



Apprendre la programmation avec Delphi

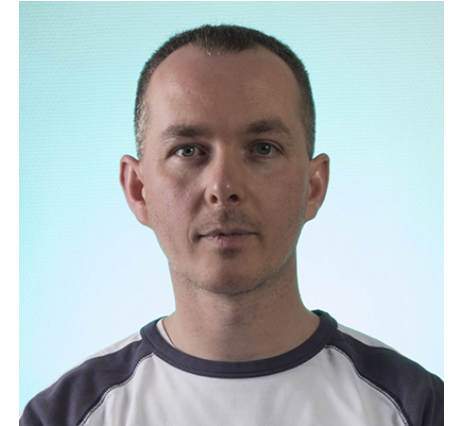
25 mars 2020

#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.



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.



Faire un projet VCL

(Windows à 100% en 32 et 64 bits)



A savoir sur les composants

- Je ne ferai pas le tour de tous les composants disponibles dans Delphi, en VCL comme dans les jours qui viennent en FMX. Il y en a trop. L'aide, les exemples, les forums de discussion, les tutoriels en ligne et les nombreuses vidéos sont là pour vous aider.
- Sachez juste qu'il y a (pour moi) cinq familles de composants :
 - Les composants non visuels (on va y revenir rapidement, no stress)
 - Les composants de saisie (TEdit, TMemo, TCheckBox, TRadioButton, ...)
 - Les composants d'affichage (TLabel, TListBox, ...)
 - Les composants conteneurs (TPanel, TGroupBox, ...)
 - Leur équivalent liés aux ensembles de données (en TDBxxx)



A savoir sur les composants

- Les composants ont des propriétés communes et des propriétés qui ne dépendent que d'eux mêmes.
- Ils ont tous une propriété « Name » qui nous permet d'accéder à leurs informations depuis le code source. Elle est obligatoire et Delphi la renseigne de lui-même en numérotant les composants. Remplacez la toujours par quelque chose qui vous dit à quoi le composant correspond.



A savoir sur les composants

- Les composants visuels ont en commun des propriétés pour gérer leur affichage (position, taille, alignement, ancrage, marges intérieures et extérieures).
- Par définition, les composants non visuels n'ayant pas de représentation graphique lorsqu'on lance le programme, ils n'ont pas ces propriétés.



A savoir sur les composants

- La plupart des composants ont également une propriété « Tag » qui permet de stocker une valeur numérique entière. Ce nombre peut être négatif ou positif, à notre convenance. On en fait ce qu'on veut. Delphi n'y touche pas.
- Toutes les propriétés disponibles dans l'inspecteur d'objets le sont aussi dans le code source en Pascal.



A savoir sur les composants

- Les composants ont des événements. Ce sont des blocs de code qui s'exécutent lorsqu'une action est réalisée par l'utilisateur (à la souris, au doigt, au clavier) ou par le programme lui-même.
- Les composants visuels ont des événements communs pour gérer ces actions et en ont à eux pour gérer leur fonctionnement interne et nous permettre d'interagir avec.
- Dans les autres langages de développement on appelle aussi ça des événements ou des hook (très répandus dans les CMS pour le web)



A savoir sur les composants

- Parmi les événements les plus courants on trouve :
 - `onClic` qui se déclenche lorsqu'on clique avec la souris (ou un doigt sur les écrans tactiles) sur un composant
 - `onMouseOver` qui se déclenche quand on promène la souris sur un composant
 - `onMouseEnter` quand la souris entre sur un composant
 - `onMouseLeave` quand la souris sort d'un composant
 - `onKeyDown` quand on appui sur une touche sur un composant en saisie
 - `onKeyUp` quand on relâche la touche



A savoir sur les composants

- Parmi les événements les plus courants on trouve :
 - `onChange` qui se déclenche lorsqu'on modifie la valeur d'un champ de saisie
 - `onResize` quand le composant change de taille
 - `onMouseDown` quand on appui sur un bouton de la souris
 - `onMouseUp` quand on relâche le bouton de la souris
 - Et ainsi de suite...
- Quelques notions d'anglais peuvent vous servir, mais vous verrez qu'on se sert toujours des mêmes et pour les autres on a la documentation avec la touche F1.



Etre développeur



Etre développeur, un mode de vie !

- [Confidentiel]
Les informaticiens se doivent d'être flemmards et d'économiser leur énergie au maximum.
[/Confidentiel]
- Il y a une vraie raison à cela (ou en tout cas vous devrez l'évoquer) : cultiver l'envie d'en faire le moins possible oblige à penser à des solutions permettant d'y parvenir.
- Notre rôle est d'apporter des solutions simples aux problèmes qui ne le sont pas toujours.



Etre développeur, un mode de pensée !

- Contrairement aux idées reçues nous ne sommes pas payés (quand nous le sommes) pour « pisser du code » mais pour réfléchir et transformer les résultats de nos réflexions en algorithmes qui eux-mêmes deviennent des lignes de code et donc des programmes.
- Avec l'expérience ces étapes se font en même temps ce qui donne l'impression que nous sommes des machines à taper sur des claviers.
- Bien entendu tous les programmes ne se prêtent pas à la créativité. On fait souvent la même chose.



