

# ใบงานการทดลองที่ 8

## เรื่อง Wrapper Class และ Math Class

### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการติดต่อระหว่างงาน และการเรียกใช้งานคลาสสำคัญทางคณิตศาสตร์
- 1.2. รู้และเข้าใจการใช้งานคลาส Wrapper

### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

### 3. ทฤษฎีการทดลอง

#### 3.1. Primitive Data Type คืออะไร ?

ชนิดข้อมูลดั้งเดิมเป็นชนิดข้อมูลพื้นฐานที่ภาษาโปรแกรมให้มาเป็นแบบเอกสารสำเร็จรูปสำหรับสร้างโครงสร้างข้อมูลที่ซับซ้อนมากขึ้น  
ตัวอย่างของชนิดข้อมูลดั้งเดิมใน Java ได้แก่ boolean, byte, short, int, long, float, double, และ char  
คลาสของ Wrapper เช่น Integer, Double, Character เป็นต้น จัดเตรียมการแปลงออบเจกต์ของประเภทข้อมูลดั้งเดิมและฟังก์ชันเพิ่มเติม  
คลาส Wrapper สามารถใช้เพื่อแปลงชนิดข้อมูลดั้งเดิมเป็นออบเจกต์ แยกวิเคราะห์สตริงเป็นประเภทดั้งเดิม และดำเนินการทางคณิตศาสตร์และตรรกะต่างๆ

#### 3.2. Wrapper Class คืออะไร? มีอะไรบ้าง? อธิบายพร้อมยกตัวอย่างประกอบ

คลาส Wrapper นำเสนอการแสดงผลประเภทข้อมูลดั้งเดิมเชิงวัตถุใน Java และอนุญาตให้ใช้เป็นวัตถุด้วยวิธีการที่มีประโยชน์  
ตัวอย่างคลาสของ wrapper ได้แก่ Boolean, Byte, Short, Integer, Long, Float, Double และ Character  
คลาสคณิตศาสตร์มีฟังก์ชันทางคณิตศาสตร์ที่มีประโยชน์สำหรับชนิดข้อมูลดั้งเดิมใน Java  
คลาส Wrapper และเมธอดคลาส Math มีประโยชน์สำหรับการดำเนินการกับชนิดข้อมูลดั้งเดิมในลักษณะเชิงวัตถุ

#### 3.3. ยกตัวอย่าง Method ที่สามารถเรียกใช้งานได้ใน Wrapper Class

valueOf() - method นี้ส่งคืนวัตถุ wrapper ใหม่ที่แสดงถึงค่าดั้งเดิมที่ระบุ ตัวอย่างเช่น Integer.valueOf(42) ส่งคืนวัตถุจำนวนเต็มใหม่ที่มีค่า 42  
parseInt() - เมธอดนี้มีอยู่ในคลาส Integer และใช้เพื่อแยกวิเคราะห์สตริงและส่งกลับค่าจำนวนเต็ม ตัวอย่างเช่น Integer.parseInt("42") ส่งคืนค่าจำนวนเต็ม 42  
compareTo() - วิธีการนี้เปรียบเทียบค่าของวัตถุ wrapper ปัจจุบันกับวัตถุ wrapper อื่นและส่งกลับจำนวนเต็มแทนผลการเปรียบเทียบ ตัวอย่างเช่น  
Integer.valueOf(10).compareTo(Integer.valueOf(5)) ส่งกลับ 1 แสดงว่า 10 มากกว่า 5

#### 3.4. คำสั่ง Math.max( x, y ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เมธอด Math.max(x, y) หาค่า x และ y ที่กำหนดได้สูงสุดสองค่า ตัวอย่างเช่น Math.max(5, 10) คืนค่า 10 method  
นี้ยังสามารถใช้กับประเภทข้อมูลอื่นๆ เช่น double, float และ long ตัวอย่างเช่น Math.max(2.5, 4.7) คืนค่า 4.7  
ผลลัพธ์ของการเรียกเมธอดจะมีความมากกว่าสองค่าเสมอ

#### 3.5. คำสั่ง Math.min( x, y ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เมธอด Math.min(x, y) หาค่า x และ y ที่กำหนดอย่างน้อยสองค่า ตัวอย่างเช่น Math.min(5, 10) คืนค่า 5  
เมธอดนี้ยังสามารถใช้กับประเภทข้อมูลอื่นๆ เช่น double, float และ long ตัวอย่างเช่น Math.min(2.5, 4.7) คืนค่า 2.5  
ผลลัพธ์ของการเรียกใช้เมธอดจะน้อยกว่าค่าทั้งสองเสมอ

