

Apprendre la programmation multiplateforme avec Delphi

24 avril 2020

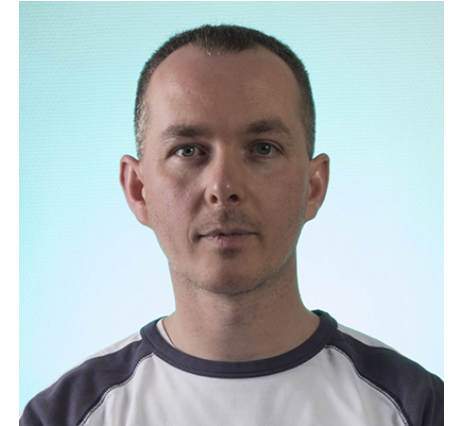
#DelphiBootCamp



Préambule



Patrick Prémartin

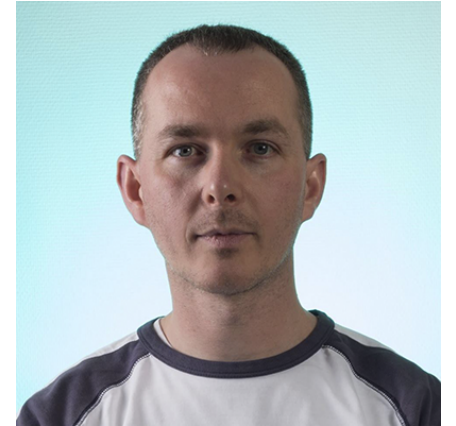


- MVP Embarcadero, prestataire informatique et formateur Delphi
- Entreprise Olf Software
(conseil, développement et assistance technique) :
<https://olfsoftware.fr>
- Formations Delphi en entreprise ou en groupe :
<https://se-former-a-delphi.fr>
- Blog sur l'actualité du développement en Pascal avec 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.
- Les rediffusions accompagnées des PDF de présentation et des sources des exemples sont disponibles sur le site

<https://apprendre-delphi.fr>

et la chaîne YouTube de Barnsten

<https://vasur.fr/delphibootcamp>



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 ?

- Toute personne désireuse de faire de la programmation peut reprendre ce bootcamp du début afin d'apprendre les bases du développement logiciel et le langage Pascal.
Les rediffusions des sessions précédentes sont là pour ça.
- Pour suivre la phase 2 de cette formation il faut avoir des connaissances en développement.
- Des connaissances du langage Pascal ou de la VCL peuvent être utiles mais ne sont pas obligatoires.



Découpage de cette formation

- Cette formation n'est pas le MCU mais elle est quand même découpée en deux phases :
 - La phase 1 concernait la prise en main de Delphi, du Pascal et de la programmation pour Windows avec la VCL.
 - La phase 2 concerne la programmation multiplateforme en utilisant le framework FireMonkey de Delphi.
 - La phase 3 on ne sait pas encore, ça dépend de vous...



Matériel nécessaire

- Pour suivre cette formation il ne vous faut qu'un navigateur web et une connexion Internet. Cela vous permettra de suivre les sessions en direct et de voir les rediffusions.
- Pour pratiquer, en revanche, vous devrez posséder une version de Delphi et l'exécuter sous Windows (sur un PC ou dans une machine virtuelle).
- Pour développer sur d'autres appareils (smartphones et tablettes par exemple) il vous faudra un appareil de test à connecter à votre ordinateur.



Matériel nécessaire

- Le formateur utilise une édition Entreprise de Delphi 10.3.3 Rio.
- Les exemples proposés fonctionnent dans cette version.
- Aucune garantie n'est proposée pour les versions antérieures (ni les suivantes mais en 25 ans Delphi a largement démontré sa capacité à garder une compatibilité ascendante).
- L'ordinateur utilisé est une machine virtuelle Windows gérée par VM Ware Fusion sur un iMac de 2011.



Où trouver des infos ?



