

บทที่ 0 แนะนำการโปรแกรมเชิงวัตถุ

บรรยายโดย ผศ.ดร.ธราวิเชษฐ์ ธิติจรูญโรจน์

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ออปเจ็ค คืออะไร

เมธอด

- กิน (Eat)
- นอน (Sleep)

คุณลักษณะ

- พันธุ์ (Breed)
- ขนาด (Size)
- อายุ (Age)



ออปเจ็ค หรือวัตถุ (Object) คือ สิ่งประกอบไปด้วย 2 ส่วนหลัก ได้แก่ คุณลักษณะ (Attribute) และ เมธอด (Method) ขณะที่ในมุมมองของผู้พัฒนาโปรแกรม ออปเจ็ค คือ โครงสร้างข้อมูลรูปแบบหนึ่ง ตัวแปร หรือฟังก์ชัน ซึ่งจัดเก็บบนหน่วยความจำ

ออปเจ็ค คืออะไร

เมธอด

- ????
- ????

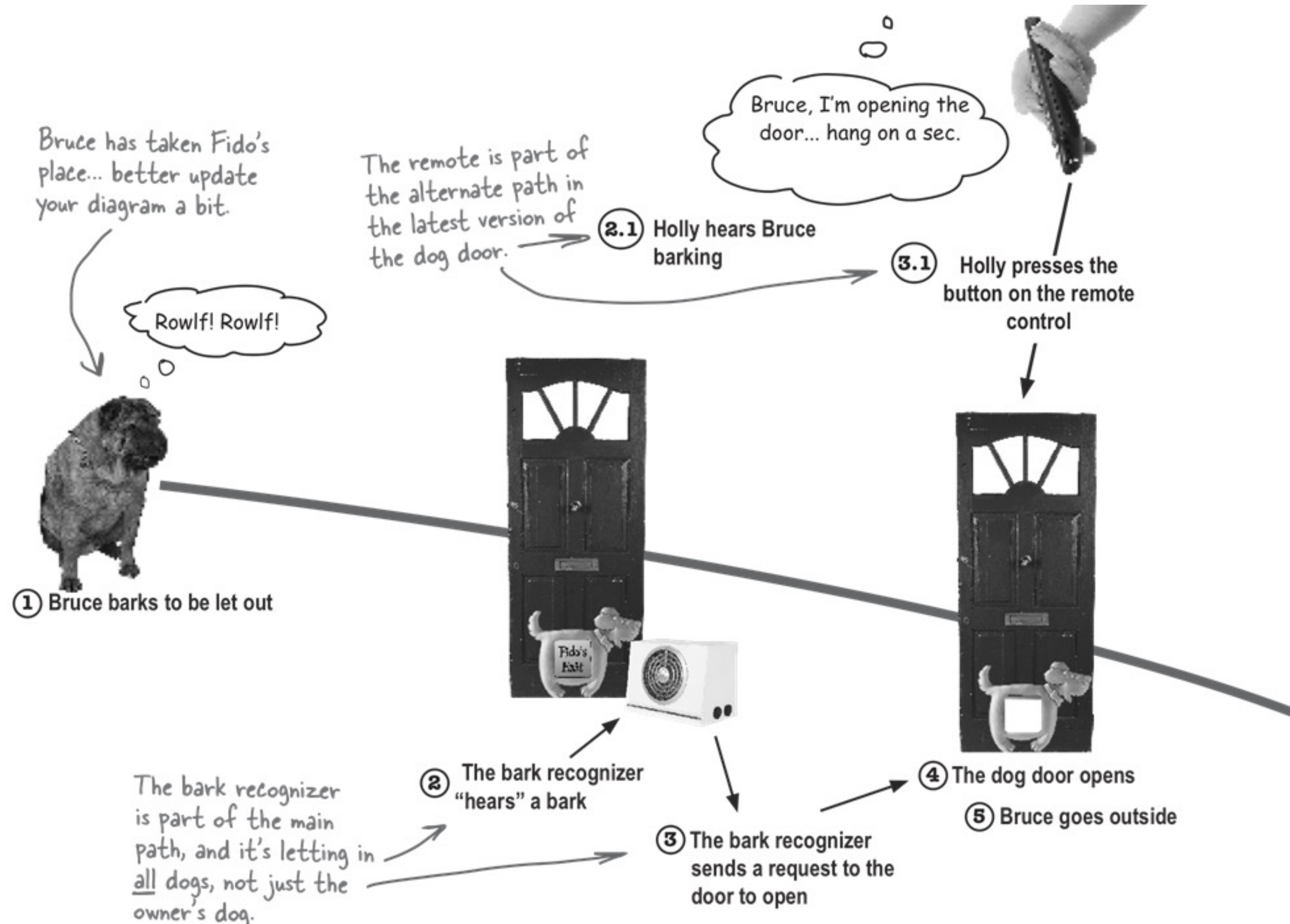


คุณลักษณะ

- ????
- ????
- ????