Etre développeur, un mode de pensée !

- Il y a de nombreuses façons de faire quelque chose mais nous avons tendance à toujours utiliser la même.
- Il faut s'efforcer à choisir la solution la plus adaptée et la plus facile à mettre en œuvre car ce sera celle qui sera aussi la plus facile à maintenir sur la durée.
- Comme pour les auteurs, il y a des jours sans et le syndrome de la page blanche peut nous atteindre aussi.



Un développeur doit savoir s'économiser

- Nous avons tous une souris, un touchpad ou un trackpad pour déplacer la souris à l'écran, mais quand on code ça peut vite faire mal au poignet.
- Le syndrome du canal carpien est d'ailleurs reconnu comme maladie professionnelle pour les informaticiens.
- On doit donc se servir le moins souvent possible de la souris.



Simplifier la vie du développeur

- Pour ne pas se servir de la souris il faut savoir se servir de son clavier.
- Delphi propose de très nombreuses combinaisons de touches pour accéder aux différents éléments des écrans, aux menus, afficher des boîtes de paramétrage et d'options.
- J'ai déjà évoqué les touches de fonctions F1 pour l'aide, F12 pour passer du code à la conception sur une fiche.
- Il y en aura d'autres à base de Ctrl, MAJ et ALT quand on sera dans le code source.



Simplifier la vie du développeur

- Une liste des raccourcis les plus utilisés (en tout cas par moi) est disponible sur la page https://developpeur-pascal.fr/p/_005-les-combinaisons-de-touches-et-raccourcis-clavier-utiles-dans-l-edition-delphi.html
- Vous en trouverez aussi sur un PDF de Cary Jensen disponible sur <http://caryjensen.blogspot.com/2013/02/delphi-editor-key-combination-table.html>



Les bases du Pascal



Les bases du Pascal

- Le langage Pascal a une syntaxe stricte.
- C'est un langage fortement typé.
- Il est composé de moins de 100 instructions, le reste est fourni par des librairies ou ce qu'on développe nous-mêmes.
- Les librairies de Delphi communes à tous les types de projets sont regroupées sous le nom RTL (runtime library).



Les bases du Pascal

- Chaque ligne de code doit être séparé de la suivante par un point virgule.
- On peut regrouper plusieurs lignes de code entre un « begin » et un « end ». On appelle ça un bloc. C'est considéré comme une seule ligne de code par le compilateur.
- La dernière ligne de code avant un « end » peut se terminer par un point virgule. Pas obligatoire, mais prenez l'habitude.



Les bases du Pascal

- Le fichier principal d'un projet (son programme) commence par « program » et se termine par « end. »
- Les fichiers complémentaires (les unités) commencent par « unit » et se terminent aussi par « end. »
- S'il y a quelque chose après c'est ignoré.
- Ne vous affolez pas, Delphi fait une grande partie du boulot lui-même. Dans la plupart des cas on n'a pas à se préoccuper de la structure des fichiers.



Les bases du Pascal

- Les programmes n'ont aucune raison d'être s'ils ne font rien. On les crée pour manipuler des données ou interagir avec leurs utilisateurs.
- Les données sont stockées en mémoire (dans des variables ou les propriétés des composants) ou dans des fichiers (quelle qu'en soit le format).
- Chaque donnée utilisée dans le Pascal doit avoir un type et être déclarée avant d'être utilisée.



Les bases du Pascal

- En informatique il y a des types de données classiques :
 - Les booléens pouvant prendre la valeur 0, 1 ou -1, True ou False selon les langages.
 - Les entiers pouvant être positifs ou négatifs et prenant des valeurs numériques.
 - Les décimaux pouvant être positifs ou négatifs et prenant des nombres à virgule.
 - Les chaînes de caractères qui correspondent à des ensembles de lettres, chiffres et symboles divers.
 - Les tableaux qui sont des ensembles de données. Ces données devant être à leur tour d'un type défini.



Les bases du Pascal

- On les retrouve dans tous les langages avec des façons de les écrire et de les déclarer (lorsque c'est nécessaire) propres à chaque langage.
- Pour déclarer des variables dans Delphi on passe par un bloc de déclaration qui commence par « var ». On liste ensuite les noms des variables que l'on veut utiliser et leur type.



Les bases du Pascal

- Pour changer la valeur d'une variable on lui affecte une constante (un nombre, un texte, un booléen) ou une autre variable avec « := » (dans les autres langages on utilise généralement un « = »).
- Les propriétés des composants sont accessibles dans le code source (plus ou moins) comme n'importe quelle variable. En revanche comme elles sont liées à un élément particulier, on doit indiquer de quel composant on parle.
- Par exemple « `Edit1.Text := 'coucou';` » pour modifier la propriété Text du composant (dont le « Name » est) Edit1.



Parlons algorithmes



Qu'est-ce qu'un algorithme ?

- Définition prise sur Wikipedia :
« Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre une classe de problèmes. Le mot algorithme vient du nom d'un mathématicien perse du IX^e siècle, Al-Khwârizmî. »
- Un algorithme est la base de travail de tout développeur de logiciels. C'est la description de ce que va faire le programme, comment il va le faire et donc ce que l'on doit coder.



