



Apprendre la programmation avec Delphi

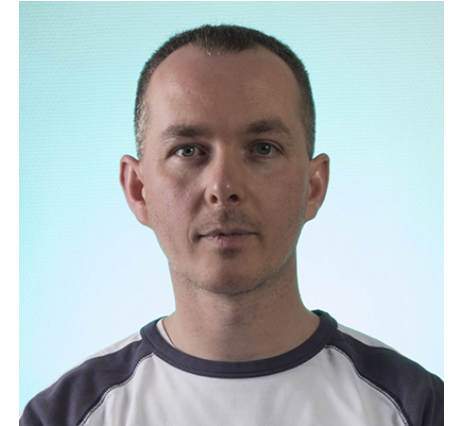
30 mars 2020

#DelphiBootCamp

Préambule



Patrick Prémartin

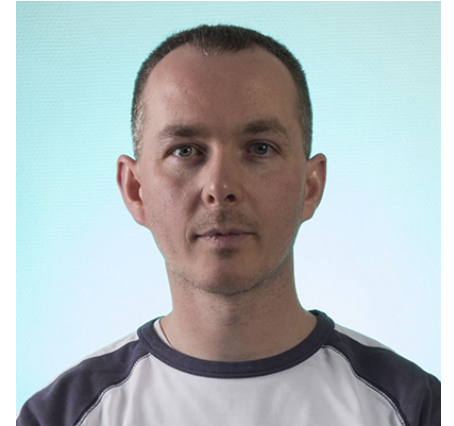


- MVP Embarcadero, prestataire informatique et formateur Delphi
- Entreprise Olf Software
(conseil, développement et assistance technique) :
<https://olfsoftware.fr>
- Formations Delphi en entreprise ou en groupe :
<https://se-former-a-delphi.fr>
- Blog sur l'actualité du développement en Pascal avec Delphi :
<https://developpeur-pascal.fr>



Pour contacter Patrick Prémartin

- Le blog :
<https://developpeur-pascal.fr/contact/>
- Twitter :
<https://twitter.com/premartinpatric>
- LinkedIn :
<https://www.linkedin.com/in/patrickpremartin/>
- Par téléphone uniquement sur rendez-vous ou (vraies) urgences de clients existants.



Fonctionnement du bootcamp

- Ce bootcamp a pour objectif de vous montrer comment faire de la programmation de logiciels et applications mobiles en utilisant le Pascal et Delphi.
- Je vous donne rendez-vous en direct du lundi au vendredi entre 10 heures et midi (heure française) pour vous faire découvrir le développement logiciel.
- Je répondrai à vos questions en fin de chaque chapitre.



Fonctionnement du bootcamp

- Pour les absents et la postérité ce bootcamp est enregistré.
- Vous retrouverez les rediffusions accompagnées des PDF de présentation et des sources des exemples sur le site <https://apprendre-delphi.fr>
- Les vidéos seront également publiées sur la chaîne YouTube de Barnsten.



Fonctionnement du bootcamp

- Pour discuter entre vous, partager vos impressions et votre progression sur les réseaux sociaux, vous pouvez utiliser #DelphiBootCamp
- Je répondrai aux questions posées avec ce hashtag sur Twitter, Facebook et LinkedIn durant toute la durée du bootcamp.
- Famille, amis et collègues sont les bienvenus s'ils sont intéressés. Les inscriptions restent possibles depuis <https://apprendre-delphi.fr>



Qui est concerné par ce bootcamp ?

- Aucun prérequis n'est nécessaire pour suivre cette formation (à part de savoir se servir d'un ordinateur et d'avoir Windows pour faire les exercices).
- Ce bootcamp est progressif et intégralement basé sur des exemples. Je vais alterner entre cette présentation et mon écran. Vous pourrez travailler de chez vous et me poser vos questions le lendemain.



Qui est concerné par ce bootcamp ?

- Les non informaticiens sont les bienvenus.
- Ils pourront se familiariser avec
 - les concepts de base de la programmation (algorithmie et bonnes pratiques)
 - un langage de développement simple à apprendre (Pascal et Object Pascal)
 - un environnement de développement complet (Delphi)



Qui est concerné par ce bootcamp ?

- Les développeurs débutants ou chevronnés qui ne connaissent pas le Pascal ou Delphi auront aussi un intérêt à suivre ce bootcamp car ils y découvriront un ensemble d'outils permettant de développer des programmes variés pour les 5 systèmes d'exploitation les plus courants.



Qui est concerné par ce bootcamp ?

