

Apprendre la programmation avec Delphi

27 mars 2020

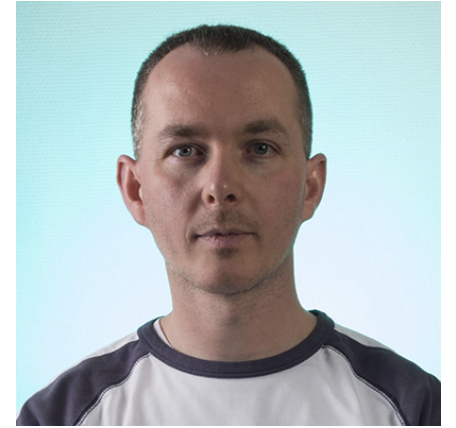
#DelphiBootCamp

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-edi-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.



Un peu de pratique



Changer la valeur d'un libellé

- Prenons un projet VCL vierge.
- Ajoutons un libellé initialisé à 0 et deux boutons.
- Le premier bouton va augmenter le nombre présent sur le libellé.
- Le second bouton va diminuer le nombre présent sur le libellé.



Horloge

- Prenons un projet VCL vierge.
- Mettons un libellé et un timer sur l'écran.
- Toutes les secondes le libellé doit être mis à jour avec la bonne heure.



Gérer les plantages



Gérer les plantages

- Vous l'avez vu lorsque nous avons fait les programmes d'exemples en direct et sans doute quand vous avez pratiqué : il existe plusieurs types d'erreurs dans les programmes.
- Le premier type d'erreur est une erreur de saisie, de syntaxe ou de code d'une manière générale qui empêche Delphi de compiler et donc d'exécuter le programme.
- Le second type ce sont les erreurs d'exécution.



Gérer les plantages

- Pour les erreurs à la compilation la fenêtre de message (par défaut en bas de l'écran) donne un certain nombre d'indications et si on cherche ou clique sur un message on obtient l'endroit dans le source où se trouve l'anomalie repérée.
- Ces erreurs là ne posent pas de vrai problème : comme elles empêchent de compiler, on peut les identifier et les corriger.



Gérer les plantages

- Les erreurs d'exécution sont en revanche beaucoup plus gênantes.
- Elles sont nombreuses à pouvoir se produire :
 - Des erreurs liées à des manipulations de l'utilisateur
 - Des erreurs liées à l'ordinateur sur lequel est lancé le programme
 - Des erreurs liées à des incohérences ou des problèmes dans ce que le développeur a codé (erreur de logique dans l'algorithme par exemple)
 - Des erreurs et plantages aléatoires
 - Des erreurs d'affichage d'informations
 - Des erreurs de conception



Gérer les plantages

- C'est à cause de ces erreurs qu'il faut toujours faire de nombreux tests de nos programmes avant de les transférer à quelqu'un.
- C'est encore plus critique si nous manipulons des données car une erreur pourrait générer leur perte et donc des conséquences gênantes ou désastreuses pour les utilisateurs.



Gérer les plantages

- La Pascal et Delphi permettent de contourner le problème dans plusieurs de ces cas, en revanche rien ne permet de détecter ou corriger les erreurs de conception ou celles qui dépendent de l'utilisateur et de la configuration de son ordinateur.
- Nous allons voir comment intercepter et traiter les nôtres, pour les autres la règle est toujours la même : faire des sauvegardes régulièrement et en conserver plusieurs exemplaires hors de chez soi ou de son bureau.



Gérer les plantages

- Les plantages comme les violations d'accès que je vous ai montrées ces derniers jours sont liés à des erreurs de conception ou de logique. On utilise quelque chose qui ne devrait pas être utilisé à cet endroit.
- Seule la reproduction du problème permet de les identifier en faisant du débogage du programme.
- Certains outils comme EurekaLog permettent d'avoir des informations complémentaires sur les anomalies une fois les programme diffusés.



Gérer les plantages

- Pour les erreurs d'interface utilisateur, c'est la pratique qui vous dira quoi faire, où et comment. La simplification des écrans reste de mise.
- Enfin, pour les autres erreurs, certaines sont prévisibles comme par exemple de saisir des lettres dans un champ de saisie à l'écran où on demande de saisir des chiffres.
- Ces erreurs là doivent être traitées dans les programmes pour ne pas planter l'utilisateur avec des messages bizarres.



