



Apprendre la programmation avec Delphi

26 mars 2020

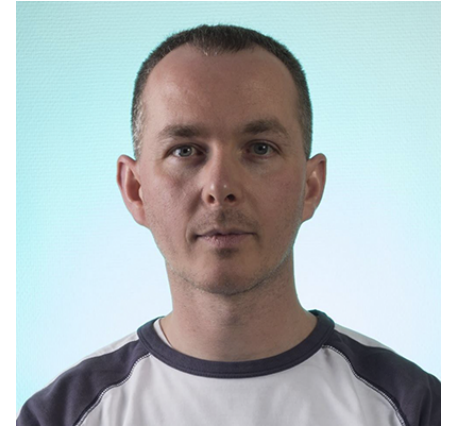
#DelphiBootCamp

Préambule



Patrick Prémartin

- MVP Embarcadero, prestataire informatique et formateur Delphi
- Entreprise Olf Software :
<https://olfsoftware.fr>
- Blog sur l'actualité et le développement en Pascal avec Delphi :
<https://developpeur-pascal.fr>
- Formation Delphi en entreprise :
<https://se-former-a-delphi.fr>
- Contactez moi par le formulaire de l'un de ces sites, en répondant à l'un de mes emails ou faites moi signe sur Twitter et LinkedIn.



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

- 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
- Blog sur le Pascal et Delphi :
<https://www.developpeur-pascal.fr>



Liens utiles

- Sources des exemples et corrigés de cette formation :
<https://github.com/DeveloppeurPascal/ApprendreLaProgrammationAvecDelphi>
- Documentation en ligne « officielle » :
<http://docwiki.embarcadero.com>
- Actualité, blogs, forums « officiels » (en anglais) :
<https://community.idera.com/developer-tools/b/blog>



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

- Embarcadero :
<https://www.embarcadero.com/fr/>
- Portail qualité :
<https://quality.embarcadero.com/>
=> signaler et consulter les anomalies recensées
=> faire des demandes de nouvelles fonctionnalités
- Barnsten :
<https://www.barnsten.com/fr/>



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.



Un peu de pratique



Faire un bouton qui fuit la souris

- Prenons un projet VCL vierge.
- Ajoutons un bouton sur l'écran.
- Déplacer le bouton si la souris s'approche de lui de telle sorte que jamais elle ne puisse cliquer dessus et de préférence sans faire sortir le bouton de l'écran.



Inverser un texte en cours de saisie

- Prenons un projet VCL vierge.
- Ajoutons un libellé et un champ de saisie.
- Pour chaque lettre saisie, on l'affiche en début du libellé.
- Si on écrit « Hello World » dans le champ de saisie on veut obtenir « dlroW olleH » dans le libellé.



Utiliser plusieurs fenêtres dans un programme VCL



Plusieurs fenêtres dans un projet

- Jusque là nous avons vu comment créer des projets VCL avec une seule fenêtre que j'ai qualifié de « fenêtre principale » à plusieurs reprises. Vous allez maintenant comprendre pourquoi.
- Il est rare d'avoir des programmes n'ayant qu'un seul écran. La plupart du temps on a une fenêtre Windows à l'intérieure de laquelle on peut en avoir d'autres qui s'ouvrent comme des boites de dialogue.
- Exemple de boite de dialogue : Outils / Options dans Delphi



Plusieurs fenêtres dans un projet

- Delphi permet bien entendu de gérer plusieurs fenêtres dans un même programme, il suffit de les ajouter au projet.
- Dans le gestionnaire de projet, nous avons la liste de toutes les unités associées aux projets ouverts.
- Jusque là nous n'en avons qu'une. Pour en ajouter une nouvelle on doit passer par l'assistant de création de fiches.



Plusieurs fenêtres dans un projet

- Plusieurs solutions pour ajouter une fiche à un projet VCL :
 - Le menu contextuel sur le projet avec l'option « ajouter nouveau / fiche VCL »
 - Le menu « Fichier / Nouveau / Fiche VCL – Delphi »
 - Le menu « Fichier / Nouveau / Autres » puis « Fiche VCL » dans « Projets Delphi / Fichiers individuels »
- Une fois les nouveaux écrans ajoutés au projet, on se retrouve avec de nouvelles unités visibles dans le gestionnaire de projet.