- Les développeurs connaissant déjà Delphi ne devraient pas apprendre grand chose.
- Je ne rentrerai pas dans le détail du fonctionnement des classes, des interfaces, des API système et des subtilités de plateformes dans FMX.
- Si vous désirez approfondir certains sujets ce n'est clairement pas le bon endroit mais nous pouvons en discuter en dehors pour une formation spécifique.



Où trouver des infos ?



Liens utiles

- Rediffusion des sessions de cette formation :
<https://vasur.fr/delphibootcamp>
- Sources des exemples et corrigés de cette formation :
<https://github.com/DeveloppeurPascal/ApprendreLaProgrammationAvecDelphi>
- Blog sur le Pascal et Delphi (actualité, trucs et astuces) :
<https://www.developpeur-pascal.fr>



Liens utiles

- Formations individuelles chez soi :
<https://apprendre-delphi.fr>
=> pour le moment juste celle que vous suivez en ce moment
- Formations en entreprise ou en groupe :
<https://se-former-a-delphi.fr>
=> de nouvelles dates et programmes dès qu'on peut
=> des formations personnalisées sur demande, en ligne ou en présentiel



Liens utiles

- Embarcadero :
<https://www.embarcadero.com/fr/>
- Documentation en ligne « officielle » :
<http://docwiki.embarcadero.com>
- Actualité, blogs, forums « officiels » (en anglais) :
<https://community.idera.com/developer-tools/b/blog>



Liens utiles

- Embarcadero sur Twitter :
<https://twitter.com/EmbarcaderoTech>
- Embarcadero sur Facebook :
<https://www.facebook.com/embarcaderotech/>
- Embarcadero sur YouTube :
<https://www.youtube.com/user/EmbarcaderoTechNet/featured>



Liens utiles

- Learn Delphi :
<https://www.learndelphi.org>
- Embarcadero Academy :
<https://www.embarcaderoacademy.com>
- Portail qualité :
<https://quality.embarcadero.com/>
=> signaler et consulter les anomalies recensées
=> faire des demandes de nouvelles fonctionnalités



Liens utiles

- Barnsten :
<https://www.barnsten.com/fr/>
- Barnsten sur Twitter :
<https://twitter.com/BarnstenFR>
- Barnsten sur Facebook :
<https://www.facebook.com/EmbarcaderoFrance/>



Liens utiles

- Barnsten sur YouTube :
<https://www.youtube.com/channel/UCSrWP5Y01uSHS3014Pfprlw>



Liens utiles

- Agrégateur d'infos et de blogs liés au Pascal et Delphi :
<http://www.delphimagazine.com>
- Blaise Pascal Magazine (en anglais) :
<https://www.blaisepascalmagazine.eu>
=> magazine disponible en ligne dédié à la programmation en Pascal.
- Programmez! (en français) :
<https://www.programmez.com>
=> magazine mensuel disponible en kiosques et en ligne dédié au développement logiciel dans tous les langages avec des articles ponctuelles sur Delphi et la programmation en Pascal.



Liens utiles

- Webinaires mensuels en français sur une thématique précise :
<https://developpeur-pascal.fr/p/6007-webinaires.html>
- Raccourcis au clavier dans Delphi :
<https://developpeur-pascal.fr/p/005-les-combinaisons-de-touches-et-raccourcis-clavier-utiles-dans-l-edition-de-delphi.html>
- D'autres sources d'infos et ressources utiles :
<https://developpeur-pascal.fr/c/4-ressources.html>



Dans les épisodes précédents...



Jour 01 - Session du 23 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - La présentation de ce bootcamp.
 - La présentation de Delphi.
 - Les différentes licences de Delphi.
 - Comment et où télécharger Delphi gratuitement et légalement.
 - L'environnement de développement intégré (EDI ou IDE).
 - Un aperçu d'un projet VCL pour Windows.
- En guise d'exercice vous deviez installer Delphi pour pouvoir suivre les sessions suivantes de cette formation.



Jour 02 - Session du 24 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - Des définitions en lien avec Delphi, le Pascal et la programmation
 - La création d'un projet VCL
 - Le concepteur de fiches VCL et ses nombreux outils
- En guise d'exercice vous deviez reproduire des écrans de logiciels existants pour manipuler les différents outils du concepteur de fiche d'un projet VCL.



Jour 03 - Session du 25 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - Concepteur de fiche VCL
 - Propriétés et événements de composants
 - Bases du Pascal
 - Bases d'algorithmie
 - Exemples de programmes
- En guise d'exercice vous deviez faire plusieurs programmes en Pascal dont reproduire ceux faits lors de notre session.