ออปเจ็ค หรือวัตถุ (Object) คือ สิ่งประกอบไปด้วย 2 ส่วนหลัก ได้แก่ คุณลักษณะ (Attribute) และ เมธอด (Method) ขณะที่ในมุมมองของผู้พัฒนาโปรแกรม ออปเจ็ค คือ โครงสร้างข้อมูลรูปแบบหนึ่ง ตัวแปร หรือฟังก์ชัน ซึ่งจัดเก็บบนหน่วยความจำ

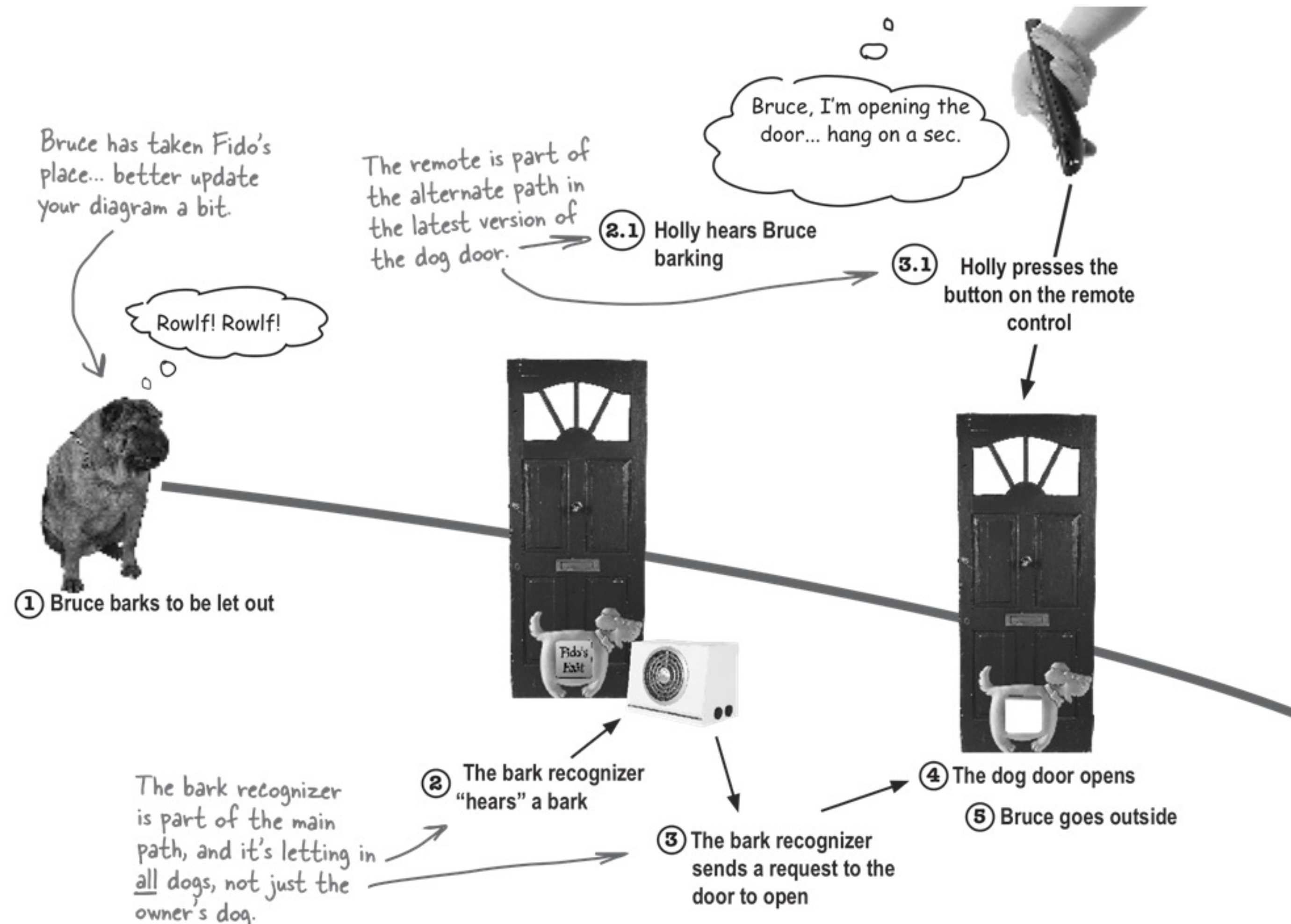
หลักการทำงานของภาษาเชิงออปเจ็ค



เรื่อง Open the dog door

ประตูสำหรับสุนัขจะเปิดก็ต่อเมื่อ (1) ได้รับสัญญาณจากรีโมต หรือ (2) เสียงสุนัขเห่า จากนั้นเมื่อประตูรับสัญญาณหรือเสียงดังกล่าวประตูจะเปิดอัตโนมัติ