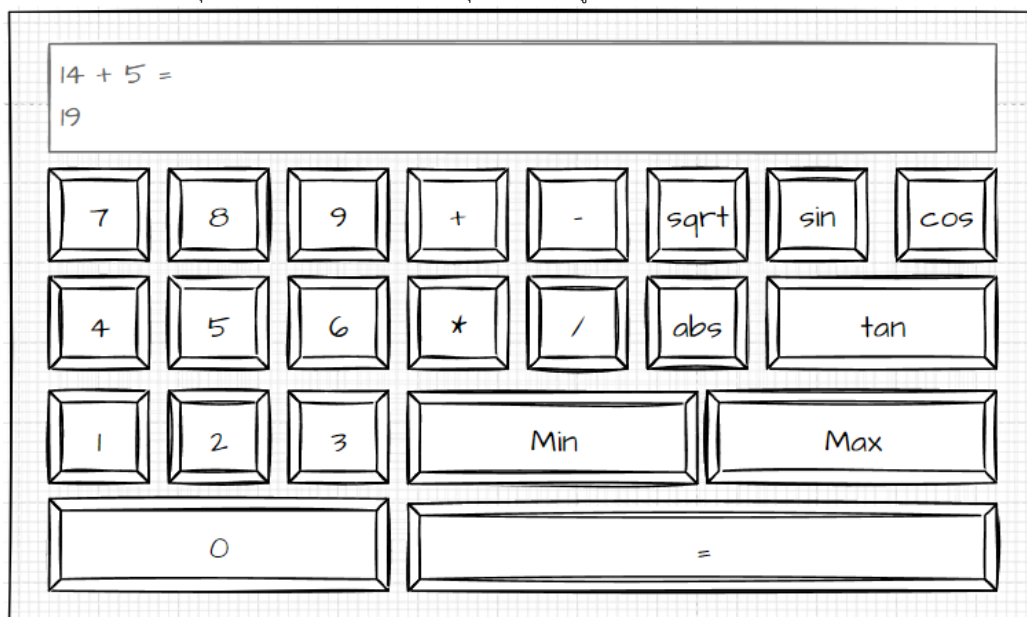
3.6. คำสั่ง `Math.sqrt(x)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ  
 method `Math.sqrt(x)` ใช้ในการหาค่าที่สองของค่าที่กำหนด `x` ตัวอย่างเช่น `Math.sqrt(16)` คืนค่า 4 method นี้สามารถใช้กับข้อมูลประเภทอื่น เช่น `float` และ `long` ตัวอย่างเช่น `Math.sqrt(25.0f)` ส่งคืน `5.0f` และ `Math.sqrt(81L)` ส่งคืน `9.0`  
 ผลลัพธ์ของการเรียกใช้เมธอดจะเป็นค่ารากที่สองของค่าที่กำหนดเสมอ

3.7. คำสั่ง `Math.abs(x)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ  
 เมธอด `Math.abs(x)` ส่งกลับค่าสัมบูรณ์ของค่าที่กำหนด `x` ตัวอย่างเช่น `Math.abs(-5)` คืนค่า 5 เมธอดนี้ใช้ได้กับข้อมูลประเภทอื่นๆ เช่น `double` และ `float` ตัวอย่างเช่น `Math.abs(-7.5)` ส่งคืน `7.5` และ `Math.abs(-2.5f)` ส่งคืน `2.5f` ผลลัพธ์ของการเรียกใช้เมธอดจะเป็นค่าสัมบูรณ์ของค่าที่กำหนดเสมอ

3.8. คำสั่ง `Math.random()` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ  
 เมธอด `Math.random()` สร้างตัวเลขสุ่มระหว่าง 0 (รวม) และ 1 (พิเศษ) ตัวอย่างเช่น `Math.random()` ส่งคืนค่าแบบสุ่มระหว่าง 0.0 ถึง 1.0  
 หากต้องการสร้างตัวเลขสุ่มภายในช่วงที่กำหนด สามารถใช้สูตร `Math.random() * (max - min) + min` ตัวอย่างเช่น `(int) (Math.random() * (10 - 1 + 1)) + 1` จะสร้างจำนวนเต็มแบบสุ่มระหว่าง 1 ถึง 10

