

Pinkuma Humstar Presents

**COBOLerのための
Net Framework
プログラミング
その前哨戦 編**



わんくま同盟 東京勉強会 #20 [pinkumaDay]

(飼い主の) 自己紹介

・ 片桐です

- MVS、CVS、ZOS、AS400、RS6000、HP-UX、VAX、LINUX、DOS、Windows、OS/2、etc かなりカオスなプログラマ
人生20年と一ヶ月
- 一年前までMVSのCOBOLer兼AIXのANSI-C&KShell er兼Office97シリーズVBAぷろぐらま
- 現在、.Net Framework系全般アプリ (VB、C#、ASP、WCF、SQLCLR、PowerShell、LINQ etc) をまとめて試用&メンテに駆り出されている人柱プログラマ (笑)
 - ・ VBとCOBOLは付き合いが長いらしい
 - 我がぷろぐらま人生に一片の悔いなし! (バグはあるけどw)



今回のお話は……

- ・ **メインフレームでCOBOLをやってきた人、JCLをバリバリ書いて来た人、カタフロのカスタマイズとかやっていた人が、ある日ある朝突然に、Net Frameworkのプログラミング世界へやってきた時、頭の隅に置いておくところとちょっと入りやすくなるかもしれない、話**
 - **アレルギーの予防接種に効くかもしれない**



その前に……

- ・ **COBOLerたちへの大きな誤解**

- **COBOLerがVBやC#と、Net Frameworkアプリケーション開発にアレルギーを起こすのは文法の問題じゃない**

- **開発手法とアーキテクチャが変わりすぎて何をどうとっかかれば「WATER!」となるのか判らないだけ**

- ・ **なまじ長く同じ環境で生きてきているだけに、この変化にどう対応すればこの先生き残れるのかわかんないだけ**



Agenda

- ・ **COBOLも、Net Frameworkプログラミングも同じだ**
- ・ **COBOLerを阻むものたちも実は○○だった**
- ・ **戦え！COBOLer**
- ・ **参考文献**



COBOLerから言いたい事

- ・ クラスとかメソッドとかプロパティとか、用語多すぎ。
- ・ 命令多すぎ。
- ・ 覚える事多すぎ。
- ・ フロントエンド多すぎ。
- わかなくて多すぎ。
- なんか、画面作り多すぎ。
- リストに出させない！なんでもこの枚数は！

ええい！鎮まれえい！
人の話を聞けえい！
それをやらなきゃならなくなったんだろうが！

う？

やがっ



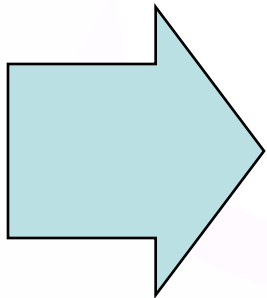
COBOLも、Net Frameworkプログラミングも同じだ

・ ちょっと視点をかえてみませう

**- そもそも、プログラムというものはソース
作ってコンパイルして動かします**

**・ COBOLでは、作成したプログラムをJCLに組み込み、
一連の動きをします**

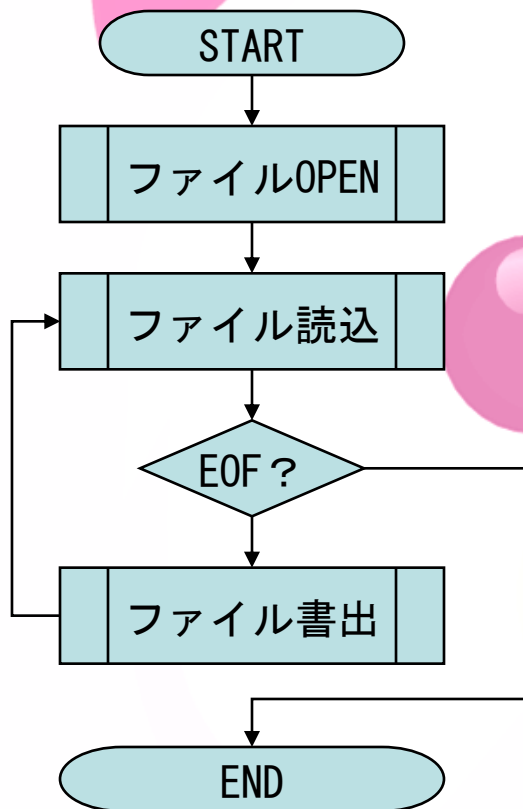
**・ 同じようにVBやC#も、Net Frameworkというライフ
ラリを組み込んでコンパイルして動かします**



