Monte Carlo method

# โค้ดเทียม

Begin;  
int NUMBER\_OF\_TRIALS = 2147483647;

while i = 1 to NUMBER\_OF\_TRIALS

if ( i < NUMBER\_OF\_TRIALS ) then

do x << random;

y << random;

if ( x\*x + y\*y <= 1 ) then

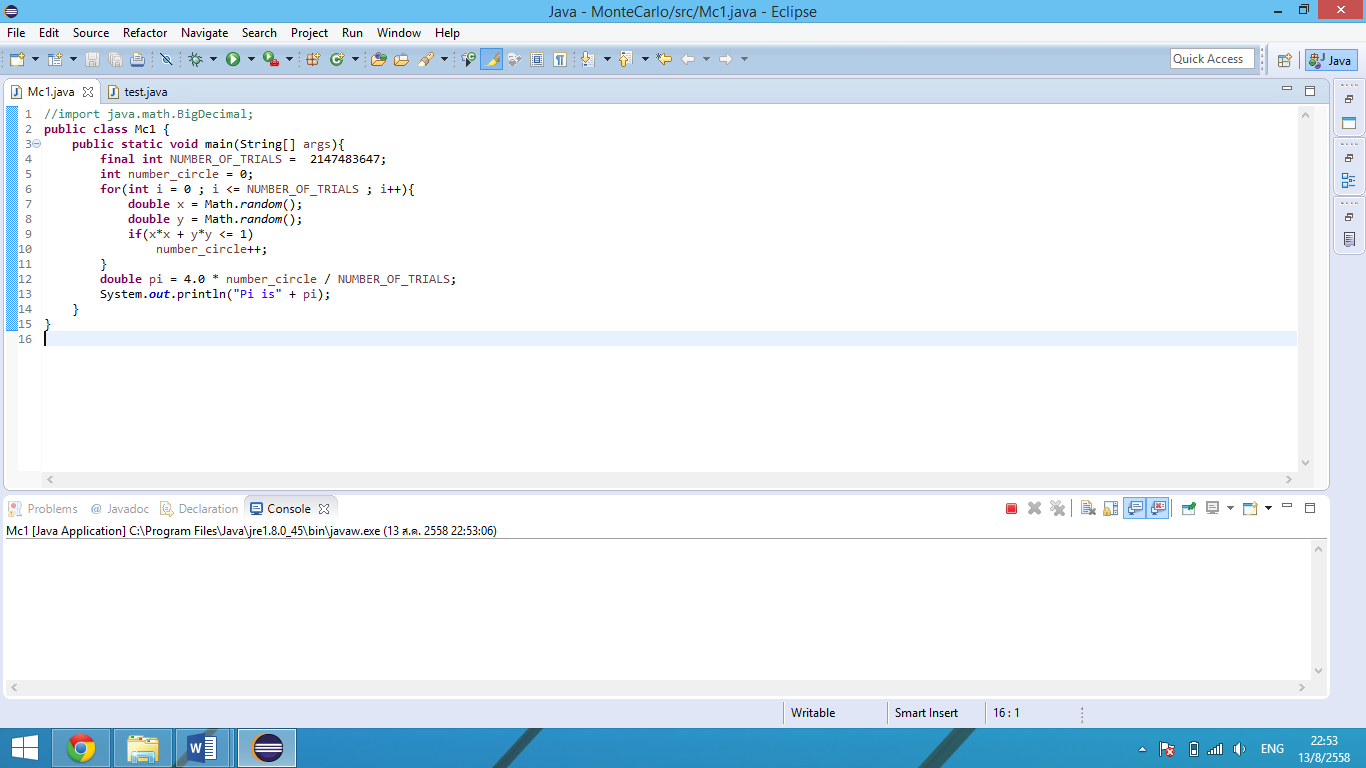
do number\_ circle++;

pi << 4\*(number\_ circle / NUMBER\_OF\_TRIALS);

