

การเขียนโปรแกรมด้วย ภาษา Java >_

รายวิชา การเขียนโปรแกรมเบื้องต้น

ผู้สอน : นายศิลาภฤตินันท์ จำปาเรือง (อ.ศิลาปลาเผา)

สำหรับนักศึกษา ปวช.3





เขียนโค้ดครั้งเดียว
(Write Once)



Java Virtual
Machine (JVM)



รันได้ทุกที่ (Run Anywhere)

Java เป็นภาษาเชิงวัตถุ (OOP) ที่มีโครงสร้างชัดเจน ปลอดภัย
และใช้งานในระดับองค์กร (Enterprise) อย่างแพร่หลาย

การเตรียมเครื่องมือ (Environment Setup)



ชื่อคลาส (Class Name)
ต้องตรงกับชื่อไฟล์

จุดเริ่มต้นการทำงานของโปรแกรม
(Main Method)

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

คำสั่งแสดงผลข้อความออกทางหน้าจอ

ใช้ {} กลุ่มขอบเขตโค้ด
และปิดท้ายคำสั่งด้วย ; เสมอ

ชนิดข้อมูลและตัวแปร (Variables & Data Types)



ข้อมูลพื้นฐาน (Primitive)

- `int` : เก็บจำนวนเต็ม (เช่น 10, -5)
- `double` : เก็บเลขทศนิยม (เช่น 3.14)
- `boolean` : เก็บค่าความจริง (true / false)



อ็อบเจกต์ (Object)

- `String` : เก็บข้อความยาวๆ (เช่น "Expert-Programming-Tutor")

รูปแบบการประกาศตัวแปร: `[ชนิดข้อมูล] [ชื่อตัวแปร] = [ค่า];`

การควบคุมทิศทางโปรแกรม (Control Flow: เงื่อนไข)



```
if (score > 50) { ... } else { ... }
```



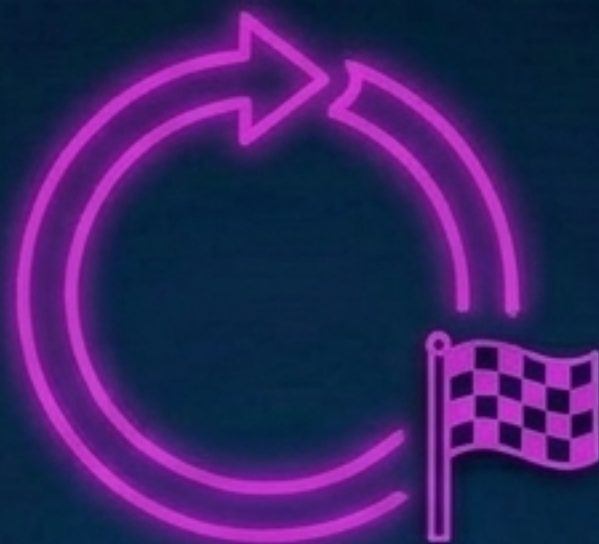
```
switch(day) { case 1: ... break; }
```

การทำซ้ำ (Loops: วงวน)



เช็คเงื่อนไขก่อนทำ
ถ้าจริงค่อยวนลูป

```
while (condition) {  
    ...  
}
```



ทำก่อนอย่างน้อย 1 ครั้ง
แล้วค่อยเช็คเงื่อนไข

```
do {  
    ...  
} while (condition);
```

