RPG
free forever

Marco Riva

www.markonetools.it

Ultimo aggiornamento: 4/12/2020

Free format: breve storia

- da V5R1 (mag-2001) per le specifiche C
- da V5R3 (giu-2004) per embedded SQL
- da 7.1 TR7 (nov-2013) per specifiche H, F, D, P
- da 7.1 TR11 / 7.2 TR3 (nov-2015) formato totalmente libero

https://www.markonetools.it/liberiamo-lrpg/

Free format forever

- Il formato libero è il PRESENTE e FUTURO dell'RPG.
- Scrivere codice in formato libero dopo qualche tempo di pratica è sicuramente
  - più veloce
  - più chiaro
  - più moderno

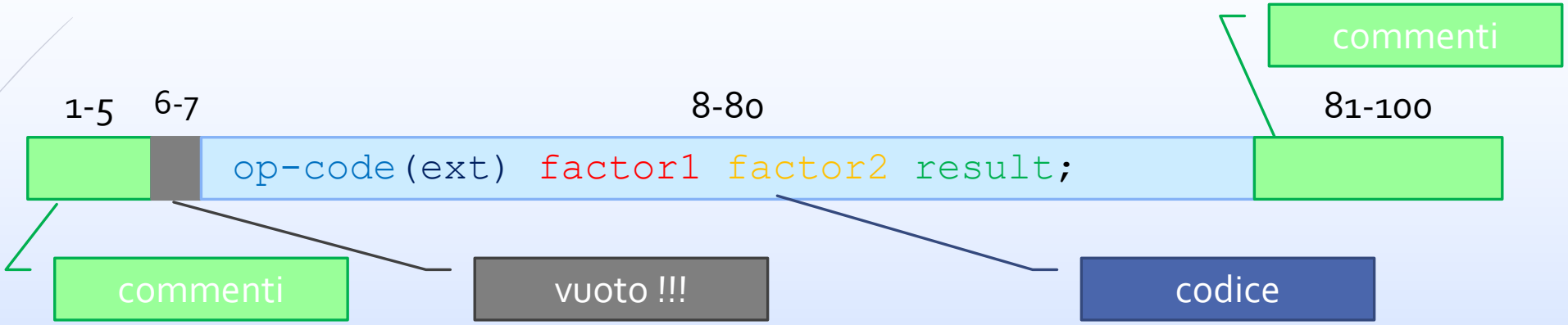
ma con uno strumento di sviluppo altrettanto moderno:
RDi

Perché usare il “free”

- sinteticità permessa dalle BIF e dalle espressioni
- la leggibilità: è consentito indentare, spaziare…
- più spazio
- alcune nuove caratteristiche sono disponibili sono nel formato libero
- esempi di codice nella letteratura web sono in rpg free
- più simile agli altri linguaggi di programmazione
- più facile da imparare per chi NON conosce l’RPG

la leggibilità è responsabilità del programmatore

Regole base



- ogni istruzione deve terminare con il carattere **;**
- codice operativo `eval` e `callp` sono facoltativi
- `/free` … `/end-free` **non** sono più necessari da 7.1 TR7

Altre regole importanti

- Si può scrivere un'**istruzione su più righe** senza nessun carattere di continuazione, perché la fine dell'istruzione è determinato dalla presenza del `;`
- **non esistono indicatori** risultanti e di condizionamento
- gli indicatori si referenziano con `*INxx`
- Il codice operativo CALL deve essere sostituito con `CALLP` e i parametri devono essere convertiti nella struttura dei prototipi
- le istruzioni SQL devono essere precedute dalla keyword `exec sql` e terminare anch'esse sempre con un `;`
- da `7.1` le **specifiche F e D possono essere in ordine misto**
  in questo modo è possibile raggruppare logicamente le definizioni di variabili correlate logicamente alle definizioni di file

