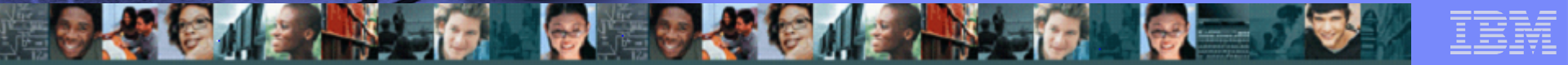


IBM Academic Initiative



Job Control Language



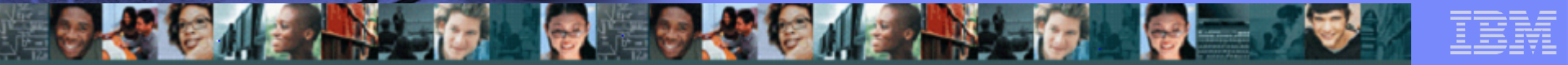


JCL

- **Introduzione**

Le JCL sono un mezzo di comunicazione tra un programma, scritto in Cobol, Assembler, PL/1..., e il sistema operativo z/OS.

Attraverso le JCL, si dichiara il programma da eseguire e si fornisce una descrizione dell'input e dell'output necessari al programma.

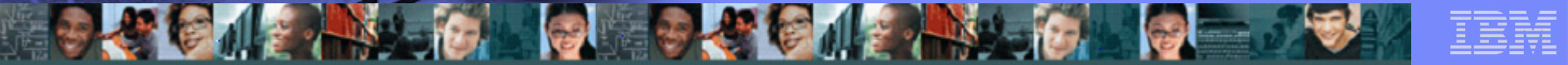


Origine delle JCL

Le JCL derivano dalle schede perforate, usate fino alla fine degli anni '70 per caricare lavori in macchina

Dalle schede hanno conservato alcune caratteristiche fisiche, ad esempio la lunghezza di 80 colonne.

1 riga JCL = 1 scheda



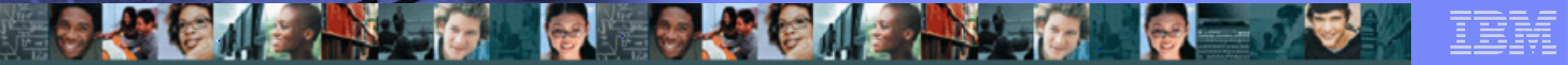
JCL statements

Le JCL sono composte di 3 statements:

- JOB
- EXEC
- DD

Ogni statement può essere composto, al massimo, di 5 tipi di *fields*:

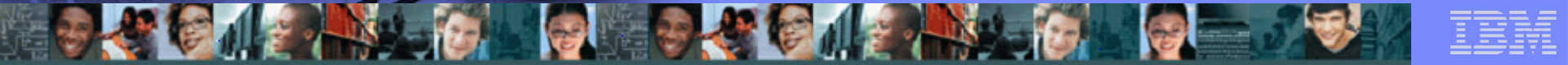
- //
- Name
- Operation
- Operand
- Commenti



JCL example

```
//JOB1      JOB (034D),'RAMESH',CLASS='A',PRTY=6  
//STEP01   EXEC PGM=COBPROG.  
//INFILE   DD DSN=SED.GLOB.DES.INFILE,DISP=SHR  
//OUTIFILE DD DSN=SED.GLOB.DES.OUTFILE,  
//         DISP=(NEW,CATLG,DELETE),  
//         UNIT=DISK,  
//         SPACE=(CYL,(1,5),RLSE),  
//         DCB=(RECFM=FB,LERECL=70,BLKSIZE=700)
```

