



Apprendre la programmation avec Delphi

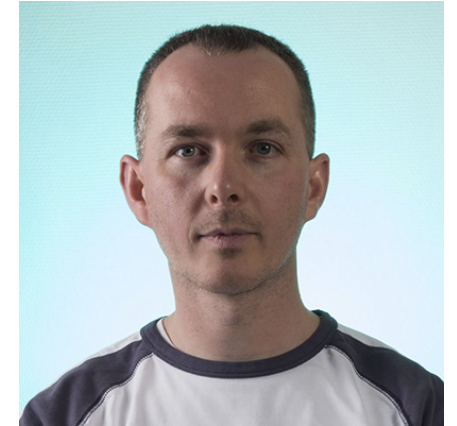
10 avril 2020

#DelphiBootCamp

Préambule



Patrick Prémartin

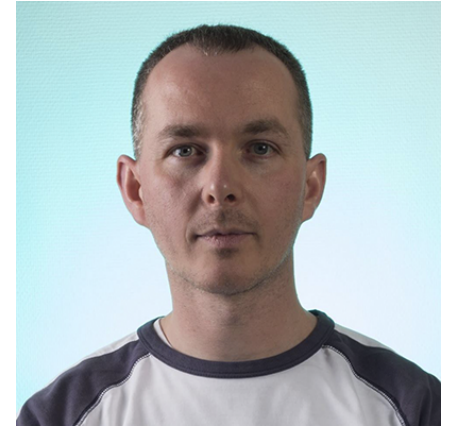


- MVP Embarcadero, prestataire informatique et formateur Delphi
- Entreprise Olf Software
(conseil, développement et assistance technique) :
<https://olfsoftware.fr>
- Formations Delphi en entreprise ou en groupe :
<https://se-former-a-delphi.fr>
- Blog sur l'actualité du développement en Pascal avec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

- Vidéos des rediffusion des sessions :
<https://vasur.fr/delphibootcamp>
- Téléchargement des présentations en PDF :
<https://apprendre-delphi.fr/apprendre-la-programmation-avec-delphi/>
- Sources des exemples et corrigés :
<https://github.com/DeveloppeurPascal/ApprendreLaProgrammationAvecDelphi>



Liens utiles pour la formation

- 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 pour s'informer

- Développeur Pascal
<https://www.developpeur-pascal.fr>
=> actualité de Delphi, articles de fond, trucs et astuces
- Awsome Pascal
<https://github.com/Kromster80/awesome-pascal>
=> une liste d'outils, librairies, composants et logiciels liés au développement en Pascal et Delphi



Liens utiles pour s'informer

- Webinaires mensuels en français sur une thématique précise :
<https://developpeur-pascal.fr/p/6007-webinaires.html>
- D'autres sources d'infos et ressources utiles :
<https://developpeur-pascal.fr/c/4-ressources.html>



Liens utiles pour s'informer

- 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 d'Embarcadero

- 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 d'Embarcadero

- 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 d'Embarcadero

- 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 d'Embarcadero

- Delphi
<https://www.embarcadero.com/fr/products/delphi>
- Téléchargement de Delphi Trial
<https://www.embarcadero.com/fr/products/delphi/start-for-free>
- Téléchargement de Delphi Community Edition
<https://www.embarcadero.com/fr/products/delphi/starter>



Liens utiles de Barnsten

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



Liens utiles de Barnsten

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



A savoir pour coder...

- Raccourcis au clavier dans Delphi :
<https://developpeur-pascal.fr/p/005-les-combinaisons-de-touches-et-raccourcis-clavier-utiles-dans-l-edi-de-delphi.html>



Quelques banques d'images

- Icons8
<https://icons8.com>
- Material Design Icons
<http://materialdesignicons.com>
- GlyFX
<http://www.glyfx.com>



Quelques banques d'images

- Kenney Studio
<https://kenney.nl/assets>
- Adobe Stock
<https://stock.adobe.com/fr/>



Librairies open source

- Icon fonts image List

<https://github.com/EtheaDev/IconFontsImageList>

=> projet open source permettant d'utiliser des caractères d'une fonte TFF comme images dans une liste d'images compatible avec les composants VCL et FMX.