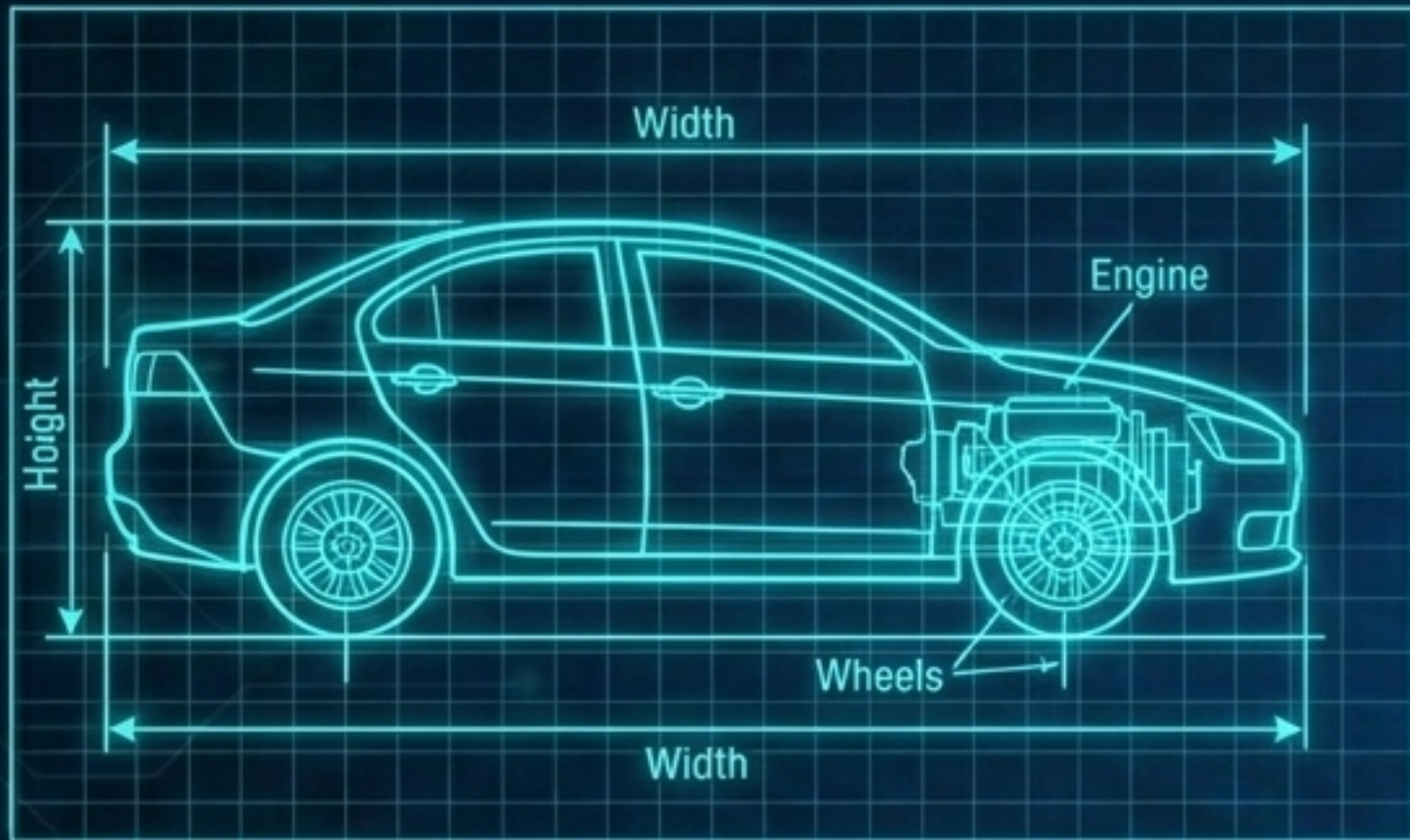


รู้จำนวนรอบที่แน่นอน
(กำหนดจุดเริ่ม, เงื่อนไข, การเพิ่มค่า)

```
for (int i=0; i<10; i++)  
    ...  
}
```



หัวใจของ Java: คลาสและอ็อบเจกต์ (Class vs Object)



Class (คลาส): พิมพ์เขียวหรือแบบแปลนที่กำหนดคุณสมบัติ (Variables) และการทำงาน (Methods)

Object (อ็อบเจกต์): สิ่งที่ถูกสร้างขึ้นมาจากคลาส แต่ละอ็อบเจกต์มีข้อมูลเป็นของตัวเอง (Instances)

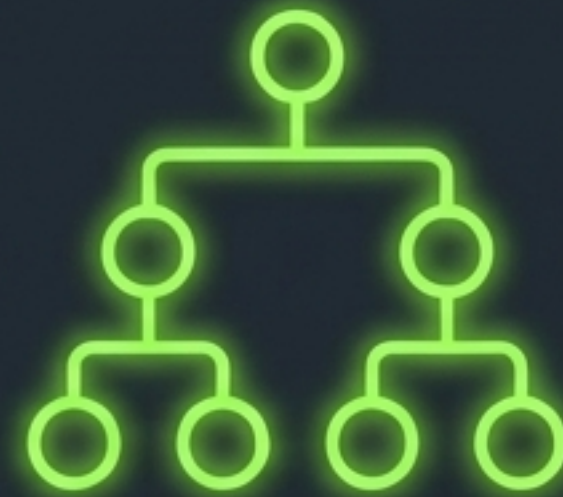
✨ ✨ คลาสเดียวสามารถสร้างอ็อบเจกต์ได้ไม่จำกัด! (Code Reuse) ✨ ✨