Gérer les plantages

- Surtout qu'un certain nombre d'erreurs (ou plutôt d'anomalies) prévisibles n'informeront pas l'utilisateur par une fenêtre disant qu'il y a eu un problème.
- Gardez en tête que les plantages logiciels peuvent être sournois et que nous devons faire en sorte de les éliminer proprement au maximum avant qu'ils ne se produisent.
- Je dis « proprement » car il y a une méthode sale...



Gérer les plantages

- Sous Delphi, les plantages de programmes génèrent des exceptions.
- Pour gérer des plantages nous allons donc gérer des exceptions.
- La gestion des exceptions se fait à deux niveau :
 - L'envoi d'exception pour signaler un problème, ce que fait Delphi quand « ça plante ».
 - L'interception des exceptions et leur traitement.



Gérer les ~~plantages~~ exceptions

- Nous avons vu les blocs « begin ... end » pour regrouper des lignes de commandes ensemble.
- Voyons maintenant les blocs « try ... finally ... end » et « try ... except ... end ».



Gérer les exceptions

- Les blocs d'instruction commençant par « try » sont des blocs qui exécutent du code comme un « begin ... end » mais qui en plus agissent si une exception est déclenchée dans leur code.
- « try ... finally ... end » ne traite pas l'exception, il s'assure juste que ce qui se trouve dans la partie « finally » sera exécuté même si une exception est déclenchée entre « try » et « finally »
- « try ... except ... end » en revanche intercepte les exceptions et nous permet de les gérer.



Gérer les exceptions

- Les blocs d'instructions peuvent contenir d'autres blocs d'instructions.
- Vous pouvez avoir des « begin ... end », « try ... finally ... end » et « try ... except ... end » imbriqués les uns dans les autres.



Gérer les exceptions

- Pensez à garder un code source lisible.
- L'indentation en est la clé.
- Ctrl+D ou « Formater le source » dans le menu contextuel de l'éditeur de code ou le menu « Projet / Formater les sources du projet » lorsque votre version les proposent permettent de le faire simplement.



Méthode sale, à ne pas utiliser !

- Je vous le montre mais c'est à ranger dans la case « c'est dangereux, ne le refaites pas chez vous sans la présence d'un professionnel de la profession ».
- Le bloc « try ... except ... end » intercepte toutes les exceptions, il empêche l'affichage des messages d'erreur à l'utilisateur.
- Il suffit d'encadrer notre code par lui et de ne rien mettre entre « except » et « end ».



Les fiches vues du code



Les fiches vues du code

- Nous avons vu hier les propriétés qui permettent de gérer le format d'affichage des fenêtres. Il est temps maintenant de terminer sur leur manipulation.



Les fiches vues du code

- Les méthodes Show et ShowModal sur une variable correspondant à une fiche permettent de l'afficher normalement ou en modal.
- La méthode Hide permet de masquer une fenêtre affichée.
- On peut consulter la propriété Visible pour savoir si une fiche est affichée ou pas.



Les fiches vues du code

- Show n'interrompt pas l'exécution du code.
- ShowModal interrompt l'exécution du code jusqu'à la fermeture de la fiche appelée.
- On ferme une fiche en appelant sa méthode Close.



Les fiches vues du code

- Lorsqu'une fenêtre était affichée avec ShowModal, Close retourne la valeur de sa propriété ModalResult au bloc de code appelant et lui rend la main.
- La propriété ModalResult peut être remplie par le développeur dans le code mais elle peut aussi l'être directement depuis le concepteur de fiche.



Les fiches vues du code

- On retrouve une propriété ModalResult sur les boutons.
- Si on met autre chose que mrNone sur le ModalResult d'un bouton, celui-ci va déclencher la fermeture de la fenêtre et renvoyer l'information au bloc de code appelant.
- Si le bouton a un événement onClic géré par le programme, il sera exécuté avant de fermer la fenêtre.