Editeurs de textes

- Notepad++
<https://notepad-plus-plus.org>
- GBEPad
<http://www.gbsoft.fr/gbepad.php>
- Sublime Text
<https://www.sublimetext.com>



Logiciels de conception de bases de données

- TMS Data Modeler

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

- Aqua Data Studio

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

- E/R Studio

<https://www.idera.com/er-studio-enterprise-architecture-solutions>

- Heidi SQL

<https://www.heidisql.com>



Liens utiles pour les bases de données

- SQLite
<https://www.sqlite.org/>
- DB Browser for SQLite
<https://sqlitebrowser.org>
- Les bases de données relationnelles
<https://info.blaisepascal.fr/cpge-bases-de-donnees-relationnelles>
- Cours sur le langage SQL
https://www.w3schools.com/sql/sql_intro.asp



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).



Jour 06 - Session du 30 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - Les modules de données
 - Les listes d'images pour écrans High DPI et 4K
 - L'instruction "if"
 - Stocker des données
 - Conception de bases de données
 - Exemples de programmes



Jour 07 - Session du 31 mars 2020

- Lors de cette session nous avons vu les points suivants :
 - Cas pratique : TSpeedButton et autres boutons VCL
 - Utiliser les exemples fournis avec Delphi
 - Les boucles en Pascal
 - Les bases de données dans Delphi
 - Les composants FireDAC
 - Exemples de programmes



Jour 08 - Session du 1 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Gérer des données en mémoire
 - VCL et bases de données
 - Exemples de programmes



Jour 09 - Session du 2 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Révision des liens utiles
 - Borland Database engine (BDE)
 - Konopka Signature VCL
 - Icon Fonts Image List
 - Pascal et bases de données
 - Exemples de programmes



Jour 10 - Session du 3 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Une session dédiée au codage
 - Gestion des contacts (exercice 4)
 - Gestions d'une liste de courses (exercice 5)



Jour 11 - Session du 8 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - La notion de maître détail
 - Codage de la gestion de bibliothèque



Jour 12 - Session du 9 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Créer des bases automatiquement
 - Conditionner du code
 - Finalisation de la gestion de bibliothèque
 - Faire des requêtes SQL où on ne peut pas en faire



Les cadres VCL



Les cadres VCL

- Nous avons vu les fiches qui accueillent des composants visuels et non visuels.
- Nous avons vu les modules de données qui n'accueillent que les composants non visuels.
- Nous pouvons voir maintenant un troisième type de fiche : les cadres.



Les cadres VCL

- Les cadres sont à la fois des fiches et des composants visuels (ou plutôt un regroupement de composants pouvant être utilisé comme un composant visuel).
- Les cadres ont un concepteur de fiche comme les fiches ou les modules de données.
- Les cadres acceptent des composants visuels et non visuels. Il ont des propriétés et des événements.



Pourquoi utilisez les cadres ?

- Si vous avez un groupe de composants à l’affichage identique et au fonctionnement semblable sur plusieurs fenêtres de votre projet, pourquoi vouloir les copier/coller et donc multiplier les risques d’erreurs lors de la maintenance ?
- Les cadres sont faits pour ça : on définit une seule fois l’affichage des composants et leur comportement.
- Exemple une zone de connexion commune à plusieurs écrans ou plusieurs projets.



Comment utiliser les cadres ?

- Les cadres ont aussi la capacité à être ajoutés dans la palette de composants (par l'option « ajouter à la palette » du menu contextuel de la fiche).
- Une fois dans la palette, les cadres sont vus par les autres fiches comme des composants visuels classiques et peuvent être intégrés à n'importe quelle fenêtre.



Comment utiliser les cadres ?

- Si vous utilisez des cadres sur un projet, pensez à ajouter l'unité qui contient le cadre à votre projet si elle n'en fait pas encore partie ou modifier le chemin de recherche par défaut pour inclure le dossier dans lequel vous stocker les unités de vos cadres à vos projets.
- Sans ça le concepteur de fiches fonctionnera, mais vous aurez des erreurs lors de la compilation puisque Delphi ne saura pas retrouver le source du cadre.



Procédures et fonctions en Pascal



Procédures et fonctions en Pascal

- Je l'ai dit à plusieurs reprises : un développeur de logiciels travaille beaucoup par copier coller.
- Le job d'un développeur est aussi de limiter les choses redondantes afin de simplifier le travail de maintenance.