Liens utiles de cette formation en ligne

- Pour voir les rediffusions de de la formation :
 - Sur la chaîne YouTube de Barnsten :
<https://vasur.fr/delphibootcamp>
 - Sur Vimeo :
<https://vimeo.com/channels/1552230>



Liens utiles de cette formation en ligne

- 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>
- 25 ans de Delphi
<https://delphi.embarcadero.com>



Liens utiles d'Embarcadero

- Portail qualité :
<https://quality.embarcadero.com/>
=> signaler et consulter les anomalies recensées
=> faire des demandes de nouvelles fonctionnalités
- Portail de téléchargement et licences :
<https://my.embarcadero.com>



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/featured>
- Rediffusion des webinaires :
https://www.youtube.com/watch?v=sgPSimYtzYc&list=PLHLdMyq6m8_sTwOTGwziNlcn443POnmDd
- Rediffusion de cette formation :
https://www.youtube.com/watch?v=2bXx63WQank&list=PLHLdMyq6m8_s-OxVA_qZ4O_YvVgl_1nOp



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 TTF comme images dans une liste d'images compatible avec les composants VCL et FMX.
- TFrameStand et TFormStand
<https://github.com/andrea-magni/TFrameStand>
=> pour animer vos écrans FireMonkey et créer des boîtes de dialogues sous forme de cadres presque sans codage.



Librairies open source

- De nombreux exemples de projets FMX
<https://github.com/FMXExpress/Cross-Platform-Samples>



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...

Phase 2 : Pascal, Pascal Objet, FMX et FireDAC



Jour 15 - Session du 15 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Introduction à la phase 2 de cette formation
 - Licences Delphi
 - Où trouver des infos ?
 - Dans les épisodes précédents... de la phase 1
 - Rappels sur Delphi, le langage Pascal et le langage Object Pascal
 - Présentation de FireMonkey
 - Préparer Delphi pour le développement multiplateforme
 - Un premier projet



Jour 16 - Session du 16 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Projets FMX
 - L'assistant de création de projets
 - Le gestionnaire de projets
 - Les options de projet
 - Le concepteur de fiches FMX
 - Des exemples d'écrans FMX
 - Différences entre FMX et VCL
 - Les fiches multiplateforme
 - Second projet multiplateforme



Jour 17 - Session du 17 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Les propriétés des composants FMX
 - Les événements des composants FMX
 - Des composants FMX de base
 - Exercice 6 : faites vos jeux



Jour 18 - Session du 20 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Retour sur la session 17
 - La compilation conditionnelle
 - Les animations
 - Les effets
 - Moteur physique : Box 2D
 - La 3D dans un projet FireMonkey



Jour 19 - Session du 21 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Les sprite sheets
 - Mettre un peu d'ambiance
 - Correction exercice 6 : le jeu du pendu (première partie)



Jour 20 - Session du 22 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Les cadres dans FMX
 - Correction exercice 6 : le jeu du pendu (suite et fin)
 - Les modules de données FMX
 - Les bases de données avec FMX
 - LiveBindings



Jour 21 - Session du 23 avril 2020

- Lors de cette session nous avons vu les points suivants :
 - Des écrans dans un programme



Prévisualiser les écrans



Prévisualiser les écrans

- Lorsque nous avons vu les fonctionnalités du concepteur de fiches de FMX j'en ai évité deux volontairement : la prévisualisation en temps réel depuis l'EDI et depuis les appareils connectés.
- Dans les onglets disponibles au niveau du gestionnaire de projets nous avons un onglet « aperçu multipériphérique » dans lequel voir à quoi ressemblerait la fiche en cours avec le style et le format de différents appareils.



Prévisualiser les écrans

- L'EDI de Delphi / C++Builder intègre aussi une fonctionnalité permettant de visualiser les fiches en cours d'édition directement sur les appareils concernés.
- Il faut pour cela y installer le programme LivePreview disponible dans le dossier d'installation C:\Program Files (x86)\Embarcadero\Studio\20.0\source\Tools\FireUIAppPreview
- Ce programme dialogue avec l'EDI et « exécute » le .FMX sur l'appareil où il est lancé.



TImageList de FireMonkey



TImageList de FireMonkey

- En VCL nous avons vu les TImageList historiques puis les composants TImageCollection et TVirtualImageList pour gérer des images en plusieurs définitions destinées aux écrans 4K et High DPI.
- En FMX nous n'avons que TImageList mais celui-ci est depuis le début destiné à gérer des images en plusieurs résolutions. Malheureusement il est moins intuitif que la collection d'images apparue avec Delphi 10.3 Rio en VCL.



