

今からでも遅くない!

RPG IIIユーザーに贈る RPG IVの魅力と可能性

第4回 RPG IIIからRPG IVへの移行の実際

中村 潤株式会社アイ・ラーニング
IT 研修本部 IBM 製品研修部
ラーニング・アドバイザー

次世代に、RPG IIIのまま 引き継ぎますか?

この連載も、早いもので4回目となります。これまでの各回でRPG IVのよさをお伝えしてきましたが、そのことを言い換えれば、RPG IIIを捨ててRPG IVに乗り換えましょう、というメッセージにはかなりません。

しかしながら現実問題として、私たちアイ・ラーニングでは、RPG IVだけでなくRPG IIIのコースもご提供しています。これはひとえにお客様からのご要望があるからで、まだ多くのRPG IIIユーザーがおられるのが実情です。

RPG III、あるいはRPG IVのプログラムであっても、IBM i上では*PGMのオブジェクトであり、CALLで実行可能なことに変わりはありません。プログラム作成のプロセスと管理方法に違いはありませんから、リプレースしても書き換えの必要は一切ありません。それがIBM iのよさであり、RPG IIIにとどまる大きな要因であるとも言えましょう。RPG IIIのユーザーは、IBM iの「変えなくてよい」というよさを享受されている方々、と言えるかと思えます。

ここ数年、IBM i関連の研修コースを担当していて頻繁に耳にするのは、「他部門から情報システム部門に異動になり、RPGプログラムの保守を担当することになった」や「これまでオープン系システムを担当してきたが、配置換えでIBM iの担当になった」という声です。

この背景には、S/38やAS/400時代からの古参のRPG技術者の引退があることは言うまでもありません。また、一度はIBM iを離れて他のシステムへ移ってみたものの、

IBM iが一番使いやすいことをあらためて認識し戻ってこられた企業も少なくないと見えています。

そうした世代交代やシステムの切り替えが発端となって研修のお客が増えることは、研修会社としては非常にありがたいお話ですが、その一方で、以前のRPG IIIをそのまま引き継ぐのでよいのだろうか、という疑問を抱かずにはいられません。なぜなら、RPG IVが優れているということだけでなく、若い世代にはRPG IVのほうが圧倒的に理解しやすいからです。

RPG IVには、RPG IIIにない新しい演算命令が加わっています。そして、自由形式のコーディングが可能で、JavaやVBを使ってきたプログラマーにはまったく違和感のないプログラミング・スタイルと言えます。RPGユーザーは今、RPG IIIにとどまるかRPG IVへ移行するかの、判断の岐路に立たされているのです。

RPG IIIからRPG IVへ まず移行してみる

RPG IIIからRPG IVへの、ソースの変換自体は非常に簡単です。CVTRPGSRCコマンドを使えば、RPG III資産をすぐRPG IVに変換できます。ここまでは非常に簡単なのですが、ソースをコンパイルした後のプログラムのリプレースは、そう簡単にはいきません。そしてシステムテストの必要も出てきます。

しかし前述のように、RPG III、あるいはRPG IVのどちらのプログラムであっても、*RPGのオブジェクトであるこ

RPG IVの魅力と可能性

とに変わりはありません。それゆえプログラムに2つの言語が混在しても問題ありません。

そこで、プログラムの仕様変更が発生したら、まずRPG IVで書いてみるという方法もあるかと思います。少々強引なやり方ですが、こういう形でRPG IVのよさを実感することも少なくないと思えます。何よりも、第一歩を踏み出すことが大切です。

では、仕様変更となったプログラムを、CVTRPGSRCで変換してみましょう(リスト1)。この変換にかけるだけでも、大きな前進です。変換したプログラムは従来のRPG IIIとさほど変わらないように見えますが、スペルが変化

している演算命令があります。次が代表的な例です。

EXCPT → EXCEPT
 LOKUP → LOOKUP
 SELEC → SELECT
 WHEQ → WHENEQ
 UPDAT → UPDATE

省略形だった演算命令がふつうのスペルに戻されているので、受け入れやすくなっています(DSPLYだけ、どうしてDISPLAYにならないのか不思議ですが……)。RPG

リスト1 CVTRPGSRCによる変換

```
CVTRPGSRC FROMFILE (KAIHATSU/QRPGSRC) FROMMBR (HIN010)
TOFILE (KAIHATSU/QRPGLESRC) TOMBR (*FROMMBR)
```

リスト2 RPG IIIプログラム

```
CL0N01N02N03FACTOR1+++OPCDEFACOR2+++RESULTLENDHHILOEQCOMMENTS+++++++
C          *IN03      DOWEQ*OFF
C          WRITERCD03
C          EXFMTRCD01
C          *IN03      IFEQ *ON
C          MOVEL*ON          *INLR
C          RETRN
C          ELSE
C          *IN13      IFEQ *ON
C          MOVEL*ON          *INH1
C          RETRN
C          ELSE
C          RETRN
C          ENDIF
C          ENDIF
```

リスト3 リスト2をRPG IVに変換

```
CL0N01FACTOR1+++++++OPCODE&EXTFACTOR2+++++++RESULT+++++++LEN++D+HHILOEQ
C          WRITE      RCD03
C          EXFMT      RCD01
C          *IN03      IFEQ      *ON
C          MOVEL      *ON          *INLR
C          RETURN
C          ELSE
C          *IN13      IFEQ      *ON
C          MOVEL      *ON          *INH1
C          RETURN
C          ELSE
C          RETURN
C          ENDIF
C          ENDIF
```