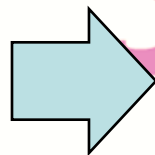
まずは、わかりやすいところで
よくあるバッチプログラムパターンで見てください



ファイルを読み込んで書き出す処理

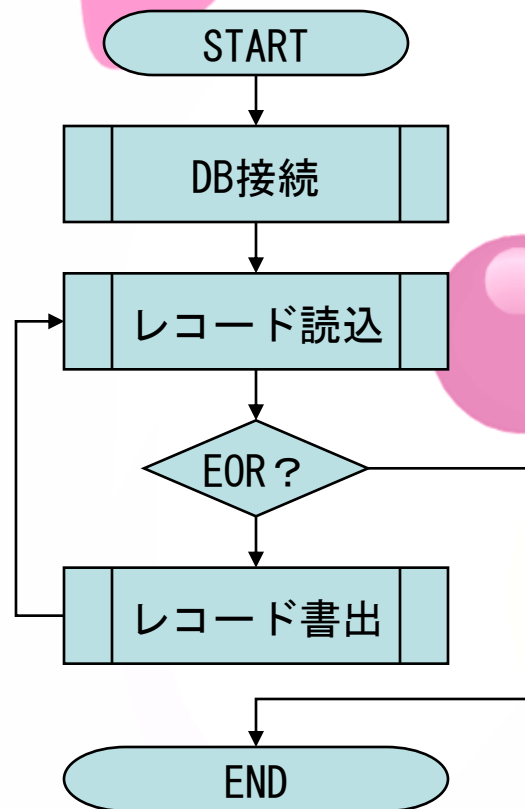


このチャートから、COBOLの処理プログラムが思い浮かびますよね？
同じように、VBやC#でも同じ動きをそれぞれのプログラムで書きあらわす事ができます。

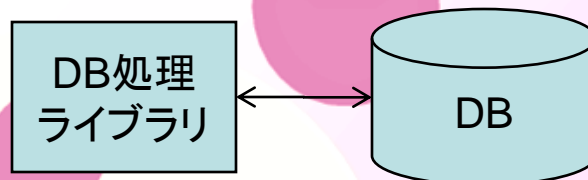


FILEのI/Oでは基本は同じなんです

DBを読み込んで書き出す処理



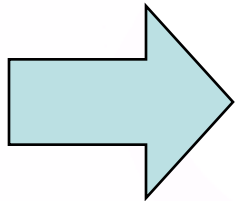
DB処理を行う際にはプリコンパイラ処理 (EXEC-SQL) を埋め込むことが多いと思います。実際にはそれによって、コンパイルされた後のCOBOLモジュールにはDB処理のライブラリを使うように組み込まれます



実は、このDB処理ライブラリがNet Frameworkでも同様に提供されていてモジュールに組み込む形になっています。

バッチプログラムパターン で見る二つの差

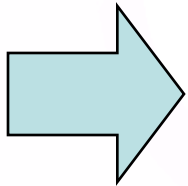
- ・ フローチャートや図で見ても、実は大差ないんです
 - COBOL + JCLによって作られていたものと、(VBやC#) + .Net Frameworkによって作られていたものも、図にすれば同じになります



では、どうして、簡単にいかないのでしょうか？

阻む敵、その代表はクラス

- ・ **COBOLには無い用語であり概念、それがこの「クラス」です**
 - **実はこの「クラス」こそ、.Net Frameworkの正体であり、最初に攻略するべき存在であり、多くのCOBOLerを挫折に追いやってきた強敵でもあります**



けれど、実は、私達COBOLerはすでにクラスの概念に近いとあるものをJCL上で使いこなしています。
一連の処理をカプセル化しておき、必要に応じてJCLに記載して使用し、時には与えるパラメータによって展開されるプログラムやJCLまで変えてしまえるすぐれものの機能。そう、あいつです。

概念は同じ JCLプロセスとクラス

- ・ JCLを作成する際に、避けて通れないのがJCLプロセス、そしてカタログプロセス。
 - 処理をカプセル化し、必要に応じて適切なパラメータと共に呼び出して一連のプログラムを実行します
 - ・ Net Frameworkプログラミングで使用される「クラス」の概念もまた、まさしく一連の処理、一連の小さなプログラムの概念をカプセル化して公開し、使用できるようにしたものです
- COBOLerは.Net FrameworkやVB、C#で使われる関数処理の中身、つまりクラスの中の関数処理、手続き処理をプログラミングしてきた人たち、と言えるのです



