage Unified Expression Language
- Les différentes catégories de balises

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Développement JavaScript avec AngularJS

Code
CP-ANG
Durée
4 jours
Tarif
1990 € HT
Sessions
sur demande

Description

AngularJS a été créé pour développer des applications Web professionnelles, maintenables et portables. Cette formation permet d'appréhender les facettes du développement d'une application complète.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà le langage HTML et souhaitant apprendre à mettre en place des applications Web utilisant le langage JavaScript.

Pré-requis

- Connaître le langage HTML et les technologies du web
- Connaître les bases de Javascript

Programme

Les grandes lignes d'AngularJS

Présentation d'AngularJS.
- Data binding bidirectionnel
- Séparation de la logique et des vues, contrôleurs et contextes
- Pattern MVVM : Model View ViewModel
- AngularJS et HTML5

Contrôleurs et contextes

Gestion des portées ...
- Fonctionnement des contrôleurs
- Modèle et contexte, hiérarchie des contextes
- Création d'un contexte enfant par certaines directives
- Evénements utilisateur : $scope.emit(), $scope.broadcast(), $scope.on()

Les vues, moteur de template et binding d'AngularJS

Les templates de vues.
- Utilisation des contrôleurs dans les vues
- Principales directives pour l'affichage, les champs de saisie, les répétitions, les actions utilisa
- Langage d'expressions, syntaxe, appel de méthodes et utilisation de filtres
- Inclusion de templates, et templates "inline"

Debug d'AngularJS

Comment gérer la mise au point
- Méthodes utiles de l'objet angular.
- Utilisation de Batarang
- Interception des exceptions avec le service $exceptionHandler

Les tests automatisés

Comment gérer le test unitaire
- Tests unitaires avec Jasmine et Testacular
- Tester les contrôleurs, tester les services
- Injection des dépendances et mock objects
- Service $httpBackend du module ngMock pour les tests unitaires
- Tests fonctionnels (End-to-End) avec "Angular Scenario Runner"

Le fonctionnement interne d'AngularJS

Les fonctionnalités sous jacentes.
- Le DOM comme template : compilation en deux phases des templates, intérêt des directives ng-src, ng-bind-template et ng-cloak
- Rafraîchissement des templates : les "watchers" créés par les directives de binding ou par $scope. $watch(...), boucle $digest et liste des $watch, intérêt de l'approche "dirty checking" d'AngularJS
- Cadre d'exécution d'AngularJS : exécuter du code dans le cadre d'AngularJS depuis l'extérieur avec $apply(...)

Application mono-page multi-vues, et routage

Paramètrage pour une page.
- Configuration du routage avec $routeProvider
- Paramètres de l'URL
- $route et historique du navigateur
- Avantages et inconvénients des "hash-bang URIs"
- Effectuer des traitements avant le chargement d'une vue

Les formulaires

Formulaire et contrôles.
- Eléments de formulaire
- Validation des données du formulaire

Echanges avec le serveur

Echanges de données.
- Service $http, et intercepteurs de réponse
- Service $resource pour les API de type "REST"
- Utilisation de "Server-Sent Events" ou de "WebSocket" avec AngularJS

L'API de "promise"

Définition de template.
- Fonctionnement des promises
- Utilisation des promises dans le binding des templates
- Asynchronisme et enchaînement de tâches avec des promises
- Utilisation des promises dans le service $http

Application et modules

Modularité.
- Création d'un module
- Configuration d'un module
- Démarrage automatique ou manuel d'une application
- Utilisation des modules pour partager des services entre applications

Services et injection des dépendances

Principe d'injection de donnée.
- Les différentes méthodes pour créer un service : factory(), ...
- Injection des services dans les contrôleurs et dans d'autres services
- Injection des dépendances et minimisation du code JavaScript
- Configuration des services, $injector

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Développement Web avec HTML5

Code
CP-HTML5
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Lorsque l'on veut mettre au point une application Web, on va devoir mettre au point l'affichage que l'on souhaite en utilisant le langage HTML. Le langage a connu une forte évolution à l'occasion lors de la sortie de la version 5 du langage.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà le langage HTML et souhaitant apprendre à mettre en place des applications Web utilisant HTML5.