Programma da eseguire



Elementi di JCL

```
//JOB1 JOB (034D),'RAMESH',CLASS='A',PRTY=6
```



JOB statement

```
//STEP01 EXEC PGM=COBPROG.
```



EXEC statement

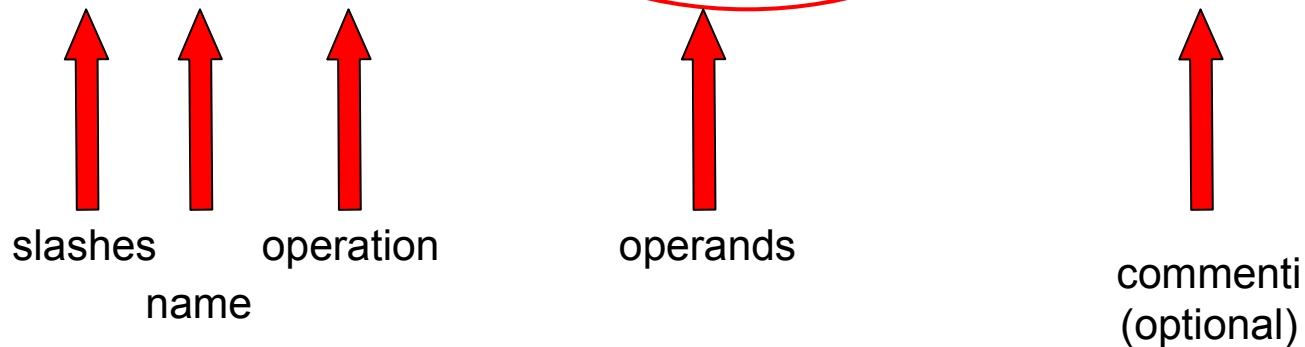
```
//INFILE DD DSN=SED.GLOB.DES.INFILE,DISP=SHR  
//OUTIFLE DD DSN=SED.GLOB.DES.OUTFILE,  
// DISP=(NEW,CATLG,DELETE),  
// UNIT=DISK,  
// SPACE=(CYL,(1,5),RLSE),  
// DCB=(RECFM=FB,LERECL=70,BLKSIZE=700)
```

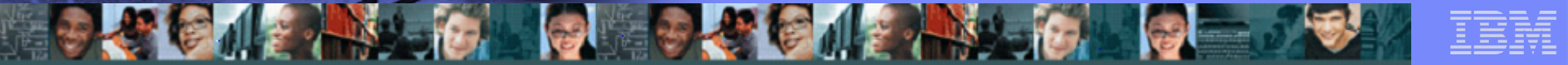


DD statement

Elementi di JCL(2)

```
// JOB1      JOB (034D),'RAMESH',CLASS='A',PRTY=6
// STEP01    EXEC PGM=COBPROG.
// INFILE    DD DSN=SED.GLOB.DES.INFILE,DISP=SHR
// OUTFILE   DD DSN=SED.GLOB.DES.OUTFILE,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=DISK,
//           SPACE=(CYL,(1,5),RLSE),
//           DCB=(RECFM=FB,LERECL=70,BLKSIZE=700)
```





Definizioni

// Le 2 “slashes” sono poste all’inizio di ogni statement e cominciano a colonna 1

NAME Campo opzionale, definisce il nome dello statement. Inizia a colonna 3

Operation Indica il tipo di operazione da compiere

Operand specifica gli operandi per ogni statement. Inizia obbligatoriamente a colonna 16

Comment opzionale, inizia con /* e termina con */

Job Statement

```
//JOB1 JOB (034D) , 'RAMESH', CLASS=A, MSGCLASS=1, MSGLEVEL=(1,1), PRty=6
```

Account informations

Keywords

- Class** usata per classificare il Job . Valori da A a Z
- Msgclass** definisce l'output dove verranno scritti i messaggi di sistema e delle JCL
- Msglevel** specifica quali messaggi e statements di JCL verranno scritti in output
- Prty** definisce la priorità, all'interno della Classe, con la quale verrà eseguito il job

Exec Statement

```
//STEP01 EXEC PGM=COBPROG, "keywords", REGION=
```

Programma da eseguire

- **Keywords**

Parm usata per passare, se necessario, parametri al programma

- **REGION**

Specifica la quantità di spazio di memoria che lo Step può usare. Espressa in M or K

Data Definition Statement

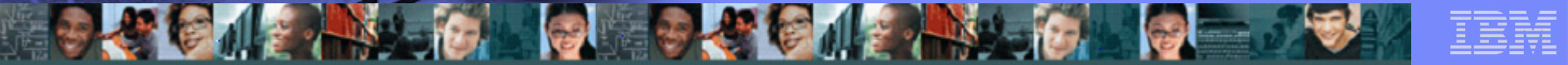
```
//INFILE DD DSN=SED.GLOB.DES.INFILE,DISP=SHR
```

File di Input

• Keywords

DSN	specifica il file
DISP	indica se il file esiste o se deve essere creato
UNIT	indica l'indirizzo del device sul quale risiede il file
SPACE	usato per allocare lo spazio, in cilindri tracce o blocchi, per il file
DCB	indica se i record nel file sono di lunghezza fissa o variabile
VOLUME	indica il nome del disco sul quale risiede il file

Identifica l'input e l'output necessari al programma



DD Statement particolari

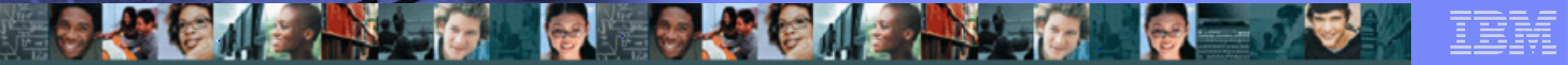
//INFILE DD DUMMY

permette di simulare la presenza del file

//SYSIN DD *

modalità per passare i dati al programma

```
//JOB1      JOB (034D),'RAMESH',CLASS='A',PRTY=6
//STEP01    EXEC PGM=COBPROG.
//INFILE    DD DSN=SED.GLOB.DES.INFILE,DISP=SHR
//OUTIFILE  DD DSN=SED.GLOB.DES.OUTFILE,DISP=SHR
//SYSIN     DD *
1234567
/*
```