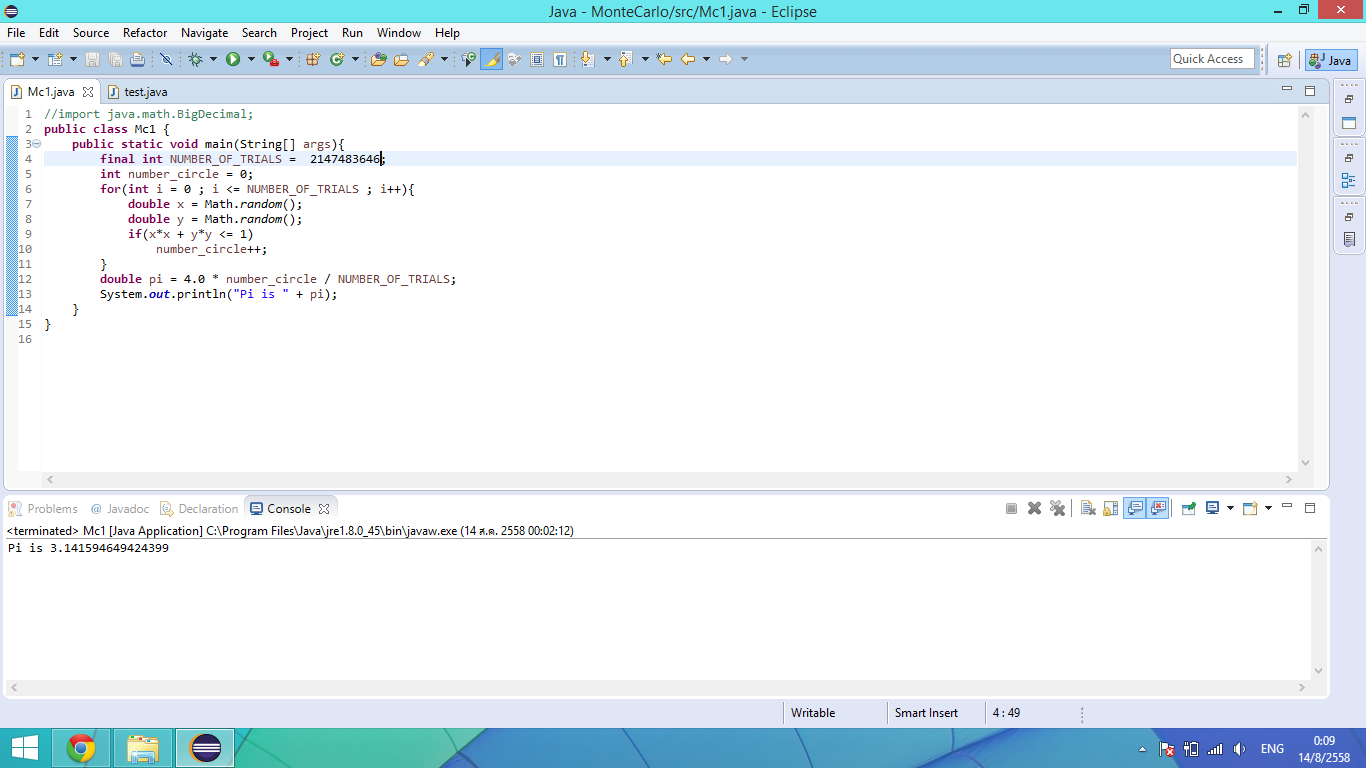
println(pi);

End;