Pré-requis

- Connaître un langage informatique

Programme

Le langage HTML
Présentation des origines du langage HTML et du Web.
- Le Web et ses origine
- Le W3C
- Le langage HTML

Evolution du langage HTML5 par rapport à HTML 4
Présentation des évolutions du langage HTML.
- Les balises qui disparaissent dans la version 5
- Les nouvelles balises de HTML5
- L'état sur les navigateurs compatibles
- Exemple d'utilisation

Le langage JavaScript
Présentation de notions de bases du langage Javascript.
- Le langage JavaScript
- L'accès aux éléments d'une page via JavaScript

Les Canvas (dessiner sur une page HTML)
Présentation des Canvas et exemple d'utilisation.
- Présentation des Canvas
- Utilisation de JavaScript pour dessiner dans un Canvas

Le Web Storage
Présentation du Web Storage pour stocker des données du coté client.
- Qu'est-ce que le Web Storage !
- Le stockage local et le stockage de session

L'tilisation des formulaires en HTML5
Présentation des formulaires en HTML5 avec les nouveaux input existants.
- Rappel sur le rôle des formulaires
- Les nouveaux inputs
- Exemple d'utilisation

Le multimédia avec HTML5
Présentation de l'utilisation de vidéos et fichiers audio avec HTML5.
- L'utilisation de fichiers vidéos
- L'utilisation de fichiers audio

Présentation des bases de ruby
Présentation des bases du langage ruby.
- Le langage ruby
- Notions de base

Utilisation de ruby avec HTML5
Présentation de l'utilisation de ruby avec HTML5.
- La balise ruby
- Exemple d'utilisation

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Mise en forme HTML avec CSS

Code
CP-CSS
Durée
2 jours
Tarif
1190 € HT
Sessions
sur demande

Description

Cascading Style Sheets (CSS) est un standard W3C permettant de décrire la présentation des documents HTML et XML. Cette formation permet d'en appréhender les bases.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà le langage HTML et souhaitant apprendre la mise en forme en utilisant CSS.

Pré-requis

- Connaître le langage HTML

Programme

Le langage CSS
Présentation des origines de CSS.
- Les origines de CSS
- Le W3C
- Le langage CSS

La syntaxe de base de CSS
Présentation de la syntaxe de base de CSS.
- Définition de l'utilisation de CSS dans une page HTML
- La notion d'id et de class
- Utilisation de CSS inline
- Exemples d'utilisation

Ajouter du style aux composants d'une page
Présentation de la modification des éléments HTML de base avec CSS.
- Modification du style d'un texte et de sa fonte
- Modification du style des liens hypertexte
- Modification du style des listes et des tables

Le modèle en boites de CSS
Présentation et utilisation du modèle en boite de CSS.
- Présentation du modèle en boites
- La gestion des bordures
- Mise en place de marges

Le positionnement et le dimensionnement des éléments
Présentation des méthodes de dimensionnement et de positionnement des éléments HTML.
- Positionnement statique, fixe ...
- Dimensionner les éléments

L'utilisation des images
Présentation de la gestion de la modification d'affichage des images.
- Gestion de l'opacité
- Les sprites d'image
- Exemple d'utilisation

Le gestion des médias en CSS
Présentation de l'utilisation des différents média connus par CSS.
- Présentation des différents styles existants
- Utilisation du média screen
- Utilisation du média print

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Développement JavaScript

Code
CP-JS1
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Une application Web n'utilisant que le langage HTML à comme principal inconvénient d'être statique. Aujourd'hui les utilisateurs s'attendent à se trouver face à une application réagissant en fonction d'une action. Pour cela, on va souvent utiliser le langage JavaScript.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà le langage HTML et souhaitant apprendre à mettre en place des applications Web utilisant le langage JavaScript.

Pré-requis

- Connaître le langage HTML
- Connaître un langage informatique

Programme

Le langage Javascript
Présentation des origines du langage JavaScript.
- Les origines du langage JavaScript
- L'ECMA
- L'utilisation du langage JavaScript

Eléments de base du langage JavaScript
Présentation de la déclaration de variables, les boucles ...
- Insertion de JavaScript dans une page HTML
- Déclaration de variables
- Les boucles
- Les structures conditionnelles

JavaScript et POO
Présentation de la programmation orientée objet en JavaScript.
- Les objets à disposition en JavaScript
- L'Objet String
- L'Objet Date
- La création de tableau

Interaction avec le navigateur
Présentation des possibilités d'interaction avec le navigateur en utilisant le langage Javascript.
- L'objet 'navigator'
- L'objet 'window'

Interagir avec les balises HTML en utilisant JavaScript
Accès et modification des éléments HTML de la page.
- La gestion des formulaires
- Utilisation de JavaScript pour dessiner dans un Canvas

