



# ก้าวสู่โลกแห่งวัตถุ

การเขียนโปรแกรมเชิงวัตถุและพื้นฐานภาษาจาวา

จากพิมพ์เขียวทางความคิด สู่การเขียนโค้ดในโลกจริง

# รากฐานของการคิดเชิงตรรกะ



## 1. Sequence



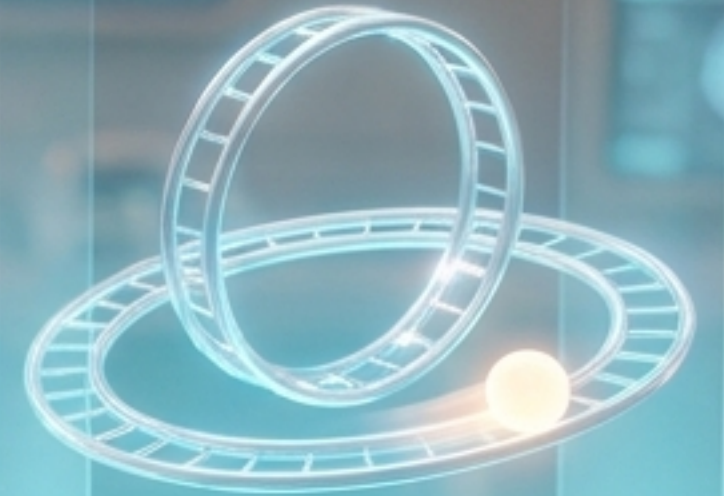
การทำงานแบบตามลำดับ:  
ทำงานจากบนลงล่าง  
มีจุดเริ่มต้นและจุดสิ้นสุดเดียว

## 2. Decision



การเลือกกระทำตามเงื่อนไข:  
ตัดสินใจจากเงื่อนไข  
(เช่น ถ้าคะแนน  $\geq 80$  ให้เกรด A)