Plusieurs fenêtres dans un projet

- Une fois les nouveaux écrans ajoutés au projet, on se retrouve avec de nouvelles unités visibles dans le gestionnaire de projet.
- De nouvelles lignes sont ajoutées aussi dans le source du projet par Delphi dans sa « clause USES ».
- Pensez à enregistrer votre projet une fois les ajouts de fenêtres effectués pour donner des noms clairs aux fichiers.



Plusieurs fenêtres dans un projet

- Chaque fenêtre comporte ses composants, ses propriétés et ses événements. Elles sont (globalement) indépendantes les unes des autres.
- Lorsqu'on les utilisera elles seront
 - Soit modales comme les boîtes de dialogues, c'est à dire qu'elles empêcheront d'accéder aux autres fenêtres du même programme.
 - Soit normales et ne bloqueront pas l'accès aux autres fenêtres ouvertes du programme.
- Le choix du mode d'affichage (modal ou pas) est fait dans le code source qui va nous permettre d'ouvrir une fenêtre depuis une autre.



Plusieurs fenêtres dans un projet

- Outre le mode d'affichage (modal ou pas), on peut modifier certains aspect des fenêtres en jouant avec leurs propriétés.
 - BorderIcons gère l'affichage des boutons que l'on trouve sur les fenêtres Windows.
 - BorderStyle modifie le dessin du cadre de la fenêtre et la possibilité d'en changer la taille à l'exécution.
 - BorderWidth spécifie la taille de la bordure lorsqu'on en a une.



Plusieurs fenêtres dans un projet

- Notez au passage la présence des propriétés ClientWidth et ClientHeight que nous avons vues hier pour le TPanel.
- Une fenêtre est un conteneur de composants, elle a sa propre taille (Width / Height) et pour éviter que les composants s'affichent sur ses rebords elle définit par défaut une marge intérieure à laquelle on peut en ajouter une par la propriété Padding.



Plusieurs fenêtres dans un projet

- Autres propriétés à voir sur les fenêtres pour en gérer le format d'affichage et la position :
 - DefaultMonitor qui permet d'indiquer où la fenêtre apparaîtra dans le cas où l'utilisateur a plusieurs écrans sur son ordinateur.
 - FormStyle où l'on retrouve SDI/MDI (obsolètes), affichage normal (par défaut) et StayOnTop (toujours à l'écran par dessus les autres fenêtres).
 - Position qui indique où la fenêtre sera affichée dans l'écran sur lequel elle s'affichera.
 - WindowsState qui permet de savoir si la fenêtre est ouverte en plein écran, réduite dans la barre de tâches ou affichée normalement.



Plusieurs fenêtres dans un projet

- Avec ces éléments vous devez pouvoir positionner et préparer les fenêtres comme vous le voulez.
- Aujourd'hui, faites des tests de ces différentes propriétés pour voir le résultat quand vous aurez une ou plusieurs fenêtres dans votre programme.



Retour sur l'IDE



Fenêtre principale ou pas ?

- Comme pour l'environnement de travail, nous avons des options dans chaque projet. Elles nous permettent d'en définir les paramètres: le comportement, l'emplacement sur le disque dur, les images associées, ...
- Dans le cas où un projet a plusieurs fenêtre il faut dire au programme laquelle ouvrir lorsqu'il se lance. Par défaut c'est la première créée dans le projet.
- Cela se passe dans le menu « Projet / Options » puis la rubrique « Application / Fiches ».



Fenêtre principale ou pas ?

- Dans cet écran on peut définir :
 - La fenêtre principale qui est donc celle que le programme ouvre au démarrage.
 - Les fiches créées automatiquement qui sont celles dont Delphi s'occupe lorsque le programme se lance et qui sont listées dans le source du projet.
 - Les fiches non créées automatiquement qui sont celles que le programmeur est seul à gérer dans son code source.
- Dans certains cas l'ordre de création des fiches automatique est important. On peut les déplacer par drag&drop dans leur liste.



Retour sur le code en Pascal



Soyons unis, parlons unités.

- Nous l'avons vu : un projet Delphi contient un fichier pour le programme et des fichiers complémentaires (les unités).
- Chaque fiche est dans une unité avec un fichier d'extension DFM pour décrire l'écran. Ce fichier est traité comme une ressource de l'unité et donc du programme.
- Voyons le source des unités.



