成都理工大学 | 地球物理学院

《地球与空间探测数据处理方法》

第二章 数据分析工具：IDL编程

陶 丹

Email：adam.tao@hotmail.com

Address: 北翼楼（地物院）5814室

课程要求和考核方式

课程基础

高等数学/线性代数/概率论与数理统计
高级程序语言与程序设计(C)/Matlab 程序设计

学习目的

1. 掌握IDL可视化分析工具的基础内容，能够独立程序编写并处理地球空间数据。
2. 能够熟悉了解空间探测数据类型及数据文件的存储格式，并能够熟练利用IDL可视化工具来实现空间探测数据的读写、数据格式的转换以及相关可视化分析等。
3. 能够熟练掌握空间探测数据的坐标变化、数据的平滑处理、滤波、谱分析、相关性分析、拟合、平均值统计、中值及误差分析等方法。

考核方式

平时成绩（40%）+ 考试成绩（60%）

其中：

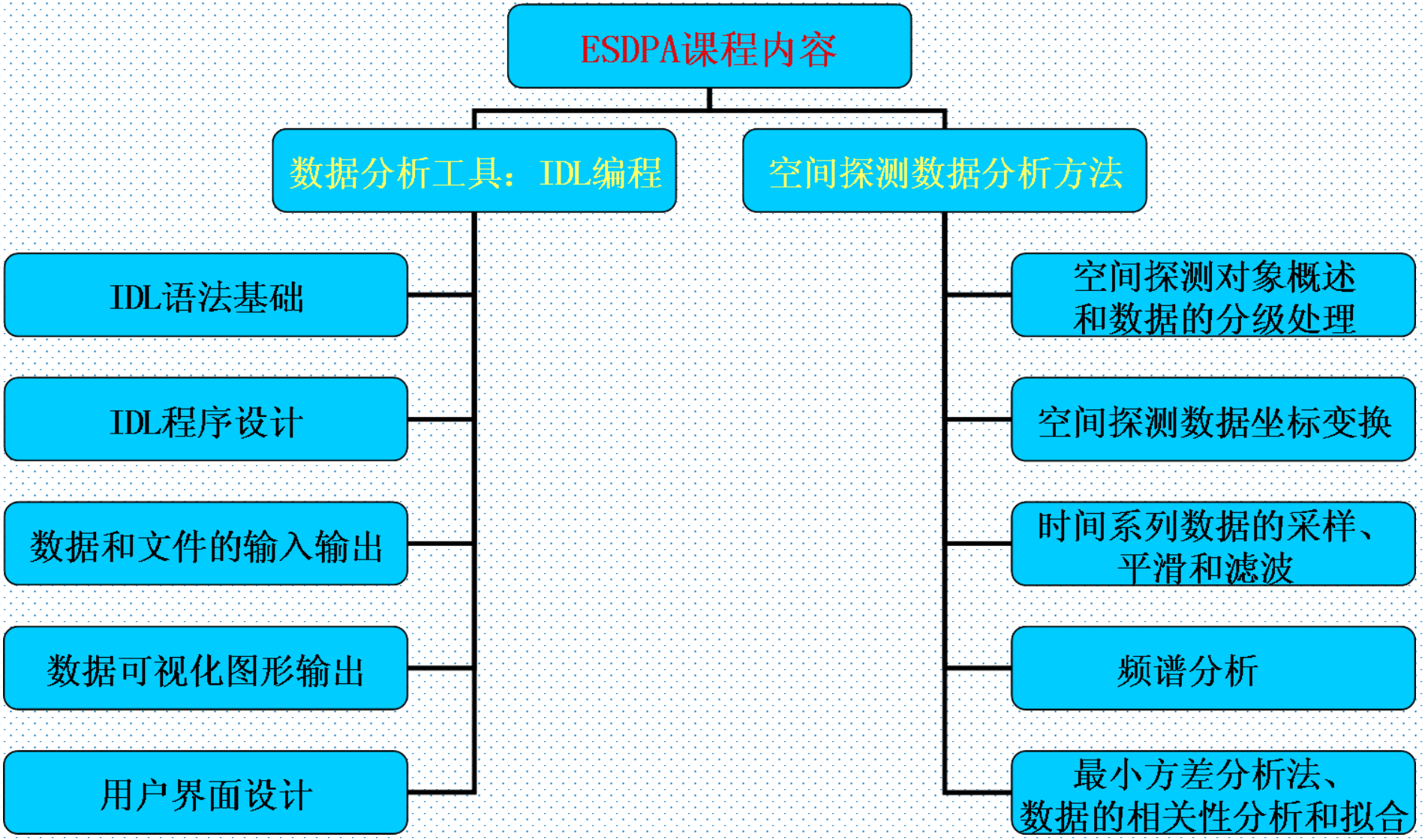
平时成绩（40%）：考勤（10%）+ 大作业/实验报告（30%）

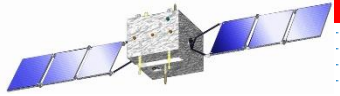
考试成绩（60%）：期末考试（60%，闭卷）

参考书目

- 《IDL可视化工具入门与提高》，闫殿武编著，机械工业出版社，2003.
- 《IDL可视化分析与应用》，韩培友编著，西北工业大学出版社，2006.
- 《IDL程序设计—数据可视化与ENVI二次开发》，董彦卿编著，高等教育出版社，2012.

地球与空间探测数据处理方法课程架构





二、 IDL语法基础：结构体

2.11 结构体

由一组不同类型(字节型、整数型、浮点型和字符串型）常量、变量、数组或结构体组合而成的组合数据类型。结构体中每一个变量、数组或结构体为结构体成员。

IDL中的两种结构体：匿名结构体和署名结构体。

- 匿名结构体：程序运行过程中，数据成员类型和个数(维度）可能发生变化。
  匿名结构一般用于程序之间值的传递。
  没有名字的结构。

- 署名结构体：程序运行过程中，命名结构体的变量遵循用户所创建的模板。
  一旦这种模板被创建，那么该结构体就不能随便改变。
  署名结构主要用在事件结构中。

二、IDL语法基础：匿名结构体

匿名结构的创建

- 创建格式1：结构变量1 = {成员1：表达式1，成员2：表达式2，…，成员N：表达式N}

```
IDL> STUDENT_LIST1={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> HELP,STUDENT_LIST1
     STUDENT_LIST1   STRUCT    = ->ANONYMOUS> ARRAY[1]