Jour 04 - Session du 26 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - Concepteur de fiche VCL
 - Manipulation des fiches dans un programme
 - Propriétés des fiches pour leur affichage
 - Les unités et leur visibilité
 - Exemples de programmes
- En guise d'exercice vous deviez jouer avec les fiches et accéder à des propriétés de fiches depuis d'autres, les afficher, les masquer, les afficher en modal ou non.



Jour 05 - Session du 27 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - La gestion des exceptions
 - Manipulation des fiches dans un programme
 - Les modules de données
 - Exemples de programmes
- Deux exercices à faire pour la session suivante : préparer les écrans d'une gestion de contacts (avec plusieurs fiches) et ceux d'une gestion de liste de courses (avec des onglets).



Les modules de données



Les modules de données

- Comme nous l'avons vu vendredi les modules de données sont des fiches permettant de stocker des composants non visuels.
- Grâce aux modules de données on peut centraliser des composants accessibles depuis d'autres fiches.
- En pratique, voici comment s'en servir.



Les listes d'images



Les listes d'images

- Nous avons vu le TImageList lors des premières sessions pour stocker des images utilisées par exemples dans les boutons.
- Ce composant est assez ancien et ne gère qu'une seule taille d'image ce qui peut générer des anomalies à l'affichage sur des écrans récents en High-DPI ou 4K.
- Depuis la version 10.3 Rio de Delphi nous avons deux nouveaux composants pour traiter ce problème.



Les listes d'images

- Le composant TImageCollection va centraliser les images.
- Il doit être placé dans un module de données.
- On y met les images que l'on veut utiliser dans l'application et on peut fournir plusieurs versions de tailles différentes pour chaque image.



Les listes d'images

- Le composant TVirtualImageList se place sur les fiches ayant besoin d'afficher des images provenant de la collection.
- Les fiches doivent avoir accès au module de données.
- La liste d'images virtuelles pointe sur la collection d'images.
- Delphi prendra l'image la plus proche de la taille physique d'affichage en piochant dans les tailles fournies.



L'instruction « if »



L'instruction « if »

- « if » sert à faire quelque chose en fonction de certains cas.
- Deux constructions sont possibles :
 - If (condition) then instruction;
 - If (condition) then instruction else instruction;
- « condition » est un booléen ou un ensemble de tests.
- « instruction » est une ligne de code ou un bloc.



L'instruction « if »

- Rappelez-vous : un booléen est un type de données pouvant prendre les valeurs « True » (vrai) et « False » (faux).
- Un booléen peut aussi avoir pour valeur une série de tests comme nous l'avons fait la semaine dernière dans les exemples vérifiant si les éléments d'interface ne dépassaient pas de leurs limites d'affichage ou pour tester si la valeur saisie par l'utilisateur était celle définie aléatoirement.
- Les variables booléennes sont de type « boolean ».



L'instruction « if »

- Les comparaisons se font avec les symboles mathématiques classiques :
 - $A < B$
Si A est inférieur à B alors on obtient True, sinon False
 - $A \leq B$
Si A est inférieur ou égal à B alors on obtient True, sinon False



L'instruction « if »

- Les comparaisons se font avec les symboles mathématiques classiques :
 - $A > B$
Si A est supérieure à B alors on obtient True, sinon False
 - $A \geq B$
Si A est supérieure ou égal à B alors on obtient True, sinon False



L'instruction « if »

- Les comparaisons se font avec les symboles mathématiques classiques :
 - $A = B$
Si A est égal à B alors on obtient True, sinon False
 - $A \neq B$
Si A est différent à B alors on obtient True, sinon False



L'instruction « if »

- On a pour habitude de mettre les conditions entre parenthèses.
- En Pascal ce n'est pas une obligation mais ça améliore la lisibilité du code, sa maintenance et ça évite les erreurs d'interprétation lorsque plusieurs tests se suivent.



L'instruction « if »

- On peut imbriquer les « if » les uns dans les autres mais pour faire plusieurs tests en même temps il suffit de les « attacher » les uns aux autres avec ces mots clés:
 - « and » pour « et »
 - « or » pour « ou »
- Exemple :
 - Si A est inférieur à B et que X est supérieur à Y alors faire ceci sinon faire cela.
 - `If (A<B) and (X>Y) then ceci else cela;`



L'instruction « if »

- Il reste la négation d'une valeur booléenne. On l'obtient avec « not ».
- Si A n'est pas inférieur à B alors
devient
if not (A<B) then

Dans ce cas on peut bien entendu écrire aussi
if (A>=B) then
mais ce n'est pas toujours possible.



