

Delphi and rounding

การคำนวณและปัดเศษ บน Delphi

เมื่อนานมาแล้วผ่านตากับคำถามหนึ่งใน internet ไม่แน่ใจว่าจากที่ไหน

ปัญหานี้

Test Currency

Rounding mode: Default system rounding mode: rmNearest.
rmNearest Digit display format: 2

Exclude VAT Amount : 11,291.50 **A**

VAT Amount : 7% 790.40 **B = A x 0.07**

Grand Total : 12,081.91 **C = A + B**

Withholding TAX : 1% 112.92 **D = A x 0.01**

NET Payment : 11,968.99 **E = C - D**

ใส่ A = 11,291.50
กคปุม 7% -> $11,291.50 \times 0.07 = 790.41$ <- ค่าที่ถูกต้อง
* แต่โปรแกรมโชว์ 790.40 ตามโปรแกรมนี้ B = 790.40
Grand Total C = A + B (11,291.50 + 790.40)
ต้องได้ค่า = 12,081.90
* แต่โปรแกรมโชว์ 12,081.91
กคปุม 1% -> $11,291.50 \times 0.01 = 112.92$ <- ค่าที่ถูกต้อง ครึ่งนี้ปัดถูก
NET Pay E = C - D (12,081.91 - 112.92)
= 11,968.99 <- ค่าถูกต้อง
ปัญหาคือ ทำไมตอน กค 7% คือ $11,291.50 \times 0.07 = 790.405$ ทำไมไม่มีปัดขึ้นเป็น 790.41
แต่ ตอนกค 1% คือ $11,291.50 \times 0.01 = 112.915$ ถึงปัดขึ้นเป็น 112.92

กำหนด ตัวแปร A, B, C, D, E เป็น currency

ใส่ A = 11291.50

กคปุม 7% -> $11291.50 \times 0.07 = 790.41$ <- ค่าที่ถูกต้อง

* แต่โปรแกรมโชว์ 790.40 ตามโปรแกรมนี้ B = 790.40

Grand Total C = A + B (11291.50 + 790.40)

ต้องได้ค่า = 12,081.90

* แต่โปรแกรมโชว์ 12,081.91

กคปุม 1% -> $11291.50 \times 0.01 = 112.92$ <- ค่าที่ถูกต้อง ครึ่งนี้ปัดถูก

NET Pay $E = C - D (12081.91 - 112.92)$

$= 11968.99$ <- ค่าถูกต้อง

ปัญหาคือ ทำไมตอน กด 7% คือ $11291.50 \times 0.07 = 790.405$ ทำไมไม่ปัดขึ้นเป็น

790.41

แต่ ตอนกด 1% คือ $11291.50 \times 0.01 = 112.915$ ถึงปัดขึ้นเป็น 112.92

คำตอบสำหรับปัญหา 2 ข้อมีดังนี้

1. ปัญหาคือ ทำไมตอน กด 7% คือ $11291.50 \times 0.07 = 790.405$ ทำไมไม่ปัดขึ้นเป็น

790.41

สำหรับ Delphi หรือ โปรแกรมตัวภาษาอื่น ๆ จะมี Floating-point data types ซึ่ง
จะมี FPU/SSE. เพื่อที่จะ จัดการเรื่อง floating-point หรือตัวเลขมีจุดทศนิยม

*The FPU (Floating Point Unit) handles floating-point calculations on 32-bit systems and the SSE (Streaming SIMD
Extensions) handles floating-point calculations on 64-bit systems.*

โดยไอ้เจ้าตัวกลางในการ handles นี้มันสามารถ กำหนดให้เลือกการปัดจุด ได้ 4 แบบดังนี้

Value	Meaning
rmNearest	Rounds to the closest value. (Default)
rmDown	Rounds toward negative infinity.
rmUp	Rounds toward positive infinity.
rmTruncate	Truncates the value, rounding positive numbers down and negative numbers up.

จากข้อมูลทีกล่าวมา ทำให้โปรแกรม Delphi เลือกใช้การปัดตัวเลขแบบทศนิยมแบบ rmNearest เพราะ เป็นค่า Default หรือค่า
เริ่มต้นของ Delphi

... ทำให้ เกิดคำถามข้อที่ 1 นั้นแหละ มันเลือกใช้การปัดแบบ rmNearest ทำให้ปัดขึ้นบ้าง ปัดลงบ้าง งงอะดิ ... แล้วมันปัดยังไง

ละ ไอ้เจ้า rmNearest เนี่ย อันนี้ต้องลองทดสอบเอง ซึ่งสามารถ download source ได้จาก [github](https://github.com/samrids/DelphiRoundingMode.git) ซึ่งผม push source ไว้แล้ว use git client เพื่อ clone source :