IDL> HELP,STUDENT_LIST1,/STRUCTURE ；关键字/STRUCTURE可以看结构体的具体内容。
     ** STRUCTURE <156FF2C0>, 3 TAGS, LENGTH=24, DATA LENGTH=22, REFS=1:
     NAME        STRING    'YANG'
     NUMBER      LONG      2017051001
     SCORE       INT       96
```

IDL低版本，高版本直接查看完全信息

匿名结构体定义后其成员个数可以增减：

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96,PHONE:'136010002546'}
```

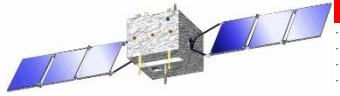
- 创建格式2：结构变量2 = CREATE_STRUCT(‘成员1’, 表达式1[, …, ‘成员N’, 表达式N][, 结构变量21 [, … , 结构变量2N]][, NAME=‘结构名’])

  或者：结构变量2 = CREATE_STRUCT([‘成员1’, …, ‘成员N’],表达式1, …,表达式N [, 结构变量21 [, … , 结构变量2N]][, NAME=‘结构名’])

  -利用结构创建函数创建包含N个成员和N个结构变量的匿名结构体。

  -NAME=‘结构名’，创建名为‘结构名’的署名结构体。

```
IDL> STUDENT_LIST2=CREATE_STRUCT('NAME','YANG', 'NUMBER', 2017051001L, 'SCORE', 96)
IDL> STUDENT_LIST3=CREATE_STRUCT(['NAME', 'NUMBER', 'SCORE'], 'YANG', 2017051001L, 96)
IDL> STUDENT_LIST41={PHONE:'13601002546'}   IDL> STUDENT_LIST42={RANK:4}
IDL> STUDENT_LIST4=CREATE_STRUCT('NAME','YANG', 'NUMBER', 2017051001L, 'SCORE',
     96,STUDENT_LIST41,STUDENT_LIST42)
```



二、IDL语法基础：匿名结构体

- 匿名结构的赋值
- 两种赋值格式：
  - 结构变量.成员 = 表达式

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> STUDENT_LIST.NAME='YANG'
```