ⅢとRPGⅣは全然違う、と思い込んでいる方が時々おられますが、RPGⅢとRPGⅣにさほど大きな違いがないことがわかりいただけると思います。

次に、RPGⅣを「らしく」変更してみましょう。ここからがRPGⅣへの「変身」です。

たとえば **リスト2** を変換したとします。これは、プログラムHIN010のRPGⅢの演算仕様書の部分だけを抜粋したものです。**リスト3** は、それをCVTRPGSRCコマンドでRPGⅣに変換した状態です。

ご覧のように、各記入項目の桁数が多くなったこと、条件付けの標識が1つになったこと、RPGⅢのRETRN命令がRETURNになっているなど変化はありますが、見た目は以前のRPGⅢプログラムとほとんど変わっていません。記入項目の位置が変わるパターンは演算仕様書だけでなく、ほかの仕様書も同様ですが、RPGⅢとRPGⅣはまったく違うという懸念は不要であることがわかりいただけ

たと思います。

よりRPGⅣらしく 変えてみる

ほかのプログラミング言語とRPGを対比して一番違和感があるのが、IF、DOなどの構造化命令で、RPGで演算項目1を使わざるを得ないのが大きなネックです。命令の左側にパラメータがあるのは見やすいプログラムとは言えません。演算項目1には気の毒ですが、ここは演算項目2を使わないスタイルに変更しましょう。RPGⅣのIF命令、DO命令は、拡張演算項目2が使用でき、従来の演算項目2から結果のフィールド、結果の標識までを式の記入欄にできます(**リスト4**)。

この変更だけでも、今までのRPGとはずいぶん変わった感じがするのではないのでしょうか。拡張演算項目2を使

リスト4 拡張演算項目2でIF命令を使用

```
CL0N01FACTOR1+++++++OPCODE&EXTEXTENDED-FACTOR2+++++++  
C          WRITE      RCD03  
C          EXFMT      RCD01  
C          IF          *IN03  
C          MOVEL      *ON          *INLR  
C          RETURN  
C          ELSE  
C          IF          *IN13  
C          MOVEL      *ON          *INH1  
C          RETURN  
C          ELSE  
C          RETURN  
C          ENDIF  
C          ENDIF  
C*
```

リスト5 ELSEIFを使用

```
CL0N01FACTOR1+++++++OPCODE&EXTEXTENDED-FACTOR2+++++++C  
C          WRITE      RCD03  
C          EXFMT      RCD01  
C          IF          *IN03  
C          MOVEL      *ON          *INLR  
C          RETURN  
C          ELSEIF     *IN13  
C          MOVEL      *ON          *INH1  
C          RETURN  
C          ELSE  
C          RETURN  
C          ENDIF
```

RPG IVの魅力と可能性

うと、プログラムが以前よりも見やすくなったことが実感できると思います。段階的に変更していけば違和感なさほどありませんし、無理に最初から今までとまったく違う形にする必要はありません。

拡張演算項目2を使えるようになったらぜひ使ってみるのが、ELSEIFです。ほかの言語には当たり前のようにあるELSEIFが、RPG IIIにないからです。ELSEIFを使うと、**リスト5**のようになります。

ELSEIFに対するENDIFは不要です。ELSEIFを使用することによって、入れ子がよりわかりやすくなります。さらにフリーフォーマットを使用すると、**リスト6**のよう

リスト6 拡張演算項目2でIF命令を使用

```

/FREE
    WRITE      RCD03;
    EXFMT      RCD01;

    IF         *IN03;

        *INLR =  *ON;
        RETURN;

    ELSEIF    *IN13;

        *INH1 =  *ON;
        RETURN;

    ELSE;

        RETURN;

    ENDIF;
/END-FREE

```

に見違えるほどのスタイルになります。

自由形式だと、定位置記入項目と違って入れ子が明確になります。またRPG IIIでは空白行を入れられないので、ステートメントの7桁目に*を入れて、コメント行を空白行にしていたのですが、RPG IVでは空白行の挿入が可能です。

また、指標を使って繰り返しを制御する命令としてDO命令を使用してきましたが、RPG IVではFOR命令が加わっています。これも演算項目1を使うと今までのDOと同じですが、拡張演算項目2を使うと、RPGと違うようにも見えます(**リスト7**)。

今まで他の言語と比べて違和感のあったこうした部分を埋め合わせていくだけでも、今後の保守作業に大きく影響を及ぼすはずです。

まとめ

本稿の目的は、RPGをほかの言語に追従させようというわけでは決してありません。しかしRPGプログラマーが世代交代しつつある現在、次のRPGプログラマーのためにRPGを受け入れやすくする準備は必要でしょう。フリーフォームのRPGプログラミングは、オープン系システムのプログラマーにも受け入れやすく、習得しやすいものです。また従来のRPGプログラマーにとっても、フリーフォームのRPGがオープン系言語に対する違和感や苦手意識を取り払うきっかけになります。そうすればオフコンとして使ってきたIBM iが、オープン系システムのサーバーに生まれ変わる足がかりになるかもしれません。まさしく、AS/400がIBM iに変わるとき、なのです。📌

リスト7 ELSEIFを使用

```

CL0N01FACTOR1+++++++OPCODE&EXTFACTOR2+++++++RESULT+++++++
C      1                DO          100          X
C                ADD          DATA (X)        OUTPUT
C                ENDDO          1

↓

C                FOR          X = 1 TO 100 BY 1
C                ADD          DATA (X)        OUTPUT
C                ENDFOR

```