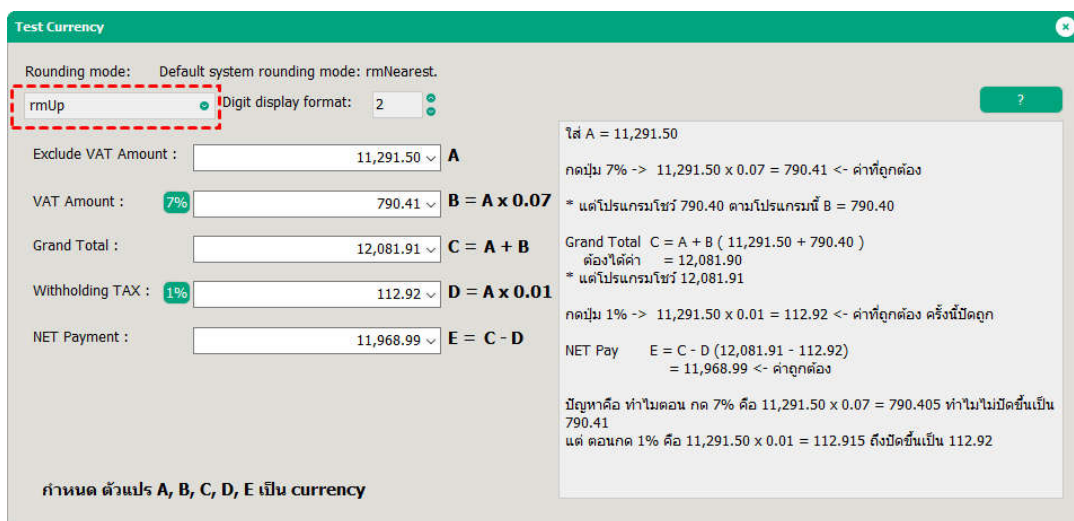
```
git clone https://github.com/samrids/DelphiRoundingMode.git
```

2. ตอนกด 1% คือ $11291.50 \times 0.01 = 112.915$ จึงปัดขึ้นเป็น 112.92

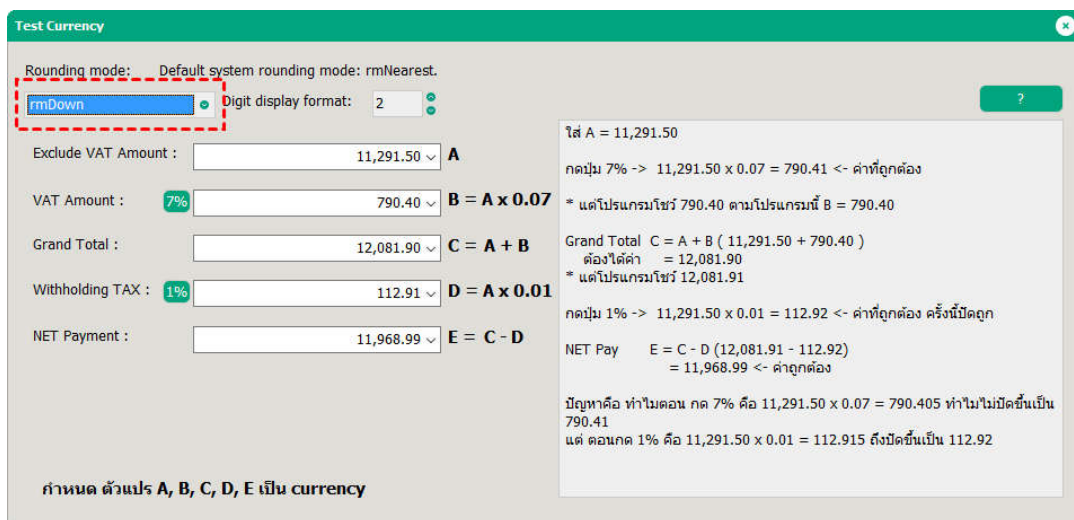
ตอบเหมือนข้อที่ 1 เนื่องจากเคลฟเลือกใช้ rmNearest ในการปัดเศษ

ทำไมมันเลือก rmNearest ก็เพราะมันเป็นค่าเรีสมแรก เริ่มต้นของ Delphi default rounding mode