## 3. Loop



การทำซ้ำ:  
ทำกระบวนการเดิมซ้ำๆ  
จนกว่าเงื่อนไขจะเป็นเท็จ

# จุดเปลี่ยนนกระบวนความคิด: จากกระบวนกา รสู่วัตตฤ



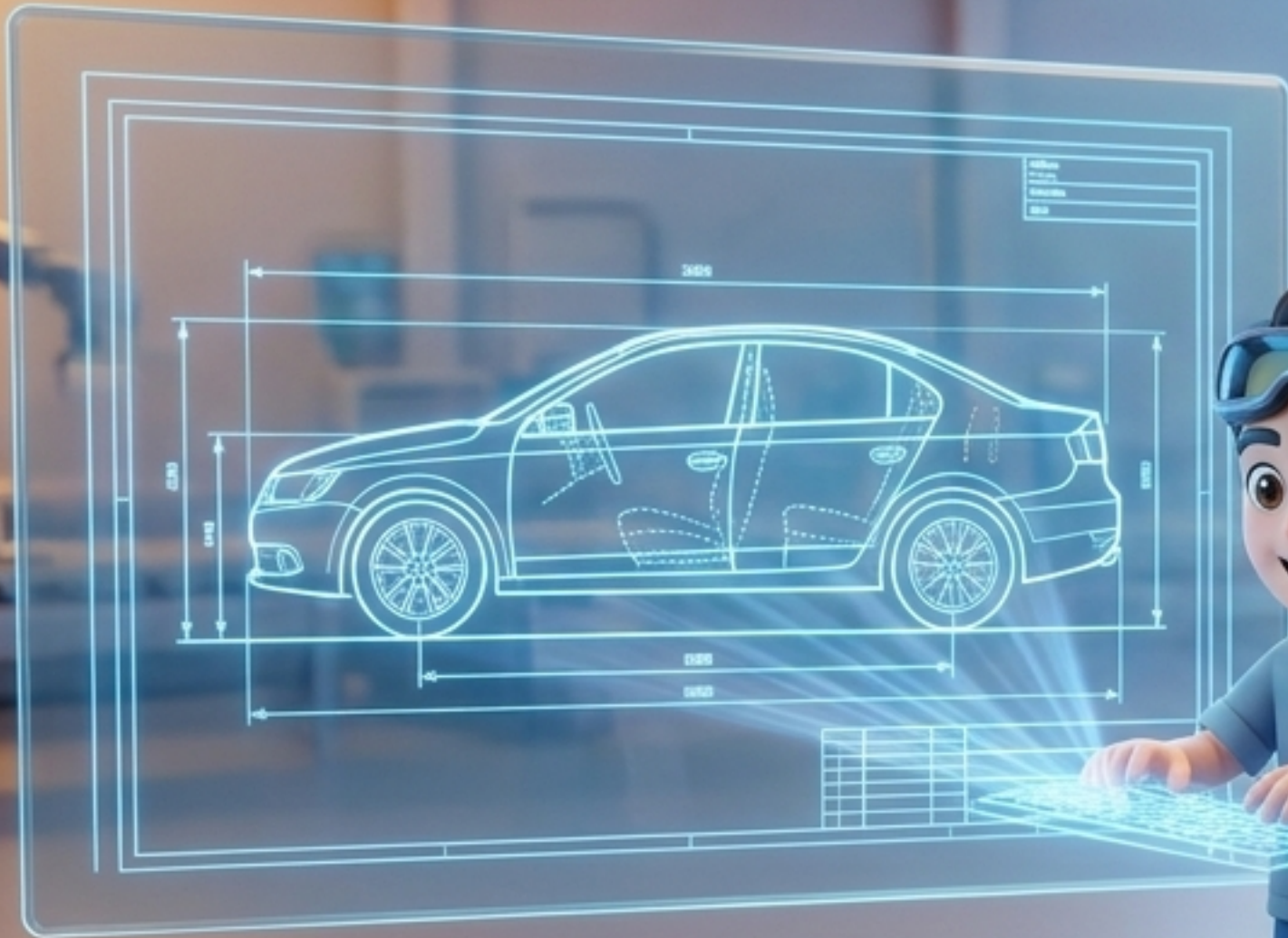
## การเขียนโปรแกรมเชิงกระบวนกา ร (Procedural)

- แบ่งตามลำดับขั้นตอนการทำงาน
- แยกข้อมูล (Data) ออกจากคำสั่ง (Function)
- ปัญหา: เมื่อสเกลใหญ่ขึ้น ข้อมูลกระจัดกระจาย ตรวจสอบยาก

## การเขียนโปรแกรมเชิงวัตตฤ (OOP)

- จำลองโลกจริงเป็นวัตตฤ
- รวมข้อมูลและพฤติกรรมไว้ด้วยกัน
- ข้อดี: พัฒนาเร็ว นำกลับมาใช้ใหม่ได้ง่าย (Reusable) ลดข้อผิดพลาด

# แม่พิมพ์ และ ความจริง (Class vs. Object)



## คลาส (Class)

- แม่พิมพ์ หรือ เค้าโครง (Blueprint)
- กำหนดลักษณะและพฤติกรรมพื้นฐาน
- ตัวอย่าง: คลาส รถยนต์



## ออปเจ็ก (Object)

- ตัวตนที่ถูกสร้างขึ้นจริงจากแม่พิมพ์
- 1 คลาส สามารถสร้างออปเจ็กต์ได้ไม่จำกัด
- ตัวอย่าง: รถยนต์สีแดงของนาย ก.,  
รถยนต์สีน้ำเงินของนาย ข.