Suggerimenti

- scrivere con «**stile**»
- evitare indicatori numerici e privilegiare le **variabili booleane**
- **indentare** il codice
- scrivere tutto in free, evitare un misto free e formato fisso
- usare le **parentesi** anche dove non necessarie per chiarire le priorità delle espressioni
- usare coerentemente **maiuscole e minuscole**

Massimo Duca, RPG & SQL: stile e produttività,

All-free / fully free

codice

1-240

```
op-code(ext) factor1 factor2 result;
```

- disponibile da IBM i 7.2 TR3 o 7.1 TR11
  In 7.1 non è supportata la compilazione su release precedenti.
- per indicare al compilatore che si sta utilizzando il formato totalmente libero bisogna scrivere ****FREE** oppure ****free** a partire dalla prima colonna
- dopo la keyword `**FREE` **non** è possibile scrivere codice a formato fisso
- con RPG fully free posso scrivere i membri sorgenti anche in file sorgenti con lunghezza superiore a 112

Carattere di continuazione per stringhe

- Simbolo **-**: la stringa continua con la **prima colonna** della riga successiva
- Simbolo **+**: la stringa continua con la **prima colonna non blank** della riga successiva

```
dcl-s Stringa char(40);

Stringa = '*** INIZIO PROGRAMMA ***';
dsply stringa;
Stringa = 'Prima riga -
     seconda riga (segno -)';
dsply stringa;
Stringa = 'Prima riga +
     seconda riga (segno +)';
dsply stringa;

*inlr = *on;
```

```
DSPLY  *** INIZIO PROGRAMMA ***
DSPLY  Prima riga      seconda riga (segno -)
DSPLY  Prima riga seconda riga (segno +)
```

Codici operativi "obsoleti"

- ATTENZIONE: non tutti i codici operativi possono essere scritti in formato libero: ADD, ANDxx, CABxx, **CALL**, CASxx, CAT, COMP, DEFINE, DIV, DOUxx, DOWxx, GOTO, IFxx, **MOVE**, **MOVEA**, **MOVEL**, MULT, MVR, ORxx, **PARM**, **PLIST**, SCAN, SETOFF, SETON, SUB, SUBST, TAG, TESTB, TESTN, **TIME**, WHENxx, Z-ADD, Z-SUB.
- I codici operativi non gestiti dal formato libero possono essere definiti "obsoleti" e "deprecabili"

come fare senza questi codici operativi?
segui i prossimi Power Coffee!

Definizioni «libere»

- spec. H: `ctl-opt`
  per scrivere una specifica H vuota digitare solo ctl-opt
- spec. F (file): `dcl-f`
- spec. D (prototipi): `dcl-pi`, `dcl-pr` … `end-pr`
- spec. D (costanti): `dcl-c`
- spec. D (variabili stand-alone): `dcl-s`
- spec. D (data structure): `dcl-ds` … `end-ds`
- spec. P (procedure): `dcl-proc` … `end-proc`

Definizioni «libere»/2

- nella definizione del tipo dati il numero di decimali se zero può essere omesso
- se non è necessario specificare i nomi dei sottocampi di una ds bisogna specificare **`*n`** dove invece nel formato fisso si poteva semplicemente lasciare vuoto
- è possibile utilizzare le costanti nella definizione di altre variabili. P.es.

```
dcl-c LenQta 13;
dcl-c DecQta 3;
dcl-s QtaOrdine packed(LenQta:DecQta);
```

Variabili "quasi uguali"

- Se si ha la necessità di definire una variabile tramite la keyword like ma si desidera modificare la lunghezza si può scrivere nel secondo parametro il valore di incremento/decremento:

```
dcl-s TotQta like(QtaOrdine:+2);
```

Dichiarazioni "condizionate"

- Le definizioni in formato libero possono contenere porzioni di definizione condizionate dalle direttive di compilazione. P.es.

```
dcl-s TotImporto
      /IF DEFINED(bignumber)
          packed(17:2)
      /ELSE
          packed(13:2)
      /ENDIF
;
```

DS esterne

- DS basata su una DS esterna è sempre opportuno **racchiudere tra apici il nome della DS esterna**:

```
dcl-ds MiaStruttura extname('NOMEDS') end-ds;
```

  - nel formato libero una **stringa non racchiusa tra apici** (unquoted) si riferisce sempre a un campo o a una variabile o a una costante.

