

Classic Generative Program



2024-10-18

三木悠吾

三木 悠吾 (Yugo Miki)



所属：グラクソ・スミスクライン株式会社
有限会社三栄
ミハルラボ（仮）

GSKで働いていることが多いです。他はたまに。
今回の発表はGSKとして発表しているわけではありません。

【職務経験】

- ・ 研究系（NC-PK, IoT, Big Data, Sensing）
- ・ 開発系（NC-PK、DM、PG、SYS）
- ・ 社内SEなど

【最近やったゲーム】

- ・ 風来のシレン6
- ・ 崩壊3rd / 原神

ことのおこり

三木がSASに初めて触れて1wほど、SAS社の研修を終えたころのこと。

S木「三木さん、これからこの会社のSASプログラムを作っていてもらうことになるんだけど、マクロパラメーターを大量に使うようなプログラムを作成するのは禁止だよ。」

三木「はあ。別にそれでいいですけど・・・。」

S木「・・・なぜかというと・・・。」

三木「・・・なるほど。だとすると、・・・。」

というような経緯の結果、Generativeなプログラムが開発された。

「Generative」とは？

Chat GPTに聞いてみた。

「Generative」 = 「生成する」「生み出す」

- ▶ 「Generative AI」とは、新しい情報やコンテンツを自動的に生成・創出する能力を持つAI技術であり、多様な分野で革新的な応用が可能です。

Chat GPT o1-miniの回答から一部省略

SASの「Generative」機能とは？

★ Call execute

データステップ内で呼び出せる。

Executeの引数をデータステップ後に実行する。

★ put statement

データステップで使えるfunction。

データの型変換だけでなく、外部ファイルへ出力することが可能。

Call execute 1

1. Call Executeの引数を現在のデータステップ終了直後に実行する。
2. 複数行でcallを書くと複数行生成される。
3. 各callの実行回数はコールされた回数 = setされたデータのobs数だけ実行される。

example1

```
data _null_;  
  call execute("data _null_; put 'Hello Generative  
SAS!'; run; ");  
run;
```

NOTE: CALL EXECUTEが生成したラインです。

```
1 + data _null_; put 'Hello Generative SAS!'; run;  
Hello Generative SAS!
```

Example2

```
data _null_;  
  call execute("data _null_;");  
  call execute(" put 'Hello Generative SAS!';");  
  call execute("run;");  
run;
```

NOTE: CALL EXECUTEが生成したラインです。

```
1 + data _null_;  
2 + put 'Hello Generative SAS!';  
3 + run;  
Hello Generative SAS!
```

Call execute 2

Call executeはデータセット内で呼び出すことで機能するため、エクセルなどの外部仕様を読み込んでprogramをキックできる。

合計行数: 2 合計列数: 3

	id	dsn	cond
1	1	class	age>=12
2	2	cars	horsepower>280

1. 上記のようなedit checkの仕様に対して、チェックコードを生成して実行する。

Example3

```
data _null_;
  set plan;
  call execute('data _null_');
  call execute('  set sashelp.' || cats(dsn) || ';'');
  call execute('  where ' || cats(cond) || ';'');
  call execute('  length msg $200. ;');
  call execute('  msg = cat("ERR", "OR:
#", ' || cats(id) || ', " Condition:",
cats("' || cats(dsn, "-" , cond) || '"));');
  call execute('  put msg ;');
  call execute('run;');
run;
```

Call execute 3

1. Executeがそれぞれのコードを作成し、実行してくれている。チェックもかかっているよう。

NOTE: CALL EXECUTEが生成したラインです。

```
1 + data _null_;  
2 + set sashelp.class;  
3 + where age>=12;  
4 + length msg $200.;  
5 + msg = cat("ERR","OR: #",1," Condition:",  
cats("class-age>=12"));  
6 + put msg ;  
7 + run;
```

ERROR: #1 Condition:class-age>=12

```
8 + data _null_;  
9 + set sashelp.cars;  
10 + where horsepower>280;  
11 + length msg $200.;  
12 + msg = cat("ERR","OR: #",2," Condition:",  
cats("cars-horsepower>280"));  
13 + put msg ;  
14 + run;
```

ERROR: #2 Condition:cars-horsepower>280

Call execute 4

マクロ変数とは相性が悪い。というか直感的に組み合わせにくいので、慣れるまではマクロ変数禁止！

実行される順番

1. マクロのdecode
2. Decodeされた状態でプログラムの実行
3. Executeで生成されたコードの実行

マクロ変数でいろいろ制御しようとする苦しみことになる。どうしても必要なら、`symput(x)`と`symget`で対応すべし。

example

```
%macro work(id);  
data _null_;  
  set file&id.;  
  where filename contains "program";  
  call execute(cats('%incl "',filename,'"');');  
run;  
%mend;  
%work(1)
```

Call execute 5

おすすめの使い方。

Sashelp.vmemberなどのメタデータとは相性がいい。Libnameから見ることのできるすべてのファイルに対して、特定の処理をかけたりするがよい。Dopenなどで取得してもいいが、このあたりの関数はあまり直感的でなく、仕様探して取り扱うよりはexecute + metadataで回した方が格段に安易。

example

```
%macro work(dsn);  
libname sdtmout xport "/libname/&dsn..xpt";  
proc copy data=sdtm.&dsn out=sdtmout;  
select &dsn.;  
run;  
%mend;  
  
data _null_;  
set sashelp.vmember;  
where libname = "SDTM" and memtype = "DATASET";  
call execute(cats('%work(', memname, ');'));  
run;
```