La gestion des événements avec JavaScript
Présentation de la gestion des événements avec JavaScript.
- Les événements disponibles
- Exemples d'utilisation des événements

Accès à l'arbre DOM d'un document avec JavaScript
Accès à l'arbre DOM d'un document pour en modifier le comportement.
- Le HTML dynamique
- Les Objets DOM

Utilisation de CSS avec JavaScript
Modifier les éléments de style en utilisant JavaScript.
- Rappel de CSS
- Modification du style de balises

Introduction à AJAX
Présentation d'Asynchronous JavScript And XML.
- Qu'est-ce qu'AJAX
- Notions de base d'AJAX

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

--> Sur demande

Développement d'applications Web 2.0 avec JQuery

Code
CP-JQY
Durée
3 jours
Tarif
1490 € HT
Sessions
sur demande

Description

Cette formation présente les bases de la programmation Web avec JQuery.

Participants

Ce cours s'adresse aux développeurs, architectes ou chefs de projet, connaissant déjà le langage HTML et souhaitant apprendre à mettre en place des applications Web avec JQuery.

Pré-requis

- Connaître le langage HTML et les fondamentaux du Web
- Connaître les bases de JavaScript

Programme

Introduction
Présentation de JQuery.
- Les apports de JQuery par rapport à JavaScript
- Les constituants JQuery (éléments, JQuery UI, ...)
- La mise en place d'un environnement de développement

Méthodes utilitaires et sélecteurs
Présentation des méthodes utilitaires et des sélecteurs JQuery.
- La manipulation d'objets ($.each(), $.extend(), ...)
- Le traitement des tableaux ($.map()) et des chaines ($.trim()).
- Le rôle des sélecteurs JQuery
- Les pseudo-classes et les combinateurs

Arbre DOM
Accès à l'arbre DOM.
- Rappels sur DOM
- L'accès aux éléments d'une page
- Insertion et suppression d'éléments
- La gestion des classes CSS

Ajax avec JQuery
Mise en oeuvre des concepts Ajax.
- Gestion des requêtes serveur et $.ajax()
- Récupération des réponses
- Transmission de paramètres

Interface utilisateur riche
Panorama des différents éléments d'IHM.
- Utilisation et définition d'effets visuels
- Gestion des événements et emploi de bind()
- Utilisation de JQuery UI
- Les composants (onglets, dialogues, boutons, barres, ...)
- Gérer la sélection, le glisser-déposer, le redimensionnement

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.

Calendrier des sessions

Formation	Jan 19	Fev 19	Mar 19	Avr 19	Mai 19	Juin 19
CP-ALGO						
CP-AND1			13/03 au 15/03			
CP-ANDK					20/05 au 24/05	
CP-ANDK2					27/05 au 29/05	
CP-ANG						
CP-BPMN			11/03 au 12/03			
CP-C						
CP-CAML			28/03 au 29/03			11/06 au 12/06
CP-CPP1						12/06 au 14/06
CP-CPP1X			20/03 au 22/03			
CP-CPP2						
CP-CS01						
CP-CSS						
CP-DOCK		07/02	28/03		30/05	
CP-EJB3						
CP-GIT		06/02	27/03		29/05	
CP-HIB						
CP-HTML5						
CP-IOTL			25/03 au 29/03			03/06 au 07/06
CP-IOTP						03/06 au 07/06
CP-IPH1						
CP-JAV8		05/02	26/03		28/05	
CP-JAVA1				24/04 au 26/04		
CP-JAVA7						
CP-JEE6						
CP-JMAJ						
CP-JPA						
CP-JQY						
CP-JS1						
CP-JXML						
CP-LWAN		04/02	25/03		27/05	
CP-MAV				01/04 au 02/04		
CP-MSVC			13/03 au 15/03			17/06 au 19/06
CP-NIFI			28/03 au 29/03			11/06 au 12/06
CP-OBJ1						
CP-OBJT						
CP-PATT						
CP-PYTH			20/03 au 22/03			
CP-QT1						
CP-REST			20/03 au 22/03		15/05 au 17/05	
CP-SCA			20/03 au 22/03			
CP-SOAP				02/04 au 04/04		
CP-SPB						17/06 au 19/06
CP-SPK					15/05 au 17/05	
CP-SPRING			27/03 au 29/03			
CP-STR2						
CP-TEST						
CP-UML1					22/05 au 24/05	
CP-WJSF						
CP-WJSP						
S-IOT						24/06

© Leuville Objects - 3 rue de la Porte de Buc, 78000 Versailles FRANCE. Tous droits réservés.