ABAP 不像Java一样有丰富的数据类型可以使用，下通过内表实现二分查找的原理

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*& Report ZRPTEST002  \*& 描述:  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*& 创建人:  \*& 创建时间:  \*& 目的：用于测试二分查找 BINARY SEARCH  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* REPORT NAME :报表名称及格式  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  REPORT ZRPTEST002.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* DATA :定义报表中使用到的变量及内表  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  "所在位置  DATA: POSITION TYPE INT4 VALUE 0.  "数据内表  DATA:BEGIN OF IT\_DATA OCCURS 0,  NUMBER TYPE INT4,  END OF IT\_DATA.  DATA:WA\_DATA LIKE IT\_DATA.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*SELECTION-SCREEN :定义选择屏幕中的选择字段  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SELECTION-SCREEN BEGIN OF BLOCK TXT01 WITH FRAME TITLE TEXT-001.  PARAMETERS:P\_TARGET TYPE INT4 OBLIGATORY.  SELECTION-SCREEN END OF BLOCK TXT01.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* INITIALIZATION:程序初始化时候执行的代码  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  INITIALIZATION.  PERFORM FRM\_INITAL.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*START-OF-SELECTION:程序运行所处理的代码  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  START-OF-SELECTION.  PERFORM FRM\_BINARY\_SEARCH\_MD.  PERFORM FRM\_BINARY\_SEARCH\_SY.  \*&---------------------------------------------------------------------\*  \*&Form FRM\_INITAL  \*&---------------------------------------------------------------------\*  FORM FRM\_INITAL.  "初始化100个数据  DATA: NUMBER TYPE INT4 VALUE 0.  DO 100 TIMES.  NUMBER = NUMBER + 1.  CLEAR IT\_DATA.  IT\_DATA-NUMBER = NUMBER.  APPEND IT\_DATA.  ENDDO.  "二分查找前提：排序  SORT IT\_DATA BY NUMBER ASCENDING.  ENDFORM. "FRM\_INITAL  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 二分查找  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  FORM FRM\_BINARY\_SEARCH\_MD.  "定义二分查找条件变量：中间值、最低值和最高值  DATA:MID TYPE INT4 VALUE 0,  LOW TYPE INT4 VALUE 0,  HIGH TYPE INT4 VALUE 0.  "初始最低值 0  LOW = 0.  "最高值内表行数  HIGH = LINES( IT\_DATA ).  "循环次数  DATA: COUNT TYPE INT4 VALUE 0.  "循环切半查找  WHILE LOW <= HIGH.  "统计循环次数  COUNT = COUNT + 1.  "取中间位置  MID = ( LOW + HIGH ) / 2.  "切半  READ TABLE IT\_DATA INTO WA\_DATA INDEX MID.  IF SY-SUBRC = 0.  "刚好等于中间位置  IF P\_TARGET = WA\_DATA-NUMBER.  POSITION = MID.  EXIT.  ENDIF.  "当目标值大于中间位时，最低位等于中间位  IF P\_TARGET > WA\_DATA-NUMBER.  LOW = MID + 1.  ENDIF.  "当目标值小于中间位时，最高位等于中间位  IF P\_TARGET < WA\_DATA-NUMBER.  HIGH = MID - 1.  ENDIF.  ENDIF.  ENDWHILE.  "目标值位置  WRITE '自行实现：目标值位置：'.  WRITE POSITION.  WRITE '循环次数：'.  WRITE COUNT.  ENDFORM.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 二分查找  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  FORM FRM\_BINARY\_SEARCH\_SY.  READ TABLE IT\_DATA WITH KEY NUMBER = P\_TARGET BINARY SEARCH.  IF SY-SUBRC = 0.  POSITION = SY-TABIX.  ENDIF.  WRITE: /.  "目标值位置  WRITE 'ABAP标准:目标值位置：'.  WRITE POSITION.  ENDFORM. |