# โครงสร้างภายในของวัตถุ (Anatomy of an Object)







**แอตทริบิวต์ (ข้อมูล)**

  SD 30  
ชื่อ 5  
HP 2010

รหัส ชื่อ นามสกุล เลขเฉลี่ย

**เมทอด (พฤติกรรม)**

ลงทะเบียนเรียน สอบ เดิน ดีใจ



**แอตทริบิวต์ (ข้อมูล)**

ชื่อ พันธุ์ สี ส่วนสูง น้ำหนัก

**เมทอด (พฤติกรรม)**

เห่า คลาน กระดิกหาง วิ่ง กิน



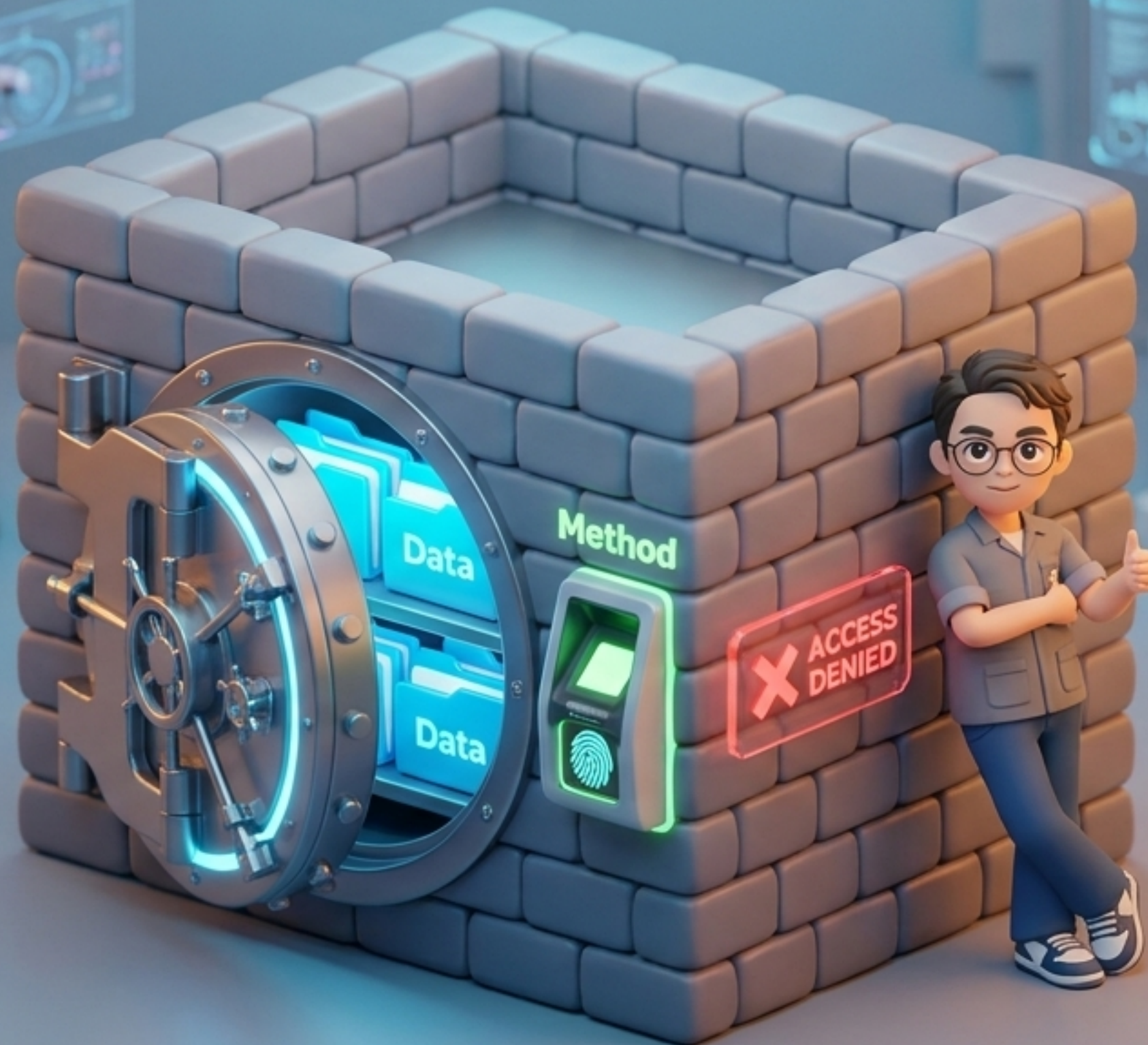
# เสาหลักที่ 1: การห่อหุ้ม (Encapsulation)