  - 结构变量.（成员序号）

```
IDL> STUDENT_LIST.(0)='YANG'
```

结构体的任何变量均可进行相同类型的赋值调整，但结构体中变量的类型不能改变。

```
IDL>STUDENT_LIST.(2)=LONG(STUDENT_LIST.(2))
IDL> HELP,STUDENT_LIST.(2)
      <EXPRESSION>   INT       =        96
IDL> STUDENT_LIST.(2)='FENG'
      % TYPE CONVERSION ERROR: UNABLE TO CONVERT GIVEN STRING TO INTEGER.
      % DETECTED AT: $MAIN$
```

二、IDL语法基础：匿名结构体

- 匿名结构的调用
- 两种调用格式：
  - 结构变量.成员名

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> PRINT,STUDENT_LIST.NAME
       YANG
IDL> PRINT,STUDENT_LIST.SCORE
       96
```

  - 结构变量.(成员序号)

```
IDL> PRINT,STUDENT_LIST.(0)
       YANG
IDL> PRINT,STUDENT_LIST.(2)
       96
```

例如：

```
IDL>SS={TAG1:220,TAG2:239}
IDL>AVERAGE_SS=(SS.TAG1+SS.TAG2)/2.
IDL> PRINT,AVERAGE_SS
       229.500
```

二、IDL语法基础：匿名结构体

- 嵌套匿名结构的创建
- 嵌套结构体的创建：结构变量 = {成员1：表达式1，…，成员N：表达式N，结构体变量：结构体变量}

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> LIST={DATE:'20_MAR_2012',TIME:'10:52:02',STUDENT_LIST:STUDENT_LIST}
IDL> PRINT,LIST.(0)
        20-MAR_2012
IDL> PRINT,LIST.(1)
        10:52:02
IDL> PRINT,LIST.(2)
        { YANG  2017051001      96}
IDL> PRINT,LIST.STUDENT_LIST.(2)
        96
```

二、IDL语法基础：匿名结构体

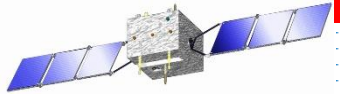
- 匿名结构体数组的创建：

  数组名 = **REPLICATE**(结构体变量，数组元素个数）

```
IDL> LIST={DATE:'20_MAR_2012',TIME:'10:52:02',STUDENT_LIST:STUDENT_LIST}
IDL> STRUCT_DATA=REPLICATE(LIST,10)
IDL> HELP,STRUCT_DATA
      STRUCT_DATA    STRUCT    = ->  <Anonymous> Array[10]
```

不可将看起来十分相似的结构体直接用[]合并到一起：

```
IDL> STRUCT_C=[STUDENT_LIST,LIST]
 % CONFLICTING DATA STRUCTURES: STUDENT_LIST,LIST.
 % EXECUTION HALTED AT: $MAIN$
```



二、IDL语法基础：署名结构体

- 署名结构的创建
  - 创建格式1：结构变量1 = {结构名称，成员1:表达式1，成员2:表达式2, …, 成员N:表达式N}

    IDL> DOT={PIXEL,X:128,Y:236,COLOR:BYTARR(3)}

    也可以用变量给结构体赋值：

    IDL> X0=0 & Y0=0 & GRAY=BYTE([127,127,127])

    IDL> ORIGIN={PIXEL,X:X0,Y:Y0,COLOR:GRAY}

    署名结构体定义后其成员个数不能改变：

    IDL> DOT={PIXEL,X:128,Y:236,COLOR:BYTARR(3),RECORD:6}

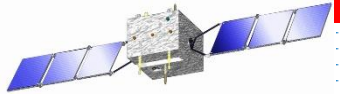
    % WRONG NUMBER OF TAGS DEFINED FOR STRUCTURE: PIXEL.