4 เสาหลักของการเขียนโปรแกรมเชิงวัตถุ (4 Pillars of OOP)



การห่อหุ้ม (Encapsulation)

ซ่อนข้อมูลภายในไม่ให้ถูกแก้ไข
โดยตรง ต้องเข้าถึงผ่าน
Methods ที่อนุญาตเท่านั้น



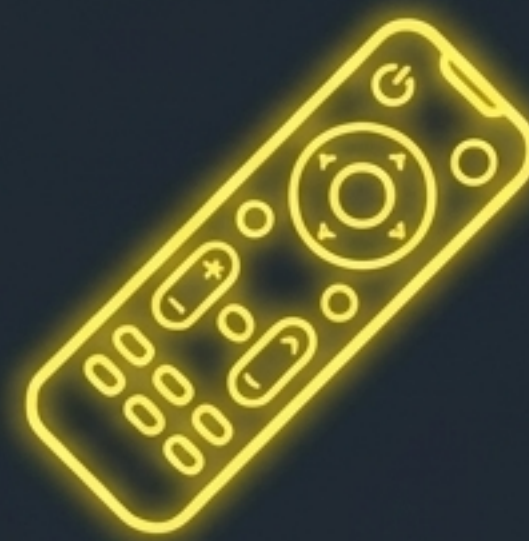
การสืบทอด (Inheritance)

คลาสลูกสามารถรับคุณสมบัติ
และคำสั่งจากคลาสแม่มาใช้
ต่อได้ทันที



พหุสัณฐาน (Polymorphism)

อ็อบเจกต์สามารถแสดงผลหรือ
ทำงานได้หลากหลายรูปแบบ
ผ่านชื่อเมธอดเดียวกัน



นามธรรม (Abstraction)

แสดงเฉพาะการทำงานที่จำเป็น
ให้ผู้ใช้เห็น โดยซ่อนความซับซ้อน
ของโค้ดเบื้องหลังไว้

การจัดการกลุ่มข้อมูลด้วย Collections Framework



List (รายการ):
เก็บข้อมูลตามลำดับที่ใส่
เข้าไป มีข้อมูลซ้ำกันได้

ตัวอย่างใช้งาน:
สร้างกระดานจัดอันดับผู้
เล่นเกม (Leaderboard)



Set (เซต):
ไม่มีลำดับที่แน่นอน และ
ห้ามมีข้อมูลซ้ำกัน

ตัวอย่างใช้งาน:
จัดเก็บ Username หรือ
Email ที่ไม่ให้ซ้ำกัน



Map (แผนที่/คู่ข้อมูล):
เก็บข้อมูลเป็นคู่
Key-Value (รหัส-ค่า)

ตัวอย่างใช้งาน:
เก็บรหัสสินค้า (SKU)
คู่กับจำนวนสินค้าคงคลัง



การเชื่อมต่อฐานข้อมูล (JDBC & MySQL)

```
jdbc:mysql://[hostname]:[port]/[database name]
```

Java Application
try-catch (SQLException e)

JDBC DriverManager

MySQL Database

ระวัง! ต้องใช้ try-catch เสมอ
เพื่อดักจับข้อผิดพลาด SQLException
หากการเชื่อมต่อล้มเหลว



เคล็ดลับการเขียนโค้ดระดับมืออาชีพ (Clean Code Principles)

	Don't ❌	Do ✅
การตั้งชื่อ (Naming)	ชื่อย่อที่ไม่สื่อความหมาย เช่น "uLst", "calcTotPrc()"	ชื่อที่ชัดเจน เช่น "userList", "calculateTotalPrice()"
ความซ้ำซ้อน (DRY Principle)	ก๊อปปี้โค้ดเดิมๆ ไปวางหลายที่ (Code Duplication)	ใช้หลัก DRY (Don't Repeat Yourself) สร้างเป็น Method เพื่อเรียกใช้ซ้ำ
การเขียนคอมเมนต์ (Comments)	คอมเมนต์อธิบายว่าโค้ดทำอะไร (What) ซึ่งโค้ดบอกอยู่แล้ว	คอมเมนต์อธิบายว่า "ทำไม" ถึงเขียนแบบนี้ (Why) หรือบอกเงื่อนไขทางธุรกิจ



การเขียนโปรแกรมไม่ใช่แค่ให้คอมพิวเตอร์เข้าใจ แต่ต้องให้เพื่อนร่วมทีมเข้าใจด้วย!