TImageList de FireMonkey

- Le principe reste le même que pour la VCL : placez une liste d'images sur un module de données (pour ne pas alourdir vos fiches) et pour chaque image vous pouvez fournir plusieurs versions de tailles différentes.
- A la différence des listes d'images VCL, le TImageList de FMX ne fixe pas la taille de l'image dès le départ, c'est le composant qui les utilise qui choisit quelle taille d'image afficher.

Finis le 16x16 imposé sur les boutons depuis Windows 3.1 !



TImageList de FireMonkey

- Autre avantage des listes d'image FMX, c'est la prise en charge des spritesheets pour le chargement des images : le même fichier peut générer plusieurs images différentes dans la liste.



Programmation synchrone ou asynchrone ?



Programmation synchrone ou asynchrone ?

- Quand on programme pour Windows avec la VCL on ne se pose pas vraiment de questions sur le fait de bloquer ou pas l'interface utilisateur avec des traitements longs même si ce n'est pas recommandé en terme d'expérience utilisateur.
- On peut contourner le problème en utilisant un « `Application.ProcessMessages` » dans les boucles.
- Avec FireMonkey ce n'est pas possible.



Programmation synchrone ou asynchrone ?

- Quand on développe pour iOS et Android il est interdit de bloquer l'interface utilisateur. La sanction est assez rapide : le système d'exploitation le détecte et demande à l'utilisateur s'il veut fermer le programme.
- On ne doit en aucun cas bloquer l'interface et la boucle principale de gestion d'événements de Delphi doit donc être active le maximum de temps possible dans la durée de vie d'un programme.



Programmation synchrone ou asynchrone ?

- Il est donc nécessaire de repenser notre façon de programmer pour faire des choses de façon asynchrones plutôt que synchrones.
- Parmi les gros changements sur mobiles, il y a notamment la méthode `ShowModal()` qui ne fonctionne pas sur Android. Il est donc recommandé de ne pas l'utiliser non plus pour iOS.
- Autre changement : les boîtes de dialogue standard de la VCL ne fonctionnent plus ou plus de la même façon en FMX, à part `ShowMessage()`.



Programmation synchrone ou asynchrone ?

- Pour programmer de façon asynchrone Delphi propose différentes solutions :
 - La possibilité de passer du code (sous forme de méthodes anonymes), des procédures, fonctions et méthodes en paramètre à des procédures et fonctions. C'est le principe des hooks et des fonctions de callback d'autres langages ou frameworks.
 - La possibilité de passer des parties du code en tâche de fond avec les processus et tâches.



Programmation synchrone ou asynchrone ?

- Et tant qu'à faire, on peut aussi utiliser ces techniques dans les projets FMX pour Windows, macOS et Linux comme on peut les utiliser avec les projets VCL puisque cela fait partie du bagage du langage Pascal implémenté dans Delphi et de la RTL.



Les méthodes anonymes



Les méthodes anonymes

- Nous avons déjà parlé des procédures, des fonctions et des méthodes (qui sont en fait des procédures et fonctions liées à des classes).
- Voyons maintenant les méthodes anonymes qui sont elles-aussi des procédures et des fonctions, mais qui ont la particularité de ne pas avoir de nom...



Des procédures et fonctions en paramètre

- ... mais avant de parler de méthodes anonymes, voyons un point important du langage Pascal : les procédures et fonction sont en fait du code attaché à des constantes de type procédure ou fonction !
- Et on peut aussi créer des variables dans lesquelles stocker des procédures ou des fonctions.
- C'est comme ça que sont gérés les événements, ce que nous avons déjà vu ensemble sans que ne le sachiez.



Des procédures et fonctions en paramètre

- On peut créer un type de donnée avec une interface de procédure ou de fonction. Ca se présente sous la forme :

Type

```
TMyProc = procedure;  
TMyFuncInt = function : integer;  
TMyFuncBool = function : boolean;
```

- Tout ce qui fonctionne en création de procédure et de fonction peut être spécifié dans un type.