Les unités en Pascal

- Pour créer une unité, il faut passer par le menu contextuel du projet ou l'assistant de création de fichiers, comme pour les nouvelles fiches, mais choisir « unité » à la place de « Fiche VCL ».
- Une unité a deux parties : l'interface et l'implémentation.
- On peut ajouter une rubrique initialization et une rubrique finalization. Nous les verrons plus tard.



Les unités en Pascal

- Le Pascal est un langage déclaratif : on ne peut utiliser quelque chose dans le code que si on a dit quelque part que ça existe. C'est à ça que sert la partie interface de l'unité.
- Dans l'implémentation on reprend ce qu'on a déclaré dans l'interface et on met le code correspondant.
- Seul ce qui est présent dans l'interface est disponible à l'extérieur de l'unité.



Les fiches, des unités un peu spéciales

- Pour les fiches VCL (comme celles de FireMonkey), c'est Delphi qui s'occupe de l'interface à partir de ce qu'on fait en conception de fiche.
- Il est bien entendu possible d'ajouter des choses dans le code mais **NE MODIFIEZ JAMAIS** ce que Delphi a généré sans avoir aussi fait les mêmes modifications dans le concepteur de fiches et les composants concernés. Ca risquerait de générer des anomalies au chargement du fichier que vous ne pourrez corriger qu'un modifiant à la main le .DFM



Les fiches, des unités un peu spéciales

- Quand on crée un fiche, Delphi crée donc deux fichiers : la ressource de l'écran et l'unité qui contient le code pour interpréter cette ressource et agir en fonction de notre code.
- Dans l'unité, la fiche est en fait un type de variable. Une classe qui contient des méthodes et des propriétés réparties dans différentes rubriques.



Les fiches, des unités un peu spéciales

- Dans notre code nous ne manipulons pas la classe mais une instance de la classe représentée par la variable qui porte le nom que nous avons donné à la fiche en conception.
- Par défaut, dans les événements nous travaillons déjà sur notre fiche et n'avons donc pas besoin de le dire.
- Nous en avons besoin quand nous y accédons d'ailleurs.



La clause USES du Pascal

- L'interface d'une unité décrit ce qui est accessible d'ailleurs et codé dans son implémentation.
- Reste à voir comment y accéder depuis une autre unité ou le programme lui-même.



La clause USES du Pascal

- Quand je vous ai montré le source du projet, vous avez vu que les fichiers sont listés en haut : un .PAS par fiche dans une partie qui commence par « uses » (« utilise » en anglais).
- C'est cette « clause » USES qui permet à un fichier de pouvoir accéder à ce qui est déclaré dans l'interface d'un autre.



La clause USES du Pascal

- Dans les fiches on a un USES dans l'interface, rempli la plupart du temps par Delphi en fonction des composants utilisés.
- On peut lui ajouter d'autres unités manuellement. Elles seront visibles dans notre code mais aussi par Delphi en conception.
- On peut aussi créer une clause Uses dans l'implémentation, qui permettra de décrire ce qui sera utilisé dans le code et n'a pas d'intérêt particulier au niveau du concepteur de fiches.



La clause USES du Pascal

- Pour ne pas faire les ajouts manuels d'unités de notre projet, on peut passer par le raccourci Alt+F11 ou le menu « Fichier / Utiliser l'unité ».



Afficher une fiche par programmation

- Lorsque la création des fiches est automatique (déclarée dans le programme par l'intermédiaire des options du projet), la variable qui a le nom de la fiche pointe sur une instance de l'écran dès l'ouverture du programme. On peut y accéder où et quand on veut.



Afficher une fiche par programmation

- Lorsque la création de la fiche n'est pas automatique, l'utilisation de cette variable va générer des erreurs de type « violation d'accès » car elle pointe « dans le vide » tant qu'on ne crée pas nous-même l'instance de la fiche.



Afficher une fiche par programmation

- Dans notre code source, lorsque l'unité qui contient la fiche est accessible dans celle dans laquelle on travaille, on a accès à toutes les propriétés et méthodes publiques de la fiche.
- Celles qui nous intéressent pour le moment sont Show et ShowModal.
- Vous l'aurez déduit : ShowModal ouvre des boites de dialogue modales, Show affiche les autres.
- Lorsqu'on appelle Show ou showModal d'une fiche, ça déclenche l'événement onShow de la fiche, puis ça l'ouvre.



A suivre...

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

Discutez sur les réseaux sociaux avec le hashtag
#DelphiBootCamp