Exemple d'algorithme (simple)

- Prenons l'exemple d'un programme avec deux boutons et un libellé à l'écran. Le libellé est à 0 par défaut. Si on clique sur le premier bouton on l'incrémente (+1), si on clique sur le second on le décrémente (-1).
- Nous sommes en programmation événementielle, donc l'algorithme du programme est géré par Delphi. En revanche c'est à nous de faire l'algorithme qui correspond aux clics sur les boutons.



Exemple d'algorithme (simple)

Action ClicSurBoutonPlus

- stocker valeur du libellé de l'écran dans X

- ajouter 1 à X

- mettre la valeur de X dans le libellé de l'écran

Fin Action

Action ClicSurBoutonPlus

- ajouter 1 à la valeur du libellé à l'écran

Fin Action



Exemple d'algorithme (simple)

Action ClicSurBoutonMoins

- stocker valeur du libellé de l'écran dans X

- retirer 1 à X

- mettre la valeur de X dans le libellé de l'écran

Fin Action

Action ClicSurBoutonMoins

- retirer 1 à la valeur du libellé à l'écran

Fin Action



Qu'est-ce qu'un algorithme ?

- Le truc c'est simplement de décrire en français ce qu'on doit faire et de le dire de la façon la plus détaillée possible.
- Au début je vous recommande d'écrire sur papier ou un traitement de texte ce que vous voulez faire, ça vous obligera à y réfléchir avant de coder n'importe quoi. C'est une façon d'organiser vos idées qui vous fera gagner du temps.
- Sur les trucs simples et avec de la pratique vous n'en aurez plus besoin.



Qu'est-ce qu'un algorithme ?

- L'avantage des algorithmes, c'est qu'ils ne dépendent pas du langage de développement. Ils doivent pouvoir être compris par n'importe qui.
- Si ce n'est pas clair, ça ne sera pas joli à voir une fois codé et vous (ou quelqu'un d'autre) aurez du mal à revenir dessus plus tard.



Qu'est-ce qu'un algorithme ?

- Quelles sont les opérations et instructions disponibles dans un algorithme ?
 - Des verbes décrivant ce que vous voulez faire de la façon la plus précise possible (et quand je dis « précis », c'est surtout « fin » ou « bas niveau » qu'il faut comprendre).
 - Des tests permettant de faire quelque chose dans un cas et autre chose dans l'autre.
 - Des boucles permettant de répéter plusieurs fois la même chose. On en sort soit au bout d'un certain nombre d'itérations, soit sur un test.
 - Des modules nommés et des appels de modules par leur nom.



Et pour le Pascal ?

- En Pascal ça donne :
 - Des appels de procédures, de fonctions ou de méthodes (qui sont aussi des procédures et des fonctions mais attachées à des objets).
 - Les instructions de tests
 - « if ... then ... else ... »
 - « case »
 - Les instructions de boucles
 - « for ... to ... do »
 - « for ... downto ... do »
 - « for ... in ... do »
 - « repeat ... until (...) »
 - « while (...) do »



Un peu de pratique



Le classique « Hello World »

- Prenons un projet VCL vierge.
- Ajoutons un bouton sur la fiche.
- Activons l'événement onClic de ce bouton.
- Affichons « Hello world » avec une boite de dialogue en utilisant ShowMessage()



Remplir une liste de textes

- Prenons un projet VCL vierge.
- Ajoutons un bouton sur la fiche.
- Ajoutons un champ de saisie sur la fiche.
- Ajoutons une liste sur la fiche.
- Activons l'événement onClic de ce bouton.
- Lorsqu'on clique sur le bouton on ajoute la valeur du champ de saisie dans la liste.



Déplacer quelque chose à l'écran

- Prenons un projet VCL vierge.
- Ajoutons quatre boutons sur la fiche.
- Ajoutons un TPanel sur la fiche.
- Ajoutons un libellé dans le TPanel.
- Chaque bouton devra faire bouger le texte dans une direction différente sans qu'il ne disparaisse.



Deviner une valeur

- Prenons un projet VCL vierge.
- Ajoutons une zone d'onglet avec deux pages.
- Ajoutons un bouton sur la page du premier onglet.
- Ajoutons un champ de saisie, un bouton et un libellé sur le second onglet.
- Sur le bouton du premier onglet on bascule l'affichage vers le second et on stocke une valeur aléatoire entre 1 et 100.
- Sur le second bouton on vérifie si on a trouvé la valeur et on affiche la réponse sous forme de boîte de dialogue.



Exercice 3

- Aujourd'hui je vous propose de faire un peu de codage.
- Reprenez les exemples que je vous ai proposés et refaites les.
- Vous pouvez aussi faire des programmes classiques :
 - Une calculatrice (pas aussi simple qu'on ne le pense)
 - Un jeu de type « pendu » ou « roue de la fortune »
 - Un jeu de type « Mastermind » ou « Mot de passe »



A suivre...

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

Discutez sur les réseaux sociaux avec le hashtag
#DelphiBootCamp