Procédures et fonctions en Pascal

- Lorsqu'on a besoin de faire plusieurs fois la même chose dans un code source on a plusieurs solutions :
 - des boucles si le code doit s'exécuter à plusieurs reprises au même endroit
 - Des procédures ou fonctions lorsqu'on doit l'utiliser à plusieurs endroits différents



Procédures et fonctions en Pascal

- Nous avons vu les boucles (for, while, repeat until) il y a quelques jours, je ne vais donc pas y revenir.
- J'ai déjà eu l'occasion de créer des procédures au travers des exemples que je vous ai présentés.
- Voyons les plus en détail.



Procédures et fonctions en Pascal

- Trois cas :
 - Si une procédure ou fonction est dans une unité et doit pouvoir être utilisée ailleurs on doit la déclarer dans son interface et la coder dans l'implémentation.
 - Si une procédure ou fonction n'est utile que dans l'unité en cours, on se contente de l'implémenter.
 - Si une procédure ou fonction est liée aux données d'une classe (une fiche par exemple), on l'appellera méthode et on la crée dedans.



Les procédures en Pascal

- La notion de procédure n'existe pas dans tous les langages.
- On pourrait les assimiler à des « fonctions void » en C et dans les langages qui en découlent.



Les procédures en Pascal

- Une procédure a un nom comme tout ce qu'on manipule dans les programmes : son identifiant.
- Une procédure peut avoir des paramètres, un bloc de déclarations locales de variables, constantes, types, procédures et fonctions.
- Une procédure a un bloc de code entre un « begin » et un « end; ».



Les fonctions en Pascal

- Les fonctions sont déclarées comme les procédures mais ont la particularité d'avoir un type.
- Ce type définit le type d'informations qu'elle retourne à son appelant.



Les fonctions en Pascal

- Les fonctions peuvent être
 - affectées à des variables
 - utilisées en paramètre lors de l'appel de procédures, fonctions ou méthodes
 - Utilisées dans des tests



Les fonctions en Pascal

- Historiquement on ne pouvait pas ignorer la valeur de retour d'une fonction, ce qui les distinguait d'une procédure. Désormais ce n'est plus le cas dans l'implémentation du Pascal faite par Delphi.
- Habituez-vous cependant à faire des procédures lorsque vous ne désirez pas avoir de valeur unique en retour et des fonctions dans le cas contraire.



Les fonctions en Pascal

- Pour retourner le résultat d'une fonction il y a trois possibilités :
 - Affecter le résultat à une variable qui porte le nom de la fonction.
 - Passer le résultat en paramètre de l'instruction « exit » qui sort de la fonction lorsqu'on l'appelle.
 - Affecter le résultat à la variable « result » créée automatiquement par le compilateur de Delphi et du même type que la fonction.



Les paramètres des procédures et fonctions

- Les paramètres sont vus comme des variables locales et peuvent être
 - En entrée seule avec « const »
 - En entrée / sortie avec « var »
 - En sortie avec « out »
- Ils peuvent aussi avoir une valeur par défaut.



Les paramètres des procédures et fonctions

- « const » est optionnel pour la déclaration des paramètres en entrée. Sa présence ajoute des contraintes de contrôle lors de la compilation et la façon dont les paramètres sont codés dans le programme final.
- « out » permet de ne pas se préoccuper de la valeur du paramètre et retourne la dernière valeur mise dans la variable associée.
- « var » permet d'avoir la valeur actuelle du paramètre et la modifier.



Réversivité

- En Pascal on peut utiliser tout ce qui est déclaré avant son utilisation.
- Une procédure ou une fonction peuvent s'appeler elles-mêmes. On appelle ça la réversivité.
- On se sert souvent de fonctions réversives pour faire des calculs ou des opérations qu'on aurait fait sous forme de boucle mais qui auraient besoin de variables dans chaque itération de la boucle.



Notions de POO



Notions de POO

- POO est l'abréviation de « programmation orientée objet ».
- Delphi propose une implémentation de l'Object Pascal, la déclinaison objet du Pascal.
- La plupart des choses qu'on manipule dans Delphi sont des objets qu'on appelle maintenant des classes.