สร้าง 'กำแพง'  
ป้องกันข้อมูล (Data)  
ไม่ให้ถูกแก้ไขโดยตรง



บุคคลภายนอกต้องติดต่อ  
ผ่านช่องทางที่อนุญาต  
(Method) เท่านั้น



เพิ่มความปลอดภัย  
และป้องกันระบบรวม



เพิ่มความปลอดภัย  
และป้องกันระบบรวม



# เสาหลักที่ 2: การสืบทอด (Inheritance)

คลาสแม่ (Employee)

รหัสพนักงาน (ID), ตำแหน่ง (Role)



คลาสลูก (Manager)

รหัสพนักงาน (ID), ตำแหน่ง (Role)

+ เพิ่มรถประจำตำแหน่ง (Company Car)



Definition:

การสร้างคลาสใหม่โดยรับเอาคุณสมบัติจากคลาสเดิมที่มีอยู่แล้ว และเพิ่มความสามารถใหม่เข้าไปได้ (Re-use ลดการเขียนโค้ดซ้ำ ประหยัดเวลา)



# เสาหลักที่ 3: การพ้องรูป (Polymorphism)



public void calculate(int x)  
(ปิดเศษทิ้ง)

public void  
calculate(double x)  
(คิดเงินละเอียด)



**Definition:**

หนึ่งรูปแบบการติดต่อ แต่ติดต่อ แต่ตอบสนองได้หลายรูปแบบ (Overloading / Overriding)  
สิ่งหนึ่งสามารถทำงานได้หลากหลายรูปแบบตามบริบท (Context)



# กำเนิดขุมพลังจาวา (The Genesis of Java)



**เป้าหมายสูงสุด: ต้องเป็นภาษาที่ไม่ยึดติดกับฮาร์ดแวร์ (Platform-Independent)**

# ปัญหาการแปลภาษา: Compiler vs. Interpreter



## คอมไพเลอร์ (Compiler)

- แปลทีเดียวทั้งโปรแกรม ทำงานเร็วมาก
- ข้อเสีย: ยึดติดกับระบบปฏิบัติการ (Platform Specific)

## อินเตอร์พรีตเตอร์ (Interpreter)

- แปลทีละบรรทัดและทำงานทันที
- ข้อดี: ข้ามแพลตฟอร์มได้ง่าย
- ข้อเสีย: ทำงานช้ากว่าคอมไพเลอร์





# เวทมนตร์ของจาวา (The Execution Pipeline)

ภาษากลาง



Write Once -> Javac (Compiler) แปลงเป็น Bytecode  
-> Run Anywhere ผ่าน JVM (Interpreter) ประจำแต่ละเครื่อง

# สถาปัตยกรรมซ้อนทับ (Demystifying JDK, JRE, JVM)



## JDK (Java Development Kit)

ชุดเครื่องมือสำหรับผู้พัฒนา ประกอบด้วย  
JRE + Compiler + เครื่องมือดีบัก

## JRE (Java Runtime Environment)

สภาพแวดล้อมสำหรับรันโปรแกรม  
ประกอบด้วย JVM + Libraries

## JVM (Java Virtual Machine)

แกนกลาง ทำหน้าที่รันโค้ด (Interpreter)