L'instruction « if »

- Pour des raisons de lisibilité évitez de faire des négations dans vos conditions. Utilisez les quand vous n'avez pas le choix.
- Nous l'avons vu la semaine dernière avec l'activation de l'horloge : on peut utiliser « not » pour inverser une valeur booléenne.



L'instruction « if »

- Ces lignes sont équivalentes :
- `Timer1.Enabled := not Timer1.Enabled;`
- `if Timer1.Enabled then
 Timer1.Enabled:=false
else
 Timer1.Enabled:=true;`
- Si c'était « True », on passe à False sinon on passe à True;



Stocker des données



Qu'est-ce qu'une donnée ?

- En généralisant, tout ce que manipule un programme est une donnée.
- On a des données en mémoire : les variables.
- On a des données sur support de stockage (clé USB, disque dur, CD, DVD, ...) : les fichiers.



Qu'est-ce qu'une variable ?

- Nous l'avons déjà vu :
 - Une variable est un nom que l'on utilise dans notre code source.
 - Une variable doit être déclarée avant d'être utilisée.
 - Une variable a un type qui permet de manipuler son contenu de la bonne façon.



Qu'est-ce qu'un fichier ?

- Pour le système d'exploitation, peu importe ce qu'il y a dedans, un fichier est un fichier.
- Un fichier est un ensemble d'octets regroupés dans une zone du support de stockage, accessibles par un chemin composé de dossiers et du nom du fichier.



Qu'est-ce qu'un fichier ?

- Pour le développeur, chaque fichier contient des données qui peuvent être libres ou structurées.
- Vus du Pascal, les fichiers sont de type texte ou binaire.
- Parmi les fichiers, certains permettent de stocker des données structurées accessibles par un ensemble de routines que l'on appelle un moteur de base de données.



Qu'est-ce qu'une donnée structurée ?

- Les données structurées sont des données qui ont un format de stockage fixe et connu.
- Ce sont par exemple les tables des bases de données ou les RECORD en Pascal (struct pour le C).
- On peut gérer tous ces formats en Pascal. Delphi propose les composants et librairies permettant d'accéder à la majorité des bases de données existantes.



Qu'est-ce qu'une base de données ?

- Une base de données est un ensemble d'éléments.
- Il en existe deux types : les bases SQL et les base NoSQL.
- Je simplifie au maximum, en réalité les bases de données sont très variées et on peut faire du SQL dans du NoSQL comme l'inverse selon le moteur utilisé.



Qu'est-ce qu'une base de données SQL ?

- Les bases SQL ont des tables.
- Les tables ont des enregistrements.
- Les enregistrements ont des champs.
- Les champs ont un type.
- Les champs sont les éléments de base de chaque table et décrivent les données stockées.



Qu'est-ce qu'une base de données SQL ?

- Pour accéder à une base de données SQL il faut :
 - Le moteur de base de données correspondant au format de stockage des données.
 - Connaître et utiliser le langage SQL adapté au moteur de base de données.
- SQL est un langage normalisé mais il existe des variantes selon la base utilisée.



Qu'est-ce qu'une base de données SQL ?

- Les commandes en SQL sont appelées des requêtes.
- Les requêtes SQL manipulent les enregistrements des tables d'une base.



Qu'est-ce qu'une base de données NoSQL ?

- Histoire de ne pas m'étendre sur la question qui est un sujet très vaste, les bases NoSQL sont des ensembles dans lesquels on stocke des informations dans différents formats.
- Elles ne sont en principe pas accessibles par des requêtes SQL.
- On peut voir un disque dur comme une base NoSQL : les fichiers représentent les données accessibles à partir d'un chemin, pas par rapport à leur contenu.



Conception de bases de données



Modèle conceptuel de données (MCD)

- Il existe plusieurs façon de concevoir une base de données. Je vous propose celle que j'utilise et qui me semble la plus simple.
- Pour la programmation on parle d'algorithme.
- Pour les bases de données on parlera de « MCD » ou « modèle conceptuel de données ».



Modèle conceptuel de données (MCD)

- Concevoir des bases de données est un métier différent de celui de développeur, mais tout développeur doit avoir des bases en conception de données.
- Une base SQL est un ensemble de tables. Elle utilise un moteur de données. Il en existe des tas : MySQL, MariaDB, SQL Server, Interbase, Oracle, SQLite, Paradox, dBase, Firebird, Sybase, ...
- Ces bases sont soit locales, soit client / serveur.



Modèle conceptuel de données (MCD)

- Pour les exemples de code et les exercices nous utiliserons le moteur de base de données locale SQLite qui est disponible sur tous les systèmes d'exploitation.
- C'est un projet open source.
- Le moteur de la base de données est contenu dans une DLL ou son équivalent selon le système d'exploitation.