# ซอร์สโค้ด



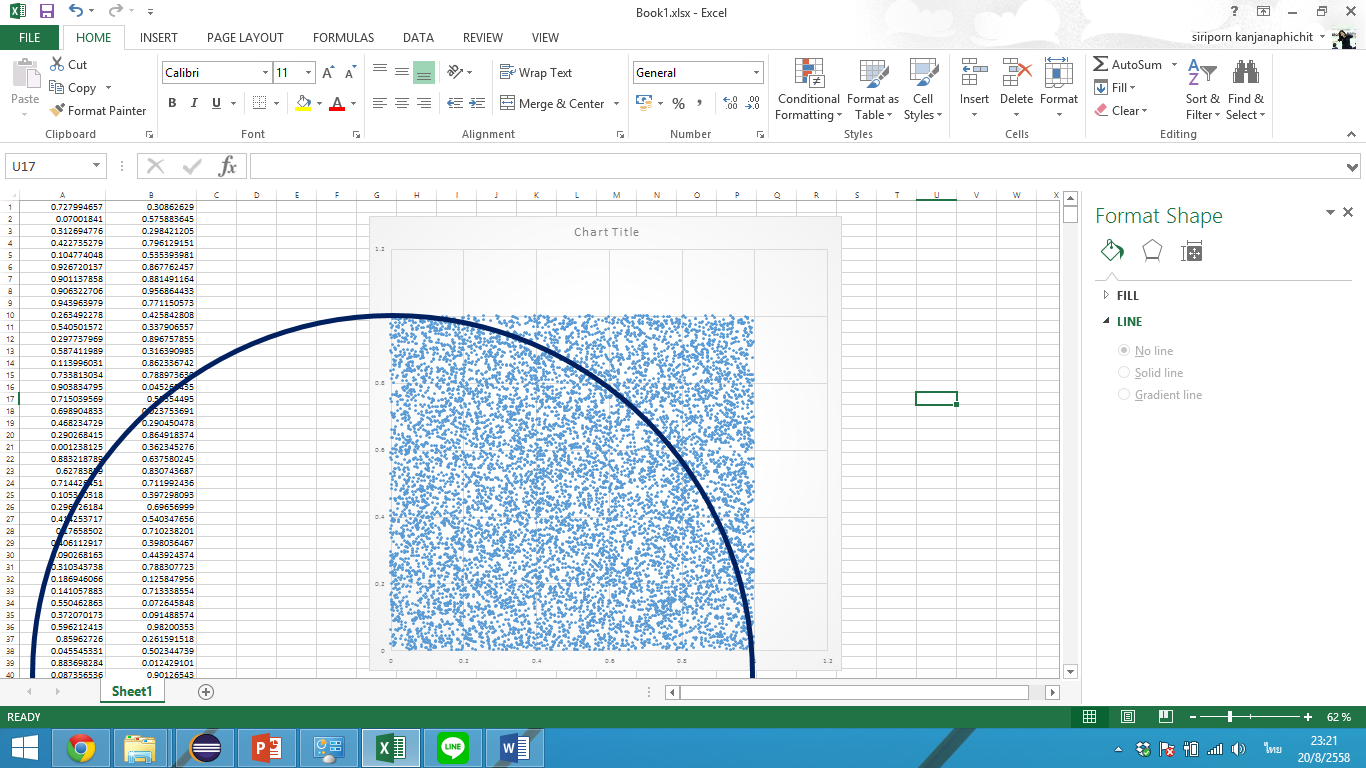
# ผลการรัน



ค่าจริง π = 3.14159265358979323846264338327950288419716939937510 มีความแม่นยำ 6 หลัก

# กราฟ

วาดจุดจากค่าข้อมูลที่ได้โดยทำการสุ่ม 10000 ค่า เพื่อนำมาทดลองวาดกราฟ



# แนวคิด หลักการ

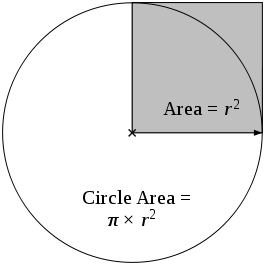
Monte Carlo method เป็นวิธีการหนึ่งที่ป็นวิธีการทางคณิตศาสตร์ซึ่งทำได้โดยการสุ่มค่าจากคอมพิวเตอร์เพื่อทำการแก้ปัญหาหลายอย่างทางคณิตศาสตร์หรือฟิสิกส์ โดยมีหลักการดันนี้

กำหนดให้ A = พื้นที่วงกลม

จากสูตร

กำหนดให้ R = 1 ;

จากสมการวงกลม



จากรูป พิจารณาจาก quadrant ที่ 1 จะได้ว่า พื้นที่แรเงา = ซึ่งหลักการนี้จะคิดจาก วงกลม 1 หน่วย (r = 1) จะได้ว่า พื้นที่แรเงา = 1 ดันนั้น พื้นที่สี่เหลี่ยมทั้งหมดที่ครอบคลุมพื้นที่วงกลม 1 หน่วย คือ 1 \* 4 = 4 หน่วย จากแนวคิดของ Monte Carlo สามารถนำมาใช้หาค่า pi ได้โดยการสุ่มค่า x และ y ใน quadrant ที่ 1 ทำให้ได้จุดขึ้นมา 1 จุด จากนั้นนับ อัตราส่วนของจุดในวงกลม ต่อ จุดที่สุ่มทั้งหมดจะได้ พื้นที่ ¼ ของค่า pi เช่นถ้าสุ่มจุดมา 10 จุด อยู่ภายในวงกลม 9 จุด อัตราส่วนคือ 9/10 เป็นต้น เมื่อเพิ่มจุดในการสุ่มจะทำให้ได้ค่า pi ที่แม่นยำมากยิ่งขึ้น

จากสูตร ปีทาโกรัส

จุดจะอยู่ในวงกลมก็ต่อเมื่อ ค่า d มีค่าน้อยกว่าหรือเท่ากับ 1

# ขั้นตอนการทำงาน

1. สุ่มจุด x และ y ที่อยู่ระหว่าง 0-1 โดยใช้ฟังก์ชั่น Math. Random
2. หาค่า d จากสูตรปีทาโกรัส
3. ถ้า d มีค่าน้อยกว่าหรือเท่ากับ 1 ให้ตัวแปรชื่อ number\_ circle เพิ่มค่าขึ้น 1
4. ทำวนข้อ 1-3 โดยจำนวนรอบขึ้นอยู่กับค่าของตัวแปรชื่อ NUMBER\_OF\_TRIALS
5. หาค่า pi ได้โดย นำ 4\* (number\_ circle / NUMBER\_OF\_TRIALS)

# ข้อจำกัด

* จำนวนรอบของการวนลูปจะต้องไม่เกินค่าสูงสุดที่ตัวแปรชนิด int สามารถเก็บได้ คือ 2,147,483,647

ข้อควรระวัง

* การสุ่มค่าอาจเกิดการซ้ำการได้แต่มีกาสเกิดขึ้นน้อยมาก

# แหล่งอ้างอิง

<https://soowoi.wordpress.com/2011/01/16/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A1%E0%B8%B2%E0%B8%93%E0%B8%84%E0%B9%88%E0%B8%B2-pi-%E0%B9%82%E0%B8%94%E0%B8%A2%E0%B9%83%E0%B8%8A%E0%B9%89-monte-carlo-method-%E0%B8%95/>

<http://newton.ex.ac.uk/research/qsystems/collabs/pi/pi6.txt>