# อาณาจักรของจาวาแพลตฟอร์ม (The Java Ecosystem)



**J2EE (Enterprise Edition)**  
สำหรับการพัฒนาระบบองค์กรขนาดใหญ่



**J2SE (Standard Edition)**  
แพลตฟอร์มมาตรฐานสำหรับแอปพลิเคชันทั่วไป

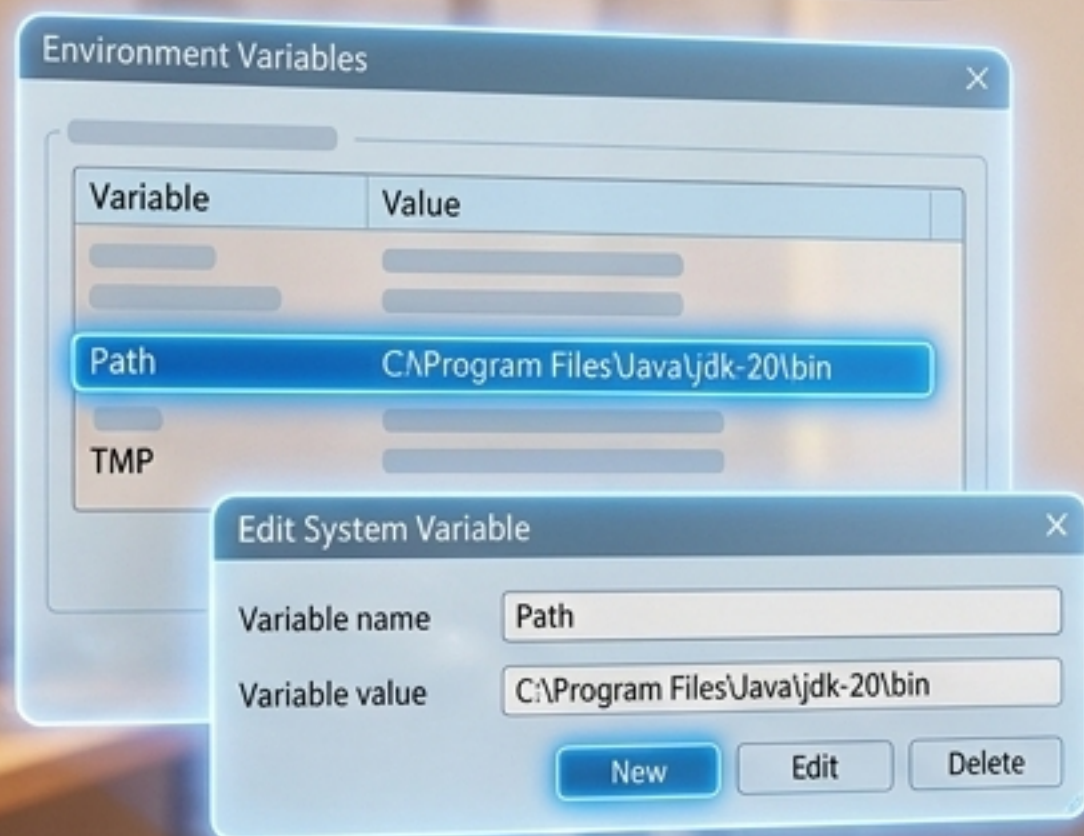


**J2ME (Micro Edition)**  
สำหรับอุปกรณ์ขนาดเล็ก เช่น โทรศัพท์มือถือ

# เตรียมความพร้อมสู่พื้นที่ปฏิบัติงาน (Workspace Setup)



## 1. การกำหนด PATH (เพื่อให้ระบบรู้จักคำสั่งคอมไพล์)



## 2. โปรแกรมเขียนโค้ด (IDE & Editors)



# โค้ดแรกในโลกจริง (The First Artifact: Hello World)



ทุกอย่างต้องถูกห่อหุ้มใน  
เค้าโครงแม่พิมพ์ Class เสมอ

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

จุดเริ่มต้น (Entry point)  
ที่ JVM จะมองหาเพื่อรันโปรแกรม

คำสั่งแสดงผลข้อความ  
ออกทางหน้าจอ

# Hello, World!

พิมพ์เขียวของคุณพร้อมแล้ว... เริ่มต้นสร้างโลกด้วยศิษย์บอร์ดของคุณ