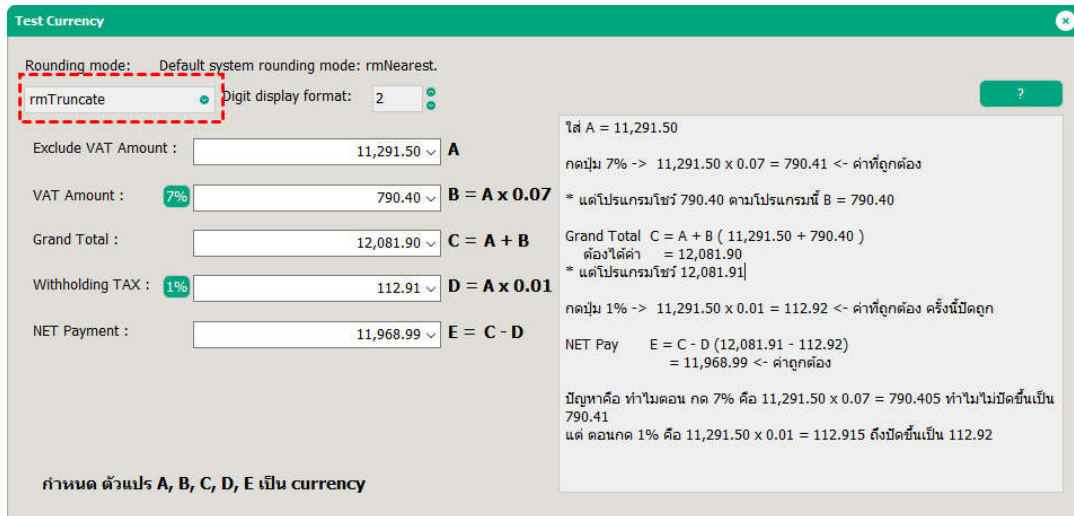
ลองเลือกใช้ การปัดแบบ rmUp ได้ผลตรงกับที่ต้องการเป๊ะ ๆ



ลองเลือกใช้ rounding mode : แบบ rmDown



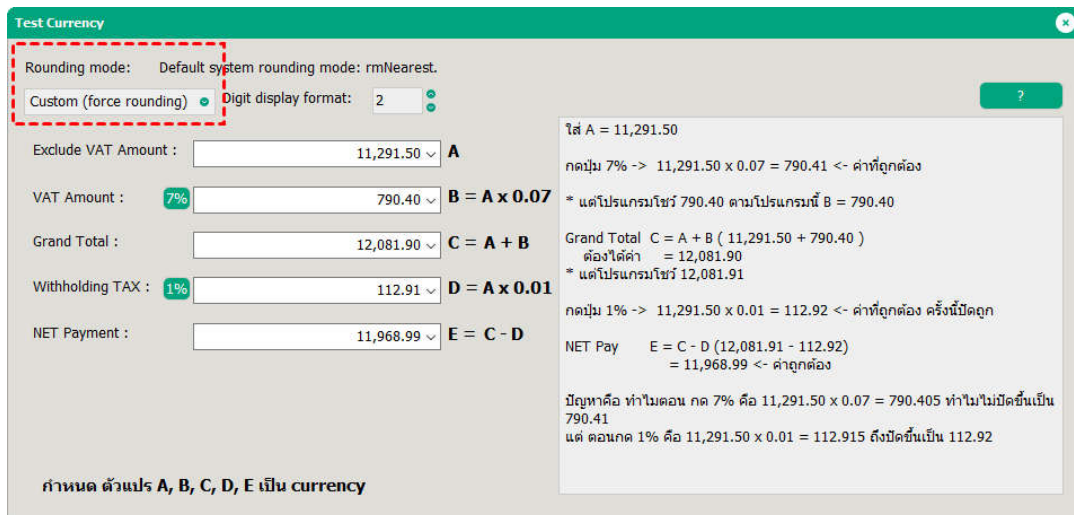
ลองเลือกใช้ rounding mode : แบบ rmTruncate



แบบสุดท้าย

ใช้ rounding mode : แบบ Custom (force rounding)

ตัวนี้ผม ค้นหาตัวอย่าง function และได้ทำการทดสอบแล้ว ไม่ว่า Delphi จะเลือก rounding mode เป็นตัวไหน ใน 4 ตัวที่กล่าวมาทั้งหมด โปรแกรมที่เราเขียนก็จะปัดเศษตามที่เรากำหนดไว้เท่านั้น เรียกได้ว่าเราบังคับ ให้มัน ปัดเศษ หรือ rounding ตามแบบที่เรากำหนดไว้คือ ปัดเศษโดยดูจากระกอบของ จุดทศนิยมหลายๆ ตัวประกอบกัน



Author: Samrid Somboon

Line: mesa.d

Email: samrid@gmail.com