



Formation Perfectionnement Delphi

L'Institut Pascal - formation@formation-delphi.com - Tél 01.42.83.69.36

Objectifs	maîtriser de techniques avancées Delphi: création de composants, bases de données multi niveau, les threads, la programmation socket, XML et les services Web
Durée	5 jours
Versions Delphi	Delphi 5 et 6, Delphi 2006 à Delphi 2010, Delphi XE
Pré Requis	expérience du développement Delphi d' au moins trois mois
Public	développeur et analyste, dans le cadre du plan de formation d'entreprise , ou d'un DIF

Programme Détaillé - Formation Perfectionnement Delphi

1 Programmation Objet Delphi

- les **UNITÉs**:
INTERFACE, **IMPLEMENTATION**, **USES**, appels croisés, **INITIALIZATION**, **FINALISATION**, ordre d'appel - type abstrait / notion de contrat
- **Classes et Objets**:
définition de types **CLASS** - attributs et méthodes - type définit en global - accès direct aux attributs - **SELF** - utilisation des objets: - allocation par : **CONSTRUCTOR** - valeurs initiales - **DESTRUCTOR** - méthodes: - surcharge des paramètres et **OVERLOAD** - méthode de **CLASSE** - les attributs **PRIVATE** et **PUBLIC** - affectation et copie - *Assign* - déclaration anticipée de **CLASS** - structures
- **L'Héritage**
le concept d'héritage - appeler une méthode de l'ancêtre: **INHERITED** - visibilité: la directive **PROTECTED** - compatibilité d'affectation - test du type - **IS**, **AS**, **WITH**, surcharge
- **Le Polymorphisme: VIRTUAL**
VIRTUAL - descendant: surcharge par **OVERRIDE** (statique ou **VIRTUAL** masquent) - **REINTRODUCE** de méthodes **VIRTUAL** - **OVERLOAD** - appel d'**INHERITED** (en début) - répartition par **DYNAMIC** - schéma mémoire - Vmt ou Dmt - **DESTRUCTOR** et **OVERRIDE** - méthode et classes abstraites - **CONSTRUCTOR** et **VIRTUAL** - référence de classe: **CLASS OF**
- **Structures objet**
structures de base: pile, liste, arbre - listes, collections - types génériques
- les flux *tStream* et ses descendants
- **Delphi et les Classes**
le modèle par délégation - *tForm*: création statique et dynamique - héritage de formes - les *tFrame*
- **Les Exceptions**
le bloc **TRY EXCEPT** - le bloc **TRY FINALLY**
- **Les INTERFACES**

2 Ecriture de Composants Delphi

- **Création d'un Premier Composant**
éditeur de paquets - procédure **REGISTER** - installation des composants
- **les propriétés**
Les différents types de propriétés : simple, énumérée, ensemble, objet - Changer le niveau de visibilité d'une propriété - Sauvegarde des propriétés dans le fichier DFM - Les propriétés de type tableau - Indice pré codés - Les mots clés **Default**, **noDefault**, **Stored** et **Index**

- Les événements

- principe de fonctionnement - délégation - Les types standard (*tNotifyEvent*, ...) - surcharge d'un événement
- **package**
packages et DLL - package en exécution et en mode conception - directives de compilation
- **interaction et connexion**
mode conception et exécution: *tComponent.ComponentState* - le test *TComponent.Loaded* - les méthodes *Notification* et *FreeNotification*
- **types de composants**
la structure de la VCL - les différents types de composant: graphiques, Windows, dépendant des données, spécifiques
- **Création de composants graphiques**
encapsulation de la GDI par Delphi - *TGraphicControl*
- **composants Windows**
- **Composants dépendants des données**
architecture *tDataset*, *tDataSource*, *tDataLink*, composant visuel - mise à jour et modification - création d'un objet *tDataLink*, événement *OnDataChange*
- **éditeur de propriété**
- **éditeur de composant**
- **fichier d'aide**

3 Bases de Données Multi Tiers

- **Architecture Multi Tiers**
architecture Client Serveur - architecture multi tiers - les composants *dbExpress* et *DataSnap*
- **Le tClientDataset et le mode nomade**
tDataSet en mémoire - création ou chargement depuis un autre *tDataSet* - sauvegarde disque en mode binaire ou XML - indexes - agrégats - champs calculés et *InternalCalc* - clonage du curseur - modifications: - maître détail - tables emboîtées (nested)
- **Connexion au Moteur Sql - dbExpress**
la nouvelle architecture *dbX4* - fichier .INI - lecture et modification de données avec les composants *dbExpress*
- **Le Serveur d'Application**
le composant de connexion - les composants d'accès aux données *tSqlQuery* - le *tDataSetProvider*
- **Connexion aux Clients**
les types de connexion: locale, DCOM, HTTP, Soap - *Remote Data Modules* - pooling et type d'instances
- **Gestion du Client**
contrôle de la taille des paquets *PacketRecords*, *FetchOnDemand*, et *OwnerData* - utilisation de contraintes importées du Serveur et sur mesure



