

RPG

CALL me, maybe...



Marco Riva



www.markonetools.it



Ultimo aggiornamento: 27/12/2020

So call me,
maybe?

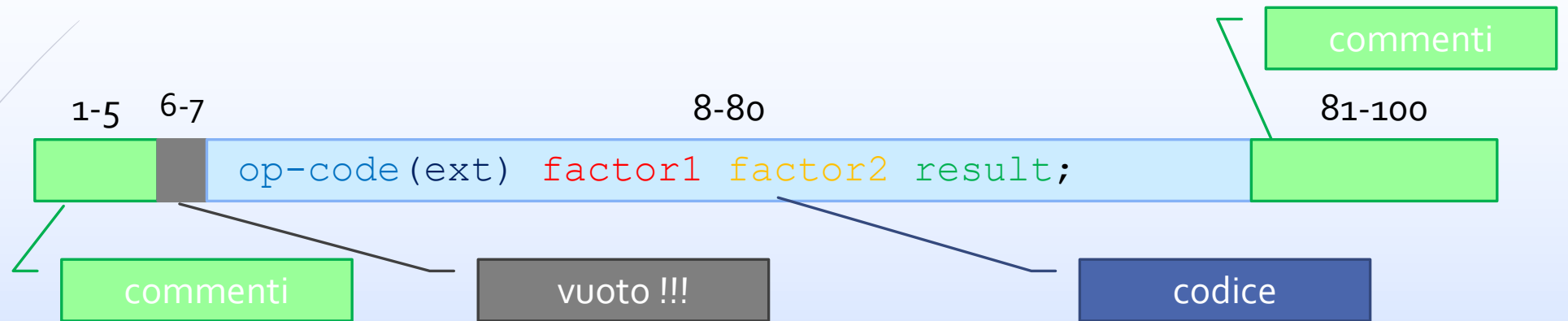
No, thanks!
callP is better

- ▶ come chiamare un programma senza CALL?
- ▶ esattamente cosa significa CALL**P**?
- ▶ e come dichiaro i parametri senza la *ENTRY PLIST?



2

Regole base



- ▶ ogni istruzione deve terminare con il carattere **;**
- ▶ `/free ... /end-free` **non** sono più necessari da **7.1 TR7**



3

Codici operativi "obsoleti"

- ▶ **ATTENZIONE:** non tutti i codici operativi possono essere scritti in formato libero: `ADD`, `ANDxx`, `CABxx`, **`CALL`**, **`CALLB`**, **`CASxx`**, `CAT`, `COMP`, `DEFINE`, `DIV`, `DOUxx`, `DOWxx`, `GOTO`, `IFxx`, `MOVE`, `MOVEA`, `MOVEL`, `MULT`, `MVR`, `ORxx`, **`PARM`**, **`PLIST`**, `SCAN`, `SETOFF`, `SETON`, `SQRT`, `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"

vediamo come sostituire i codici operativi per chiamare altri programmi/procedure



Cosa significa CALLP?

- ▶ **CALL** with **P**rototype
- ▶ NON significa CALL Procedure
- ▶ con CALLP posso chiamare sia le procedure sia i programmi
- ▶ per fare la chiamata c'è bisogno di un **prototipo**
- ▶ quindi con CALLP in formato libero posso sostituire la CALL di programmi del formato fisso

PARM



PROTOTIPO



5

https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_74/rzasd/zzcallp.htm#zzcallp

Chiamare un programma con CALLP

- ▶ non è necessario sostituire nel programma chiamato la definizione di *ENTRY PLIST con il prototipo e la procedure interface
- ▶ nel programma chiamante devo definire un prototipo che corrisponda alla *ENTRY PLIST del chiamato

PROGRAMMA CHIAMANTE

```
dcl-pr EsCP2 extpgm('ESCP2');  
  *n char(5);  
  *n packed(7:2);  
end-pr;  
  
callp EsCP2 (parm1:parm2);  
// oppure si può omettere callp  
EsCP2 (parm1:parm2);
```

PROGRAMMA CHIAMATO

c	*entry	plist		
c		parm	parm1	5
c		parm	parm2	7 2



6

*entry plist in formato libero

- ▶ il prototipo (*pr*) deve includere la keyword **export**
- ▶ la procedure interface (*pi*) deve avere lo stesso nome del prototipo
- ▶ il prototipo deve precedere la procedure interface
- ▶ se il nome del prototipo e della procedure interface sono diversi dal nome del modulo bisogna specificare il nome del modulo nella keyword export
export('nome_modulo')
- ▶ non possono essere passati parametri by value
- ▶ non può avere un return value



*entry plist in formato libero

PROGRAMMA CHIAMATO

c	*entry	plist		
c		parm	parm1	5
c		parm	parm2	7 2

equivale a

PROGRAMMA CHIAMATO «free»

```
dcl-pr EsCP2 extpgm('ESCP2');  
  *n char(5);  
  *n packed(7:2);  
end-pr;  
dcl-pi EsCP2;  
  parm1 char(5);  
  parm2 packed(7:2);  
end-pi;
```



Dove scrivere il prototipo?

- ▶ è una buona pratica scrivere i prototipi in membri sorgenti diversi da quello del programma chiamato e chiamante
- ▶ nei programmi vengono importati con la direttiva /COPY

SORGENTE PROTOTIPO

```
dcl-pr EsCP2 extpgm('ESCP2');  
  *n char(5);  
  *n packed(7:2);  
end-pr;
```

PROGRAMMA CHIAMATO

```
/copy escp_h  
dcl-pi EsCP2;  
  parm1 char(5);  
  parm2 packed(7:2);  
end-pi;
```

PROGRAMMA CHIAMANTE

```
/copy escp_h  
callp EsCP2(parm1:parm2);  
// oppure si può omettere callp  
EsCP2(parm1:parm2);
```



I sorgenti dei prototipi

- ▶ usualmente i sorgenti dei prototipi hanno suffisso « **H**»
- ▶ si possono inserire nel file **sorgente QRPGLSRC** così che nella direttiva /copy sia sufficiente specificare il nome del membro
- ▶ oppure si possono tenere in **file sorgente separato**: p.es. QPRTSRC
In questo caso nella direttiva /copy è necessario qualificare con il nome file il membro da includere



10

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