    % EXECUTION HALTED AT: $MAIN$

  - 创建格式2：结构变量2 = CREATE_STRUCT(‘*成员1*’, *表达式1*[, …, ‘*成员N*’, *表达式N*][, 结构变量21 [, …, 结构变量2N]], NAME=‘结构名’)

    或者：结构变量2 = CREATE_STRUCT([‘*成员1*’, …, ‘*成员N*’], *表达式1*, …, *表达式N* [, 结构变量21 [, …, 结构变量2N]], NAME=‘结构名’])

    -利用创建结构函数创建名为‘结构名’，且包含N个成员和N个结构变量的署名结构体。



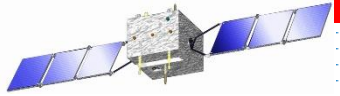
二、IDL语法基础：署名结构体

- 署名结构的创建
  - 创建格式3：结构变量3 = {结构名称}
    使用已有署名结构体的结构名，创建成员初始值为0和空字符串的结构变量。

```
IDL> STUDENT_LIST={STUDENT_LIST,NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> STUDENT_LIST1={STUDENT_LIST}
IDL> HELP,STUDENT_LIST1
     ** STRUCTURE STUDENT_LIST, 3 TAGS, LENGTH=24, DATA LENGTH=22:
     NAME            STRING      ''
     NUMBER          LONG        0
     SCORE           INT         0
```

- 署名结构的调用
  - 两种调用格式：结构变量.成员名 或 结构变量.(成员序号)

```
IDL> X0=0 & Y0=0 & GRAY=BYTE([127,127,127])
IDL> ORIGIN={PIXEL,X:X0,Y:Y0,COLOR:GRAY}
IDL> PRINT,ORIGIN.COLOR    OR   IDL>PRINT, ORIGIN.(2)
       127 127 127
```



二、IDL语法基础：署名结构体

署名结构的赋值

- 两种赋值格式：结构变量.成员 = 表达式 或 结构变量.(成员序号)

```
IDL> ORIGIN.COLOR=[254,254,254]
IDL> ORIGIN.(2)=[254,254,254]
IDL> PRINT,ORIGIN.(2)
      254 254 254
```

署名结构体的任何变量均可进行相同类型的赋值调整，但结构体中变量的类型不能改变。

- 署名结构体的复制：结构变量 = {结构体名称}

```
IDL> ORIGIN1={PIXEL}
IDL> PRINT,ORIGIN1.X
      0
```

- 署名结构体数组的建立：数组名 = REPLICATE({结构体名称}，数组元素个数）

```
IDL> STRUCT_DATA1=REPLICATE({PIXEL},10)
IDL> HELP, STRUCT_DATA1
      STRUCT_DATA1       STRUCT    = -> PIXEL ARRAY[10]
IDL> STRUCT_DATA2=REPLICATE(ORIGIN1,10)
IDL> HELP, STRUCT_DATA2
      STRUCT_DATA2       STRUCT    = -> PIXEL ARRAY[10]
```

二、IDL语法基础：匿名结构体

- 检测结构成员的个数和名称：
  - 变量=N_TAGS(结构变量 [, /DATA_LENGTH] [, /LENGTH])

    -/DATA_LENGTH，返回所有结构成员的值的长度和。

    -/LENGTH，返回结构的长度。
  - 变量=TAG_NAMES(结构变量 [, /STRUCTURE_NAME])

    -/STRUCTURE_NAME，返回结构名字。无成员名称的，默认为：FEILD1,…,FIELDN

```
IDL> SS1={TAG1:'SPACE SCIENCE'}
IDL> SS2={TAG1:'AND',TAG2:'TECHNOLOGY'}
IDL> SS3={TAG1:2019}
IDL> SS=CREATE_STRUCT(['UNIVERSITY','MAJOR1','MAJOR2'],'CDUT',SS1,SS2,SS3)
IDL> PRINT,SS
      { CDUT{ SPACE SCIENCE}{ AND TECHNOLOGY}   2019}
IDL> HELP,SS
      ** STRUCTURE <F01D690>, 4 TAGS, LENGTH=72, DATA LENGTH=66, REFS=1:
      UNIVERSITY    STRING   'CDUT'
      MAJOR1        STRUCT   -> <Anonymous> Array[1]
      MAJOR2        STRUCT   -> <Anonymous> Array[1]
      TAG1          INT      2019
IDL> STUDENT_LIST={STUDENT_LIST,NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> PRINT,TAG_NAMES(STUDENT_LIST,/STRUCTURE_NAME)
      STUDENT_LIST
```