Formation Perfectionnement Delphi

L'Institut Pascal - formation@formation-delphi.com - Tél 01.42.83.69.36

3 Bases de Données Multi Tiers

- Gestion des Modifications

paramétrage du *tDataSetProvider* - *ProviderFlags*, *UpdateMode* - *tClientDataSet.ApplyUpdates* - *CurValue*, *NewValue* et *OldValue* - événements *BeforeUpdateRecord* - gestion des erreurs de réconciliation. *OnReconcileError* et *OnUpdateError*. Les paramètres *UpdateKind* et *Action*

- Relations Maître Détail

4 Threads Delphi

- Principe des Threads

les processus Windows

- Utilisation Simple

la classe *tThread* - les données propres au thread

- Communication entre les Threads

Synchronize - communication par *SendMessage* et *PostMessage*

- Accès aux données et synchronisation

accès aux globales - mécanismes de synchronisation: section critique, mutex, sémaphore, événement

5 Programmation Sockets Delphi

- Programmation Socket

Adresses et Port - les primitives WinSock - Serveur: *Socket*, *Select*, *Bind*, *Accept*, *Send*, *Recv*, *Close* - Client: *Socket*, *Select*, *Connect*, *Send*, *Recv*, *Close* - fonctionnement de base du Serveur et de Clients. Gestion des clients - mode bloquant et mode asynchrone - UDP et TCP

- Composants Delphi

tServerSocket et *tClientSocket* Delphi - Indy - utilisation de UDP - architecture de base : *tIdIoHandler*, *tIdConnectionIntercept*, *tIdTcpConnection* - gestion des threads - mode bloquant et *tIdAntiFreeze*

- Suite TCP / IP

les protocoles de la suite TCP / IP - utilisation de HTTP, Mail, FTP

- Ecriture de protocoles personnalisés

- Programmation Internet

CGI et ISAPI

6 Xml

- Perspectives XML

échange de données avec XML: bases de données, Services Web, blogs et RSS, Ajax et pages Internet - Structure d'un document XML

- DOM (Document Object Model)

utilisation de *tXmlDocument*: analyse, création, modification de document - architecture *iDomDocument* et *tXmlDocument* - Extraction de données: *tXMLReader* - Construction de documents XML en partant de zéro - Création de noeuds d'élément et de texte - *tXMLWriter*

6 Xml

- Validation de données XML

schémas XML (.XSD) pour valider la structure et le contenu XML - Syntaxe et composants d'un schéma XML - Ecrire des schémas pour restreindre le contenu XML - Définition des éléments et attributs valides - Intégration des règles métier dans les schémas XML - l'outil *XmlMapper* pour générer un DTD ou un XSD

- Navigation dans un document

navigation : XPath - Interrogation d'un document : XQuery

- Transformation de documents avec XSL

Conversion de documents XML - le composant *tXmlTransform* - fichiers .XTR - le composant *tXmlTransformProvider*

- XML et bases de données

dbExpress et .XML - sauvegarde *tClientDataset* et paquets .XML - conversion d'autres *tDataSets* en XML - fournitures de données .XML par un *Serveur DataSnap*

7 Services Web

- Le besoin de services Web

communications Internet B2B, sans intervention utilisateur - Principales technologies : XML, SOAP, WSDL

- Création du premier service Web

architecture d'une application de Services Web : le Serveur et ses méthodes, le Client et ses requêtes - écriture du Serveur, en utilisant le *Wizard* - écriture du Client, création de l'UNITÉ d'importation, appel des méthodes du Serveur

- le protocole SOAP

Structure du message SOAP - Enveloppe, en-tête et corps du message - Gestion des types XML simples - Types complexes - Tableaux - Sérialisation des objets Delphi

- le Serveur de services web

fonctionnement du Serveur: - création et enregistrement d'une INTERFACE - ventilation par *tHttpSoapDispatcher* - appel par *tHttpSoapPascalInvoker* - la publication au format WSDL - mise en oeuvre du Serveur - utilisation de Web Application Debugger - utilisation de CGI et ISAPI - transformation de serveurs Web Application Debugger en CGI ou ISAPI - Déploiement sous IIS

- le Client Web Service

le composant *tHttpRio*

- Services Web et bases de données

Encodage et envoi de données binaires - envoi de données .XML - *tXmlBroker* - utilisation de *tSoapAttachment* - communication avec *dbExpress* et *DataSnap*