Modèle conceptuel de données (MCD)

- Pour en revenir à la conception de la base, peu importe le moteur utilisé au final, seule compte la distinction entre SQL et NoSQL.
- Le MCD permet de définir des bases de données SQL relationnelles.
- Les tables peuvent être liées entre elles par une relation.
- Idéalement on prévoit une clé unique dans chaque table pour s'y référer dans les programmes ou avec les relations.



Modèle conceptuel de données (MCD)

- Prenons un exemple avec des livres, des auteurs et des éditeurs.
- Un livre est représenté par un GENCOD (son code à barres), un titre, un ou plusieurs auteurs, un éditeur, une couverture, un descriptif.
- Un auteur est représenté par un nom, un prénom, un pseudonyme.
- Un éditeur est représenté par un nom.



Modèle conceptuel de données (MCD)

- En écrivant ça, nous avons décrit les 3 tables principales de notre base de données.
- Le seul problème est de voir comment relier les livres à leurs auteurs et éditeurs. C'est à ça que servent les relations.
- C'est aussi là qu'interviennent les clés uniques. On utilise généralement une valeur numérique entière et on laisse le moteur de base la gérer quand il le peut.



Modèle conceptuel de données (MCD)

- Un livre a un éditeur. En revanche un éditeur peut avoir plusieurs livres.
- On va donc stocker l'identifiant de l'éditeur au niveau de chaque livre, puisqu'il ne sera pas possible d'ajouter X champs « livre » dans la table éditeurs.



Modèle conceptuel de données (MCD)

- Un livre peut avoir plusieurs auteurs. Chaque auteur peut avoir plusieurs livres.
- Comme il n'est pas possible, ni sur l'un, ni sur l'autre de stocker le bon nombre de champs pour faire le lien entre eux, on ajoute une table intermédiaire « `lien_auteur_livre` »
(le nom est libre, c'est juste pour la clarté de l'explication)



Modèle conceptuel de données (MCD)

- Dans cette table intermédiaire on peut stocker autant d'informations que l'on veut.
- On peut donc mettre plusieurs fois l'identifiant du livre attachés chacun à un auteur.
- On peut donc mettre plusieurs fois l'identifiant de l'auteur attachés chacun à un livre.



Modèle conceptuel de données (MCD)

- Pour créer nos tables dans une base de données SQL, il faut saisir les requêtes SQL CREATE correspondantes.
- On peut aussi se simplifier la vie avec un logiciel qui fera le boulot à notre place et permettra de gérer les évolutions de la structure de la base sur laquelle on travaille.



Modèle conceptuel de données (MCD)

- J'ai pris pour habitude d'utiliser TMS Data Modeler depuis quelques mois. Il est pratique et pas cher.

<https://www.tmssoftware.com/site/tmsdm.asp>

- Une licence pour Aqua Data Studio est fournie avec la licence Architect de Delphi et RAD Studio.

<https://www.aquafold.com/aquadatastudio>



Modèle conceptuel de données (MCD)

- HeidiSQL, développé en Delphi, est très complet et open source.
<https://www.heidisql.com>
- Il en existe de nombreux autres avec des fonctionnalités variées dans les logiciels libres comme dans les licences moins ouvertes. Le bouche à oreille et les tests vous permettront de trouver celui qui vous plaît le plus.



Exercice 4 : Gestionnaire de contacts

- Ce programme en VCL aura une fenêtre principale dans laquelle vous listerez les contacts enregistrés.
- Trois boutons seront disponibles sur cet écran : AJOUTER, MODIFIER et SUPPRIMER.
- Ces trois boutons ouvriront une nouvelle fenêtre avec les champs nécessaires, un bouton de validation et un bouton d'annulation.
- Vous stockerez les contacts dans un ensemble de données en mémoire et un fichier JSON sur le disque dur.



Exercice 5 : Liste de courses

- Ce programme aura une fenêtre principale dans laquelle vous listerez votre liste de courses.
- Trois boutons seront disponibles sur cet écran : AJOUTER, MODIFIER et SUPPRIMER.
- Plutôt que de passer par une nouvelle fenêtre, gérez les modifications dans les pages d'un classeur à onglets.
- Vous stockerez la liste dans un ensemble de données en mémoire et un fichier JSON sur le disque dur.



A suivre...

La rediffusion de cette session sera disponible
sur <https://vasur.fr/delphibootcamp>

Discutez sur les réseaux sociaux avec le hashtag
#DelphiBootCamp