- Da 6.1 è possibile **qualificare** il nome della DS con anche la libreria

```
dcl-ds MiaStruttura extname('DB2SAMPLE/EMPLOYEE') end-ds;
```

DS nidificate

- Da 7.2 TR6 e 7.3 TR2 è possibile definire le DS nidificate (nested) solo in formato libero

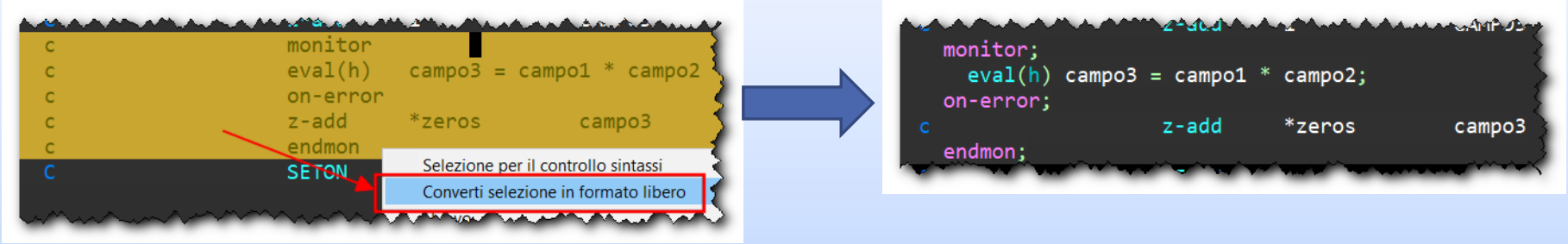
```
dcl-ds DSEsterna qualified;
  sottocampoE1 char(1);
  dcl-ds DSInterna;
    sottocampoI1 char(2);
  end-ds DSInterna;
end-ds;
```

release precedenti

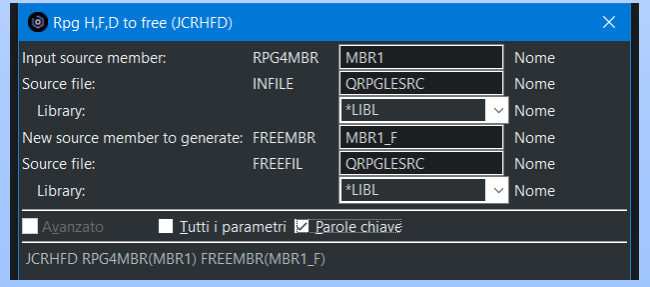
```
dcl-ds DSEsterna qualified;
  sottocampoE1 char(1);
  sottocampoDSInterna likeds(DSInterna);
end-ds;
dcl-ds DSInterna qualified;
  sottocampoI1 char(2);
end-ds;
```

Conversione

- è possibile convertire le specifiche C da formato fisso a formato libero con RDi (editor LPEX)
  - N.B. i codici operativi «obsoleti» non vengono convertiti



- Tool JCRHFD di Craig Rutledge http://www.jcrcmds.com/jcrdown2.html#JCRHFD_tag converte le specifiche H, F e D da fisso a formato libero



Limitazioni

- le specifiche I e O prevedono solo la sintassi a formato fisso
- il ciclo RPG deve essere scritto in formato fisso
- le schiere basate su tabelle (ovvero la keyword FROMFILE) non sono gestite nel formato libero delle specifiche D

Riferimenti

- E-mail aziendale: mriva@sirio-is.it
- Blog: www.markonetools.it
- Facebook: https://www.facebook.com/markonetools/
- YouTube: https://www.youtube.com/channel/UCb47YJQJCzU-5x4nnGzDu-w
- E-mail blog: info@markonetools.it
- Linkedin: www.linkedin.com/in/marcoriva-mk1