Notions de POO

- Les composants visuels et non visuels comme les fiches sont des objets :
 - Ils sont déclarés avec un type « class ».
 - Ils ont des propriétés.
 - Ils ont des méthodes.
- Les événements sont simplement des propriétés pointant vers des méthodes.



Notions de POO

- Un objet hérite des objets de base (TObject ou TInterfacedObject) ou d'une classe existante.
- On ne manipule pas les objets/classes directement : on manipule des instances de ces classes.



Notions de POO

- Pour utiliser une classe il faut créer son instance en appelant son constructeur.
- Dans Delphi on appelle la méthode « Create() » avec les paramètres nécessaires.
- « Create() » est une méthode spéciale déclarée en tant que « constructor » plutôt que « procedure ».



Notions de POO

- Quand on n'utilise plus l'instance d'une classe on doit libérer la mémoire en appelant son destructeur.
- Dans Delphi l'appel du destructeur des instances est fait par l'appel de la méthode « `Free()` » ou la procédure « `FreeAndNil()` » de l'unité `System.SysUtils`.
- Au final c'est la méthode « `Destroy()` » qui est appelée. Elle est déclarée en tant que « destructor » au lieu de « procedure ».



Notions de POO

- Les méthodes DES classes sont des procédures ou des fonctions qui ont accès aux propriétés et autres éléments disponibles dans leur instance de classe.

Leur déclaration commence par « procedure » ou « function » comme les procédures et fonctions classiques.

- Il existe aussi des méthodes DE classe qui travaillent sur le type et non sur une instance.

Leur déclaration commence par « class procedure » ou « class function ».



Notions de POO

- Dans la déclaration d'un type de classe il existe plusieurs blocs pour déclarer ce qu'on utilise :
 - « Private » pour ce qui n'est accessible que par les méthodes de la classe dans l'unité en cours.
 - « Protected » pour ce qui n'est accessible que par les méthodes de la classe et les autres procédures, fonctions ou méthodes implémentées dans la même unité.
 - « Public » pour ce qui sera accessible partout.



Notions de POO

- En pratique la notion de « private » et « protected » sont un peu plus souples que ce que j'ai indiqué mais c'est l'idée.
- On a aussi « strict private » et « strict protected » si on veut coller à la norme.
- La question ne se pose réellement que lorsqu'on partage des unités entre plusieurs projets, plusieurs développeurs ou d'une manière générale lorsqu'on diffuse des composants ou bibliothèques réutilisables.



Notions de POO

- Enfin, Delphi a ajouté la clause « Published » et un bloc de déclaration sans clause particulière pour toutes les classes manipulables par le concepteur de fiches (fiches, cadres, modules de données).
- Dans la partie « published » on retrouve des déclarations de propriétés et d'événements.
- En pratique, sauf si vous concevez des composants, vous n'y toucherez jamais.



Notions de POO

- On peut déclarer des variables dans n'importe quel bloc de déclaration d'une classe. Il faut les mettre en premier et on n'a pas besoin de mettre « var » devant sauf si on a aussi déclaré des types locaux ou des constantes.
- Par convention on met un « f » devant les noms de variables dans une classe. Ca permet de les distinguer des propriétés.



Notions de POO

- Les propriétés sont des variables spéciales : elles sont vues comme des variables par le code mais le compilateur y accède par un « getter » et un « setter ».
- Les deux sont facultatifs.



Notions de POO

- Le « getter » permet de lire la valeur depuis la variable de stockage ou une fonction de la classe.
- Le « setter » permet d'écrire la valeur dans une variable de stockage ou par l'intermédiaire d'une procédure de la classe.



Notions de POO

- Les propriétés sont déclarables en abrégé sous la forme *property NomVariable:Type;*
ce qui donnera finalement *property NomVariable:Type read fNomVariable write setNomVariable;*
- Lorsqu'on déclare une classe dans Delphi on peut lui laisser faire le prérenseignement de l'implémentation et le remplissage d'un certain nombre d'informations grâce à la combinaison de touches Ctrl + Maj + C.



Notions de POO

- Les événements sont des propriétés dont le type est une procédure ou une fonction d'objet.
- La plupart des événements de base (onClic par exemple) sont du type TNotifyEvent.



A suivre...

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

Discutez sur les réseaux sociaux avec le hashtag
#DelphiBootCamp