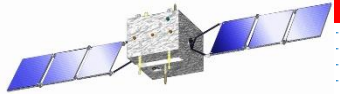
二、IDL语法基础：结构体操作函数

- 结构体操作函数列表

函数名	功能
N_tags()	返回结构中变量的个数
Tag_names()	返回结构中每个变量的名字
Create_struct()	创建一个结构，或者将变量附加到已有结构中去

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> LIST={DATE:'20_MAR_2012',TIME:'10:52:02',STUDENT_LIST:STUDENT_LIST}
IDL> PRINT,N_TAGS(LIST)
       3
IDL> PRINT,TAG_NAMES(LIST)
      DATE TIME STUDENT_LIST

IDL> BOOK=CREATE_STRUCT('TITLE','COSMOS','AUTHOR','CARLSAGAN')
IDL> BOOK=CREATE_STRUCT(BOOK,'PUBDAT',1980)
IDL> HELP,BOOK,/STRUCTURE
      ** STRUCTURE <3E6890>, 3 TAGS, LENGTH=28, DATA LENGTH=26, REFS=1:
      TITLE           STRING   'COSMOS'
      AUTHOR          STRING   'CARLSAGAN'
      PUBDAT          INT       1980
```

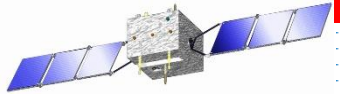


二、 IDL语法基础：指针

- 基础概念
  - 内存由内存单元构成。
  - 内存变量的值存放于内存中指定的内存单元中。
  - 内存单元是由内存单元的地址和内容组成的存储单元。
    内存单元的地址为存放内存变量值的那个指定内存单元的地址，即内存变量的地址。
    内存单元的内容为存放内存变量值的那个指定内存单元的值。

- 指针是用于存放内存变量地址的特殊变量，即指针地址。

- 指针是对IDL变量（标量、数组、结构体或其他指针）的一个指向。
  它并不具有通常意义的“值”。传递指针变量就是传递它指向的变量。



二、 IDL语法基础：指针

- 指针是对IDL变量（标量、数组、结构体或其他指针）的一个指向。它并不具有通常意义的“值”。传递指针变量就是传递它指向的变量。

- 指针的创建格式：

  指针变量 = **PTR_NEW**([VAR])。

  -创建用于存储变量VAR地址的指针变量，即指针变量指向变量的地址。

  -无参数的PTR_NEW( )将新建一个不指向任何变量的空指针。

  IDL> X=5.0 ***IDL> Y=INDGEN(5)***

  IDL> Z={FLAG:1L,P1:PTR_NEW(INDGEN(5))}

  IDL> P=PTR_NEW(X) ***IDL> P1=PTR_NEW(Y)*** ***IDL> P2=PTR_NEW(Z)***

- 创建未定义变量的指针：**PTR_NEW(/ALLOCATE_HEAP)**

  -未设置变量时，返回一个未定义的指针变量，而非空指针。

二、IDL语法基础：指针

- 指针的调用（提领）格式：*指针变量

-提取指针变量所指向地址中的数据。

-IDL变量跟踪窗口中可跟踪指针变量的类型及所指向的数据。

```
IDL> PRINT,*P                IDL> PRINT,*P1          IDL>PRINT,*P2
5.00000                      0     1     2     3     4     { 1 <PTRHEAPVAR24>}
IDL> (*P-3.)*9.+*P1
```

-当提取数组元素时，应将指针变量用小括号括起来，再加下标提取：

```
IDL> PTR0=PTR_NEW(INDGEN(4,2))
IDL> PRINT,*PTR0
       0     1     2     3
       4     5     6     7
IDL> PRINT,*PTR0[5]
 % ATTEMPT TO SUBSCRIPT PTR0 WITH <INT      (      5)> IS OUT OF RANGE.
 % EXECUTION HALTED AT: $MAIN$
IDL> PRINT,(*PTR0)[5]
       5
```

二、IDL语法基础：指针

- 指针的调用（提领）格式：*指针变量

-当提领指向指针的指针时，提领运算符 * 可以重复使用：

