

【G6】ライトニングトーク

5分でわかる

【A1】Delphiテクニカルセッション  
「Delphi 2009 Handbook以降の  
Delphi言語新機能」

東洋テクニカルシステム株式会社

システム開発部

福士 光



# A1テクニカルセッション

- 朝の早い時間からおいで頂いてありがとうございました。
- A1セッションの感想(想像)
  - 長げーよ！(セッションそのものもプレゼン資料も)
  - 途中で居眠りしちゃったんで記憶が途中で抜けてるんだけど
  - 今北産業
  - 早く本くれ (すいません、抽選みたいです)

# それでは

- 独断と偏見に基づいて、A1の内容を5分でおさらいします。

# Unicodeの全面的な採用

- 文字列はUnicodeに置き換えてString型で扱きましょう。
- Shift\_JISを前提とした文字列処理は全て作り直すつもりで。
- AnsiStringはあくまでレガシー。RTLは基本的にUnicode (UTF-16)が前提になっています。
- 文字列でないものはTBytesに置き換えましょう。
  
- あとはこちらをご覧ください。
  - RAD Studioマイグレーションセンター
    - <http://www.embarcadero.com/jp/rad-in-action/migration-upgrade-center>
  - デベロッパーキャンプ・アーカイブ - 開催順
    - <http://www.embarcadero.com/jp/developer-camp-japan/archive>



# ジェネリックス

- まずは既存のコードのTList(SysUtilsユニット)や TObjectList(Contrrsユニット)を[Generics.Collections](#) ユニットにある[同名のコンテナで置き換え](#)てみましょう。なにより型安全です。
- C++のテンプレートのような[「何でもあり」](#)はいまのところ[無理](#)です。
- [型パラメータにはクラス型](#)を使う、と考えたほうがよさそうです。
  - 特殊化(specialization)の機能がないので基本型やレコード型に適用するには不十分だったりすることも。

『特に、型パラメータをInteger、Double、Stringなどの特定の単純型に制限することはできません。さまざまな型(単純型を含む)を表現する必要がある場合は、必要な型ごとにオーバーロード関数を使用することを検討するか、System.Rttiユニットに定義されているTValueなどのレコード型を使用します。』(ヘルプより)

# 無名メソッド

- 超便利です。まずは使ってみましょう。
  - いわゆるコールバック(内部で使用する処理を呼び出し元で用意する)は無名メソッドで実装すると呼び出しが簡単になります。
  - マルチスレッドやスレッド間同期といったことにも使えます。
- パラメータ並びが長くなるとか、セミコロンの付きかたが他と違って気持ち悪いとか、そんなことは慣れです。

# RTLの強化 (Delphi 2009)

- 普通はInteger(符号あり)かCardinal(符号なし)を使います。
- サイズが決まっている整数はCライクな型名が便利です。

新しい型名	元になる型	
Int8	ShortInt	8bit 符号あり
Int16	SmallInt	16bit 符号あり
Int32	LongInt	32bit 符号あり
Int64		64bit 符号あり
UInt8	Byte	8bit 符号なし
UInt16	Word	16bit 符号なし
UInt32	LongWord	32bit 符号なし
UInt64		64bit 符号なし
NativeInt		32bit/64bit 符号あり
NativeUInt		32bit/64bit 符号なし

- 例外を捕捉して投げなおすときは例外チェーンを使いましょう。

# 拡張RTTIと属性

- メタプログラミングで手を抜くなら拡張RTTIで。
  - 多くのクラスに同じような実装を要求されているなら考慮の余地あります。
- そうでなければ... 引き出しの奥にしまっておいてください。
  - いつか役に立つ日が来るかも...



# コンパイラの変更 (Delphi 2010/XE/XE2)

- 必ず呼び出されなければならない場合を除いて、initialization/finalizationはクラスコンストラクタ、クラスデストラクタに置き換えましょう。
  - そのクラスを実際に使っている場合だけリンクされます。
  - ただしどこかで直接そのクラス型を使わないとリンクに削除されてしまいます。
- delayed指令による遅延ロードはハードウェアを扱う場合などのインタフェースDLLや、OSのバージョンによるDLLの存在の差異を吸収するのに適しています。
- {\$SCOPEENUMS ON}コンパイラ指令で列挙型につけるプレフィックスが被って悩むことがなくなります。

# RTLの強化 (Delphi 2010/XE/XE2)

- 便利なものがたくさんできました。しかし機能的に不足しているものもありますので、ヘルプを見つつ、使えるところから使う、ということ。
  - VCL/Windowsの開発なら無理して使わなくても...
  - FMX/マルチプラットフォームなら、まず新しいものを見てみるのいいかも。

# Additional time

- クラスをハックする必要があるならクラスヘルパで。
  - 従来の手法(派生クラスを作る、ユニットをプロジェクトの場所にコピーして変更、同名クラスを定義、など)よりも簡単で、それでいてやりたい放題です。
- 既存の手を入れられない(高度でない)レコード型に高度なレコード型のような機能を付加するならレコードヘルパで。
- 仮想関数テーブル(VMT)を個別に書き換えるならTVirtualMethodInterceptorを使います。
- x64でSingle型を多用するときは{\$EXCESSPRECISION OFF}コンパイラ指令のことを思い出してください。

ありがとうございました