Put statement 1

Put statementの使い方

- ①データ型変換など
- ②ログや外部ファイルへの出力

今回は②の使い方。

filenameで外部ファイルを定義し、そこへputで出力する。

example

```
filename ext "/root/test.txt";
```

```
data _null_;  
  file ext;  
  put "Hello World!";  
run;
```



Put statement 2

ユーザー定義error, warningの使い方

Exampleのようにput で" ERROR :"と記載するとlog上にERROR, WARNINGが出力される。

★Tips

"ERR" "OR:"と書くと社内のlog checkで"ERROR"を抽出している場合、すり抜けることが出来る。

2行目は"- "と記載すると行の頭のERRORやWARNINGが消えて見やすくなる。

example

```
data _null_;
  set sashelp.class;
  if age >= 16 then do;
    put "ERR" "OR: Age >= 16 detected!";
    put "ERR" "OR- " Name= ", " Sex= ", " Age= ;
  end;
  if age < 12 then do;
    put "WAR" "NING: Age < 12 detected!";
    put "WAR" "NING- " Name= ", " Sex= ", " Age= ;
  end;
end;
```

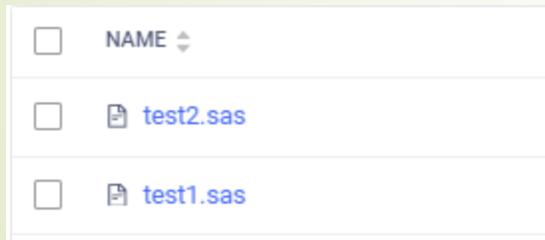
run;

```
85      data _null_;
86          set sashelp.class;
87          if age >= 16 then do;
88              put "ERR" "OR: Age >= 16 detected!";
89              put "ERR" "OR- " Name= ", " Sex= ", " Age= ;
90          end;
91          if age < 12 then do;
92              put "WAR" "NING: Age < 12 detected!";
93              put "WAR" "NING- " Name= ", " Sex= ", " Age= ;
94          end;
95      run;
WARNING: Age < 12 detected!
         Name=Joyce , Sex=F , Age=11
ERROR: Age >= 16 detected!
         Name=Philip , Sex=M , Age=16
WARNING: Age < 12 detected!
         Name=Thomas , Sex=M , Age=11
NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds
```

Combination

簡単！準自動化！！

1. Metadataから帳票情報読み込む。
2. 出力内容はマクロでひとまとめ
3. executeからkick→プログラムが出来ている。
4. プログラムは各プログラマーが手直しして実行→完！



example

```
data plan;
  length tnum t1 t2 $40.;
  tnum = "14.3.1"; t1 = "Overview of AEs"; t2 = "Safety";
output;
  tnum = "14.3.2"; t1 = "Summary of AEs by SOC and PT"; t2
= "Safety"; output;
run;
```

```
%macro create_pg(id, tnum, t1, t2);
filename ext "/root/test&id..sas";
```

```
data _null_;
  file ext;
  put "title1 &tnum. ";
  put "title2 &t1. ";
  put "title3 &t2. ";
```

```
run;
%mend;
```

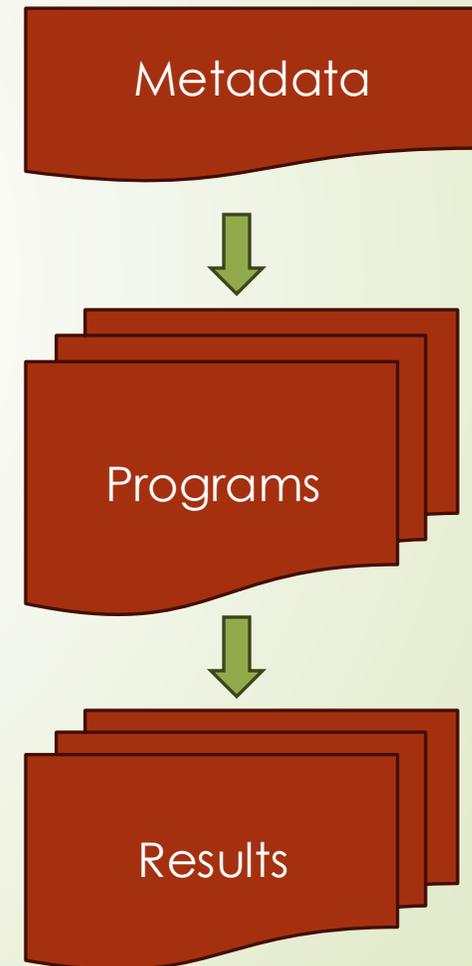
```
data _null_;
  set plan;
  n = _n_;
  call execute(cats('%create_pg(', n, ',', tnum, ',', t1,
',', t2, '));
run;
```

まとめ

Metadata、Program、結果（データや統計量など）というフローは Call execute、put statementで実装することが可能。

★ What is Your Metadata ?

解析仕様書、mockup、Define.xml、QC記録、sashelpの metadata . . .





Reference

- ▶ SAS Macro Language Magic: Discovering Advanced Techniques
- 



End

- ▶ The real voyage of discovery consists not in seeking new landscapes, but in having new eyes.
—— Marcel Proust