クラスの命名って実は……

- ・ **クラスの命名は実はシンプルです。**

- 「分類名」. 「分類名」. 「分類名」

- ・ **どこかで見覚えありませんか？メインフレームJCLでのDSN命名規則とそっくり！**

- 「クラス名. メソッド」と「データセット名. フロシージャ名」は見た目、同じに見えるし、実際、概念として考えると同じなんです

- » Namespace. Class. ReadMasterFile というメソッドを Namespace. ClassデータセットのReadMaseterFileフロシジャ、と読み替えても違和感がありません

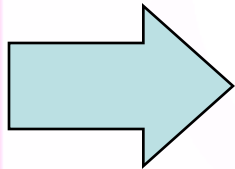
- » メインフレーム上でのフロシジャライフライも、用途によってDSNを切り替えて同じフロシジャで使い分けたりもします→これ、Overrideの概念に近いです



つまり

. Net Framework7° の「ラミング」とは

- ・ 今まで自分達がCOBOLで作ってきたプログラム、関数、プロシージャを「クラス」という単位でまとめ、外部から処理されるサブルーチンを公開して提供できるようにし、それを必要に応じて呼び出しながら処理するプログラムを作る、ということです。（ランボーに言うとおね）



COBOLerの視点で.Net Frameworkプログラミングを考えると、それは言うなれば、必要なプロシージャをライブラリから探し出して使い、無ければ作り、それらを組合せ、まさしくJCLを作り上げていくということだったんです。

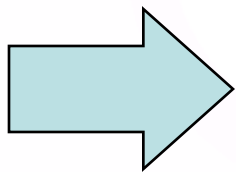


ぴんくま
同盟

わんくま同盟 東京勉強会 #20 [ぴんくまDay]

次の敵、 Formアプリケーション

- ・ **実はPCを使う以上、Formアプリは必ず触れているものだったりします。**
 - **そこでちょっと目先を変えてみると？**
 - ・ **Formアプリケーションは、実はイベント単位で動きます**
 - **つまり、JCLがキックされるタイミングのバリエーションが増え、そのタイミングごとに必要なJCLを作成する、と考えます。**



.Net Frameworkプログラミングになると、JCLのキックタイミングが異なるだけです。実際に動作する一連の処理をフローチャート化することで、戸惑わなくなります。粒度が異なるだけで、一連の処理の単位にしてしまえば、大差はないのです。



次の敵、WEBアプリケーション

- ・ **WEBアプリケーションもいわばバッチプログラムです。**
 - ブラウザからページ表示を要求するとサーバーがバッチ処理を起動し、処理終了後にアウトプットをクライアントに返すようにしてくれます
 - ・ つまり、JCLのキックされるタイミングがWEBサーバーにあるだけで、やはり作成されるJCLは今までと変わらない粒度で考える事ができます

大丈夫、まだ頑張れる

- ・ **COBOLは手続き型言語の基本であり、古くから存在しつつもその言語体系の変遷がほとんど行われてこなかった、生きた化石ともいえる言語です。けれど、だからこそ、「プログラミング概念」の基礎が詰まっています。**
 - それらを知る人が新たな世界を知る時、この基礎が決して廃れる事のないものの一つなのだと気づくと思います。
 - ・ **基本は、変わらないのです、突き詰めれば、全ては0と1の機械語なんです**

戦え！COBOLer

- ・ **概念を理解すれば、後は文法の問題だけになります。**
 - **はじめてCOBOLに触れたとき、どうやって覚えていきましたか？それと同じやり方でも良いのです。**
 - ・ **一つ、使いこなせる言語を持っていれば、その言語でできたことが、新しい言語でどうなるのか、を知っていく事できっと新しい世界が見えてきます**
 - **俺達の戦いはこれからだ！**
 - 》 **COBOLerさんの次回作にご期待ください**



参考文献&サイト

- ・ **片桐の胡桃の脳内 (IBM3090/MVS)**
- ・ **コボル開発者のためのVisual Basic入門講座 ~導入編~**
 - <http://www.int21.co.jp/pcdn/vb/noriolib/vbmag/9806/cobol/>
- ・ **COBOLer的VB.NET (文法比較表)**
 - <http://go2vb.cocolog-nifty.com/blog/>

いぢよ