Des procédures et fonctions en paramètre

- Une fois le type créé, on peut déclarer des variables de ce type et leur affecter des noms de procédures et fonction correspondant à cette description.
- On peut également le faire avec des méthodes en ajoutant « of object » à la déclaration du type. Ca devient :

Type

TMyMethodProc = procedure of object;

TMyMethodFuncInt = function : integer of object;



Des procédures et fonctions en paramètre

- C'est comme ça que sont déclarés les événements. La plupart étant des TNotifyEvent dont la déclaration est simplement :

Type

```
TNotifyEvent = procedure(Sender:TObject) of object;
```

- C'est pour ça qu'on a pu créer des événements à la main ou depuis le concepteur de fiche et les associer à « onClick » qui est en fait une propriété de type TNotifyEvent, donc une variable...



Des procédures et fonctions en paramètre

- Pouvant utiliser des variables pour « pointer » sur des procédures et des fonctions, on peut logiquement passer des procédures et fonctions en paramètre d'autres procédures et fonctions.
- Grâce à ce mécanisme on peut exécuter du code sans savoir vraiment sur quoi on travaille ou en appelant autre chose.
- Par exemple appeler une boucle sous forme de procédure qui exécute du code fait ailleurs.



Des procédures et fonctions en paramètre

- Ceci étant vu, nous pouvons passer à l'étape ultime : plutôt que passer des procédures, fonctions ou méthodes en paramètres, nous allons directement passer du code.
- C'est ce que nous appelons des méthodes anonymes et elles pullulent dans FireMonkey pour gérer tout ce qui est asynchrone.
- On peut le voir comme des événements que l'on appellerait directement depuis notre code.



Les méthodes anonymes

- Une méthode anonyme est donc une procédure ou une fonction sans nom.
- Pour pouvoir passer du code en paramètre ou dans des variables il faut définir un type décrivant la procédure ou la méthode qu'on s'attend à passer. Pour le reste ça fonctionne pareil que les autres types `procedure` / `function` / `procedure of object` et `function of object`.



Les méthodes anonymes

- La déclaration d'un type de méthode anonyme se fait sous forme de référence. On dit clairement qu'on va créer un pointeur vers la zone de mémoire dans laquelle le compilateur de Delphi stockera le code utilisé.
- Voici un exemple :

Type

```
TMyProcAnon = reference to procedure;  
TMyFuncAnonInt = reference to function:integer;
```



Exercices 7

- Sur la phase 1 de la formation je vous avais proposé de développer un carnet d'adresses et une gestion de liste de courses en VCL.
- Je vous propose de les refaire en utilisant FireMonkey et en les mettant sur vos smartphones.



Exercices 7

- Faites les avec la méthode que vous préférez en vous débrouillant pour que les programmes développés sous Windows aient un look qui fonctionne aussi sur téléphone.
- Reportez-vous à la rediffusion de la session 10 pour voir ce qui avait été codé en VCL et aux sources correspondants si vous le désirez.
- Nous ferons un corrigé dans les jours qui viennent et verrons ensuite comment synchroniser les bases mobiles et Windows.



Exercice 7a – répertoire téléphonique FMX

- En un ou plusieurs écrans mettez en place une gestion de contacts (nom, prénom, email, téléphone) qui fonctionnera sur votre ordinateur sous Windows, mais qui tournera également sur votre smartphone Android ou iOS (sous réserve d'avoir obtenu un compte développeur payant auprès d'Apple).
- Utilisez une base de données locale soit sous forme de TFDMemTable soit avec une base de données SQLite.



Exercice 7b – liste de courses FMX

- En un ou plusieurs écrans mettez en place une gestion de liste de courses (produit à acheter, acheté O/N) qui fonctionnera sur votre ordinateur sous Windows, mais qui tournera également sur votre smartphone Android ou iOS (sous réserve d'avoir obtenu un compte développeur payant auprès d'Apple).
- Utilisez une base de données locale soit sous forme de TFDMemTable soit avec une base de données SQLite.



A suivre...

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

Discutez sur les réseaux sociaux avec le hashtag
#DelphiBootCamp