```
IDL> PTR1=PTR_NEW(12.0)
IDL> PTR2=PTR_NEW(PTR1)
IDL> PRINT,*PTR2
      <PTRHEAPVAR2>
IDL> PRINT,**PTR2
      12.0000
```

-当提领结构体成员时：

```
IDL> STUDENT_LIST={NAME:'YANG',NUMBER:2017051001L,SCORE:96}
IDL> PTR3=PTR_NEW(STUDENT_LIST)
IDL> PRINT,*PTR3
      { YANG  2017051001     96}
```

-当指针包含在结构体中时：

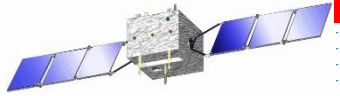
```
IDL>STUDENT_LIST1={NAME:'YANG',NUMBER:2017051001L,SCORE:96,PHONE_PTR:PTR_NEW(13620190928)}
IDL> PRINT,*STUDENT_LIST1.PHONE_PTR
      13620190928
```

二、IDL语法基础：指针

- 指针的调用（提领）格式：*指针变量

-当指针用来指向包含指针的结构体，在提领时必须先提领指向该结构体的指针（用小括号包含），然后再提领被结构体包含的指针：

```
IDL> STUDENT_LIST1={NAME:'YANG',NUMBER:2017051001L,SCORE:96,PHONE_PTR:PTR_NEW(13620190928)}
IDL> PTR4=PTR_NEW(STUDENT_LIST1)
IDL> PRINT,*(*PTR4).PHONE_PTR
      13620190928
```



二、IDL语法基础：指针

- 指针数组的创建：

  指针数组名 = **PTRARR**(D1, …, D8 [, /ALLOCATE_HEAP | , /NOZERO])。

  -若省略所有参数，则创建一个初值均为空指针的指针数组。

  -指针数组创建后，数组中每个元素相当于一个指针变量。

  IDL> VAR0=98.9

  IDL> PTRARRAY=PTRARR(3)

  IDL> PTRARRAY[2]=PTR_NEW(VAR0) ;指针数组的第3个元素指向变量VAR0

  IDL> PTR = PTRARR(10,/ALLOCATE_HEAP)

  IDL> PTR[0]=PTR_NEW(INDGEN(10))

- 检测指针的有效性：

  RESULT=PTR_VALID(指针变量名)

  IDL> P=PTR_NEW(10.0)

  IDL> PRINT,PTR_VALID(P)

  1；1表示有效，0表示无效

- 释放指针内存：

  PTR_FREE,指针变量名

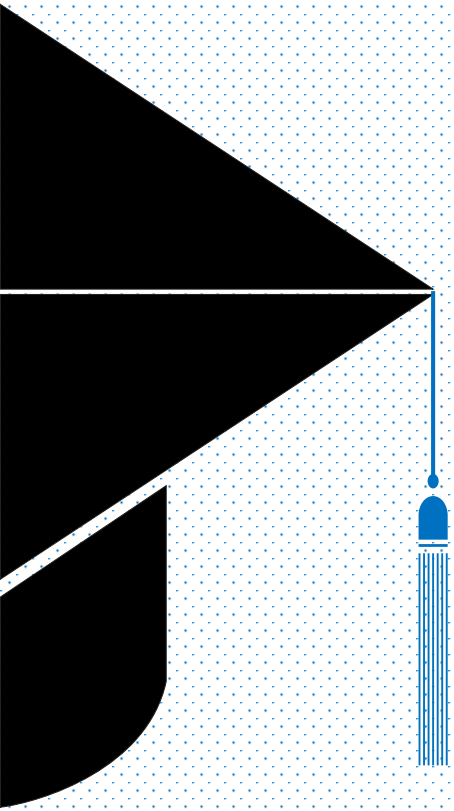
  IDL> PTR_FREE,P ；使P变成无效指针。指针被释放后可重新定义。

  IDL> PRINT,PTR_VALID(P)

  0

- 指针变量可极大提高程序运行速度；使用之后及时释放，避免占内存或导致内存溢出。
- 使用命令 HEAP_GC, /PTR, /VERBOSE 或 .RESET_SESSION释放所有指针。

成都理工大學
CHENGDU UNIVERSITY OF TECHNOLOGY



演示完毕

Thank You