#### 4. ลำดับขั้นการปฏิบัติการ

- 4.1. จงพัฒนาและปรับปรุงโปรแกรมเครื่องคิดเลขจากใบงานทดลองที่ 5 โดยการเพิ่มปุ่มให้กับเครื่องคิดเลขที่มีความสามารถเพิ่มเติมดังต่อไปนี้
  - 4.1.1. ปุ่ม `sqrt` เพื่อใช้ในการหาค่ารากที่ 2 ของตัวเลขที่ผู้ใช้กรอก
  - 4.1.2. ปุ่ม `abs` เพื่อใช้ในการหาค่าสัมบูรณ์ของตัวเลขที่ผู้ใช้กรอก
  - 4.1.3. ปุ่ม `sin` เพื่อใช้ในการหา Sin ของตัวเลขที่ผู้ใช้กรอก
  - 4.1.4. ปุ่ม `cos` เพื่อใช้ในการหา Cos ของตัวเลขที่ผู้ใช้กรอก
  - 4.1.5. ปุ่ม `tan` เพื่อใช้ในการหา Tan ของตัวเลขที่ผู้ใช้กรอก
  - 4.1.6. ปุ่ม `Min` เพื่อใช้ในการหาค่าน้อยที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า
  - 4.1.7. ปุ่ม `Max` เพื่อใช้ในการหาค่าที่มากที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า



โค้ดโปรแกรมภายในปุ่มเครื่องหมายเท่ากับ (เฉพาะส่วนที่เพิ่มเข้ามาในใบงานการทดลองนี้เท่านั้น)

## 5. สรุปผลการปฏิบัติการ

.....

.....

.....

## 6. คำถามท้ายการทดลอง

### 6.1. Wrapper Class ควรใช้เมื่อใด? เพราะอะไร?

คลาสของ Wrapper จะใช้เมื่อเราต้องการแสดงชนิดข้อมูลดั้งเดิมเป็นอ็อบเจกต์ หรือเมื่อเราต้องการทำงานในลักษณะเชิงวัตถุมากขึ้น  
โดยทั่วไปจะใช้ในคอลเลกชัน การส่งผ่านอาร์กิวเมนต์ไปยังเมธอด และการเขียนโปรแกรมเชิงวัตถุ ตัวอย่างเช่น สามารถใช้คลาส Integer wrapper  
เพื่อเก็บค่าจำนวนเต็มในรายการหรือส่งผ่านเป็นอาร์กิวเมนต์ไปยังเมธอดที่คาดว่าจะรับอ็อบเจกต์ ในทำนองเดียวกัน คลาส Double wrapper  
สามารถใช้เพื่อจัดเก็บตัวเลขทศนิยมในรายการหรือส่งผ่านเป็นอาร์กิวเมนต์ไปยังเมธอด

### 6.2. ข้อควรระวังในการใช้ Math Class คืออะไร?

เมื่อใช้คลาส Math ใน Java จำเป็นต้องระมัดระวังเกี่ยวกับข้อผิดพลาดในการปิดเศษ การโอเวอร์โฟลว์และอันเดอร์โฟลว์ การพิจารณาประสิทธิภาพ  
และการจัดการค่า NaN และค่าอินฟินิตี้ ตัวอย่างเช่น เมื่อใช้ Math.pow() เพื่อคำนวณยกกำลัง เราจำเป็นต้องทราบข้อจำกัดของเลขคณิตทศนิยม  
ในทำนองเดียวกัน เมื่อใช้ Math.exp() เพื่อคำนวณฟังก์ชันเอกซโพเนนเชียล เราจำเป็นต้องระวังเกี่ยวกับโอเวอร์โฟลว์หรืออันเดอร์โฟลว์ที่อาจเกิดขึ้น  
สิ่งสำคัญคือต้องพิจารณาถึงประสิทธิภาพโดยนัยของการใช้ฟังก์ชันที่มีการคำนวณสูง เช่น Math.sin() หรือ Math.cos() สุดท้าย  
เราจำเป็นต้องจัดการกับกรณีที่ฟังก์ชัน Math ส่งกลับ NaN หรือ Infinity เช่น เมื่อส่งพารามิเตอร์อินพุตที่ไม่ถูกต้องไปยัง Math.log()  
หรือหารด้วยศูนย์ใน Math.atan2()