

# Delphi 及 Interbase 浮点数运算的精度问题

杨庆民 (河北省工商行政管理学校计算中心 050021)

**摘要:**在 Delphi 中直接比较浮点数是否相等时出现问题。在 Interbase 中浮点数的运算存在较严重的精度问题。本文给出相应的解决办法。

**关键词:**Delphi interbase 数据库 浮点数等

在最近笔者开发财务软件时,选用 Delphi4.0 作前端工具,interbase 数据库作后端工具。在软件测试时发现数据类型问题。问题一是:判断凭证的借贷双方是否平衡时,有时软件不能正确的比较本来平衡的判为不平衡。问题二是:解决问题一后,计算资产负债表时资产类之总和与负债类之总和不平衡。

问题一的解决:程序中的变量类型:借贷双方之变量分别定义为 s1、s2,数据类型为 double。interbase 表内有关数据类型为 numeric(15,2)。笔者首先检查程序流程,流程如下:首先 s1、s2 赋值为零、然后将凭证各分录的借贷双方分别求和、记入 s1、s2,最后比较 s1、s2 是否相等。程序流程比较简单,没发现问题。况且在借贷双方各分录皆为整数时软件能够正确判断;有小数时偶尔不能正确判断。于是开始单步跟踪,发现程序能够正确求和,显示变量 s1、s2 也相等,但在用 if s1 = s2 判断时出错。

笔者开始怀疑 delphi 的数据类型是否有错误。于是做例子工程,专门检查 delphi 的浮点数。为说明问题,读者可在 delphi 缺省的 form1 中放一 button1,unit1.pas 如下:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  f1: double;
begin
  f1 := 0.01;
  if f1 = 0.01 then
    form1.caption := '比较相等'
  else
    form1.caption := '比较不相等';
end;
```

运行程序,按 button1,form1.caption 变为'比较不相等',这样的结果让人感到吃惊。可见 delphi 的浮点数运算确有问题(为了慎重起见,笔者用 vc5.0 时未发现上述

问题,可以确认为 delphi 自身的问题)。

通过大量的检查,发现浮点数运算时产生的误差可以接受。但在比较是否相等时由于误差而发生误判。解决办法即保留指定位数的小数,这样就把误差滤掉了。以下是一函数,对浮点数保留指定位数的小数。

```
function GetFloatValue(f1: Double; n: integer): double
var
  i: integer;
  f2: double;
begin
  for i := 1 to n do
    f2 := f2 * 10; result := (round(f1 * f2)) / f2;
  end;
```

在比较浮点数是否相等时,将两浮点数分别用 GetFloatValue() 进行处理,即可得到正确的比较结果。这样问题一就得到了解决(该函数在有效小数为 2 位时,整数位可达千亿。超过此范围函数无效,若小数点多保留几位,整数位就相应少几位)。

问题二的解决:

经过仔细检查,笔者开始怀疑 interbase 的浮点数是是否存在问题。于是做了一 interbase 库,专门检查其浮点数的精度。该库如下:

```
create database "c:\float.gdb" page-size 1024
CREATE TABLE CROSS-RATE
```

```
(
  FROM-CURRENCY VARCHAR(10) NOT NULL,
  TO-CURRENCY VARCHAR(10) NOT NULL,
  CONV-RATE2 FLOAT NOT NULL,
  CONV-RATE1 FLOAT NOT NULL,
  UPDATE-DATE DATE,
  NU1 NUMERIC(15,2) NOT NULL,
```

NU2 NUMERIC(15,2) NOT NULL,  
PRIMARY KEY (FROM-CURRENCY, TO-CUR-  
RENCY)  
);  
INSERT INTO CROSS-RATE VALUES('DOLLAR',  
CDNDLR', 1. 3273, 1. 3273, '11/22/93', 1. 3273, 1.  
3273);  
INSERT INTO CROSS-RATE VALUES('DOLLAR',  
FFRANC', 5. 9193, 5. 9193, '11/22/93', 5. 9193, 5.  
9193);

在 CROSS-RATE 表中,定义了 FLOAT, NUMERIC  
(15,2)型。且插入两条记录。CONV-RATE1, CONV-  
RATE2, NU1, NU2 开始时数值相等。在笔者的例子程  
序中,将各字段与自身相乘,乘积再送回表中相应字段,  
一按钮控制,循环执行。开始时 FLOAT, NUMERIC  
(15,2)型数值相同,当数据超过百万时,FLOAT 型数据

即出错。而 NUMERIC(15,2)型数据可至千亿。

问题确定下来了,即 INTERBASE 中的 FLOAT 型  
数据的精度在超长运算时,如超出有效数位范围也会出  
现问题。

笔者的数据库中,用到了 FLOAT 型,所以引起了问  
题二。将库中的 FLOAT 型用 NUMERIC(15,2)型替代,  
问题二也得到了解决。

以上是笔者用 DELPHI、INTERABSE 时的经验教  
训,DELPHI 本身是公认的优秀前端工具。INTERBASE  
是 INTERBASE 公司的数据库产品,有 LOCAL-SERVER  
和 REMOTE-SERVER 两个版本都提供了触发器、发生  
器、存储过程等功能。在 WIN9X 下安装的 LOCAL-  
SERVER 版本开发的程序几乎不用改变即可适用于 RE-  
MOTE 版本。其 REMOTE 版本适用于 NT-SERVER,对  
于开发者是很大的方便。

(来稿时间:1999 年 1 月)