Joblib/Steplib Statement

```
//Joblib DD DSN=SED.GLOB.LOADLIB,DISP=SHR
```

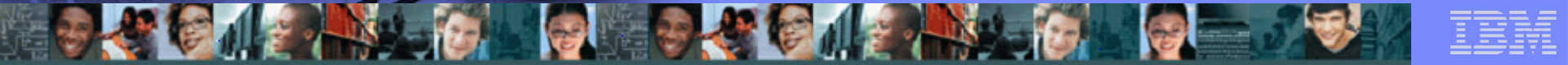
Libreria che contiene il programma

Specifica dove risiede il programma da eseguire. Si applica a tutti gli steps di quel Job.

```
//Steplib DD DSN=SED.GLOB.LOADLIB,DISP=SHR
```

Libreria che contiene il programma

Specifica dove risiede il programma da eseguire. Si applica solo a quello step.

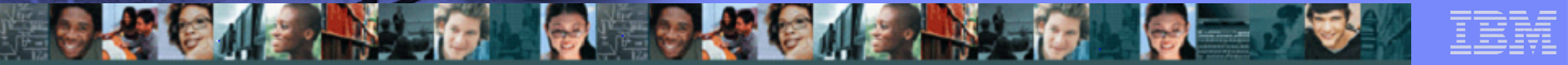


Program Search Order

Il sistema operativo “cerca” il programma da eseguire secondo questa sequenza:

- JOB Pack Area** **←===== programma caricato con il Job**
- Steplib**
- Joblib**
- Link Pack Area (LPA)** **←===== area di sistema, caricata a tempo di IPL**
- System Link List** **←===== area di sistema, caricata a tempo di IPL**

Se, alla fine della ricerca, il programma non viene trovato, il Job termina in “abend”



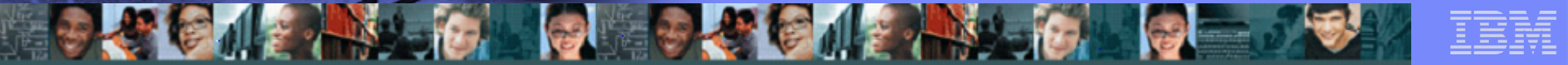
Procedure catalogate

Le JCL possono essere “riusate” e richiamate da altre JCL nella forma di “procedure”.

La modalità per richiamare una procedura, è la seguente:

```
//STEP01 EXEC PROC=name
```

La ricerca di una procedura, a differenza di un programma, viene effettuata in “librerie” particolari dette **PROCLIB**.



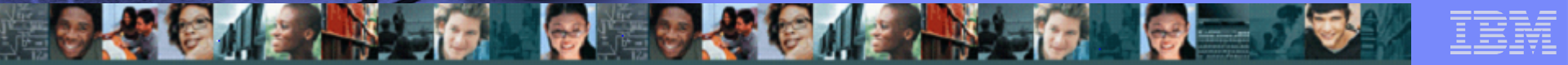
Instream

Le procedure possono essere anche "instream", cioè contenute all'interno delle mie JCL.

```
//MYJOB Job (E342),"PIPPO",MSGCLASS=A,MSGLEVEL=(1,1)
//MYPROC PROC
//STEP1 EXEC PGM=COBPROG
//INFILE DD DSN=TEST.GLOB.LIB,DISP=SHR
.....
// PENDING
//MYSTEP EXEC PGM=MAINPROG
//MYFILE DD DSN=TEST.GLOB.MYLIB,DISP=SHR
//MSTEP DD MYPROC
```

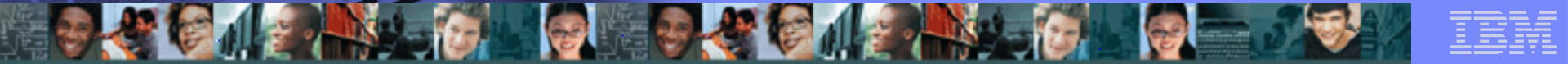
Instream Proc

Main JCL



Riferimento:

<http://www.geocities.com/srcsinc/drona/programming/languages/jcl/jcl.html>



Alcune relazioni

	System Z	Unix	Intel
Disco	Volume	Hdisk	Virtual disk
Volume Group	Volume Group	Volume Group	Volume Group
Unità di allocazione	Cylinder	Physical Partition	Physical Partition
Spazio riservato ai dati	Dataset	FileSystem	FileSystem