Les fiches vues du code

- Du côté des événements de la fiche il y en a plusieurs à voir :
 - Create : lorsque le programme (ou le développeur) crée une instance de la classe de la fenêtre et l'associe à sa variable.
 - Show : lorsque l'on affiche la fenêtre par Show, « Visible := true » ou ShowModal
 - Hide : lorsque l'on cache la fenêtre par Hide ou « Visible := false »
 - Activate : lorsque la fiche reçoit le focus (quand elle devient active)
 - CloseQuery : lorsque la méthode Close de la fiche est appelée, c'est le seul endroit qui permet d'empêcher la fermeture de la fenêtre
 - Close : lorsque la fenêtre est fermée



Les fiches vues du code

- Nous avons vu que Delphi prend en charge la création automatique des fenêtres mais nous pouvons en créer à la main.
- Pour ça il est nécessaire d'avoir une variable du type de la fiche que l'on veut créer et afficher.
- On doit en appeler la méthode `Create()` et lui passer son propriétaire en paramètre (en général « Self » qui correspond à la fiche dans laquelle on est).



Les fiches vues du code

- Une fois le `Create()` effectué, c'est comme si la fiche avait été créée au début du programme. On peut se servir de notre variable comme de celles que Delphi crée par défaut.
- Pour Supprimer une fiche on appelle sa méthode `Free` ou on passe par la procédure `FreeAndNil()` à laquelle on passe notre variable.



Les fiches vues du code

- Avant d'utiliser une variable liée à une fiche, il est préférable de s'assurer qu'elle est bien créée. C'est la fonction `Assigned()` qui permet de le faire.
- Dans le cas des fenêtres modales utilisées qu'à un seul endroit, afin d'éviter les fuites de mémoire, le plus sûr est de reprendre ce code :

```
frm := TFrm.Create(Self);  
try  
    frm.ShowModal;  
finally  
    frm.Free;  
end;
```

(c) Patrick Prémartin / apprendre-delphi.fr



Les modules de données



Les module de données

- Les fiches permettent de stocker des composants visuels et non visuels, mais comment faire quand on ne veut pas mettre partout la même chose ?
- Le copier/coller est un truc de développeur, mais il implique aussi de permettre de faire rapidement des corrections ou améliorations lorsque c'est nécessaire.
- Dupliquer du code ou pire (des composants) dans tous les écrans rend les choses ingérables à terme.



Les module de données

- Historiquement le problème se posait surtout pour les connexions aux bases de données qui ne doivent se faire qu'une seule fois dans un programme.



Les module de données

- Le problème est également apparu lorsqu'on a commencé à mettre des images un peu partout dans l'interface utilisateur.
- Quel intérêt de mettre une liste d'image identique sur chaque fiche à part faire grossir la taille des fichiers DFM et du programme ?
- Compliquer l'ajout d'une image pour un bouton qu'on utiliserait à plusieurs endroits est-il vraiment nécessaire ?



Les module de données

- Dans le code, pour éviter ces duplications inutiles, on a les procédures et les fonctions dont je parlerai bientôt.
- Dans le concepteur de fiche on a une fiche un peu particulière : le module de données (TDataModule).



Les module de données

- Un module de donnée est une fiche qui ne s'affiche pas.
- Le module de données permet d'accueillir tous les composants non visuels.
- Les composants visuels sont réservés aux fiches.



Les module de données

- Pour qu'une fiche puisse accéder aux composants présents dans un module de données en conception d'écran, il faut que l'unité du module de données soit accessible dans l'interface de la fiche.
- Nous l'avons vu hier : l'interface d'une unité est visible par tout ce qui utilise l'unité. Le concepteur de fiches a accès à l'interface des fiche qu'on ouvre dans Delphi.



Les module de données

- Pour que tout fonctionne à l'exécution il faut que les modules de données soient créés automatiquement avant les fiches qui les utilisent.
- Ca se gère dans les options du projet, dans le menu « application / fiches » comme le choix de la fiche principale du projet.



Les module de données

- On peut se servir d'un module de données pour regrouper à un seul endroit tout ce qui donne accès aux bases de données mais aussi des listes d'images, des listes d'actions et d'autres composants.
- En terme de codage, un module de données est comme une fiche avec des propriétés et des évènements, c'est aussi une variable dans les sources des programmes.



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