หลักการทำงานของภาษาเชิงออปเจ็ค



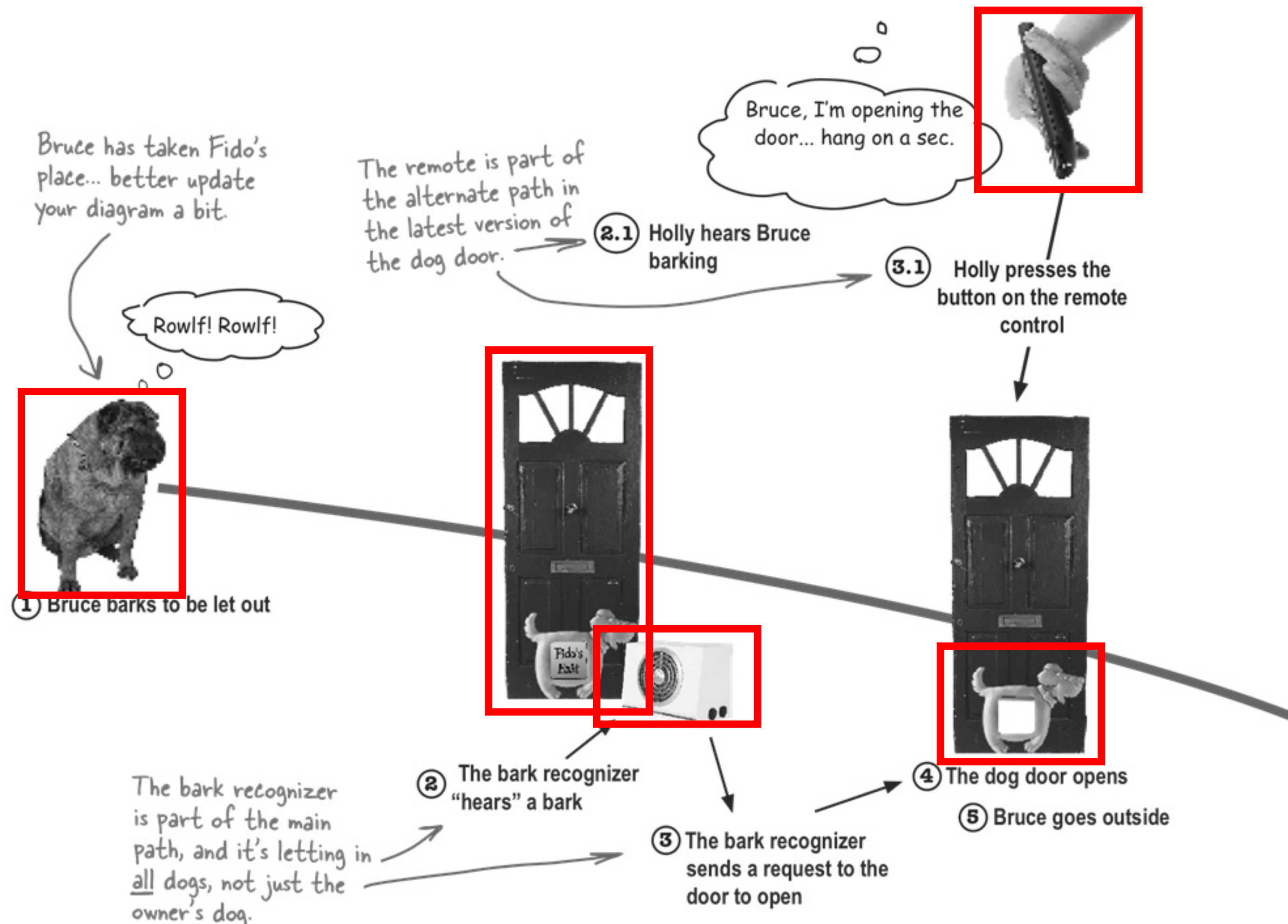
วิเคราะห์ปัญหาโดยมองปัญหาว่า
ประกอบไปด้วยออปเจ็คต่างๆ

จำลองคุณลักษณะและ
พฤติกรรมของออปเจ็ค

ออปเจ็คจะส่งข้อมูลกัน
โดยผ่านข่าวสาร (Message)

พิจารณาลำดับการทำงานและแบ่งการ
ทำงานของโปรแกรมตามหน้าที่

หลักการทำงานของภาษาเชิงออปเจ็ค



วิเคราะห์ปัญหาโดยมองปัญหาว่า
ประกอบไปด้วยออปเจ็คต่างๆ

จำลองคุณลักษณะและ
พฤติกรรมของออปเจ็ค

ออปเจ็คจะส่งข้อมูลกัน
โดยผ่านข่าวสาร (Message)

พิจารณาลำดับการทำงานและแบ่งการ
ทำงานของโปรแกรมตามหน้าที่

หลักการทำงานของภาษาเชิงออพเจ็ค

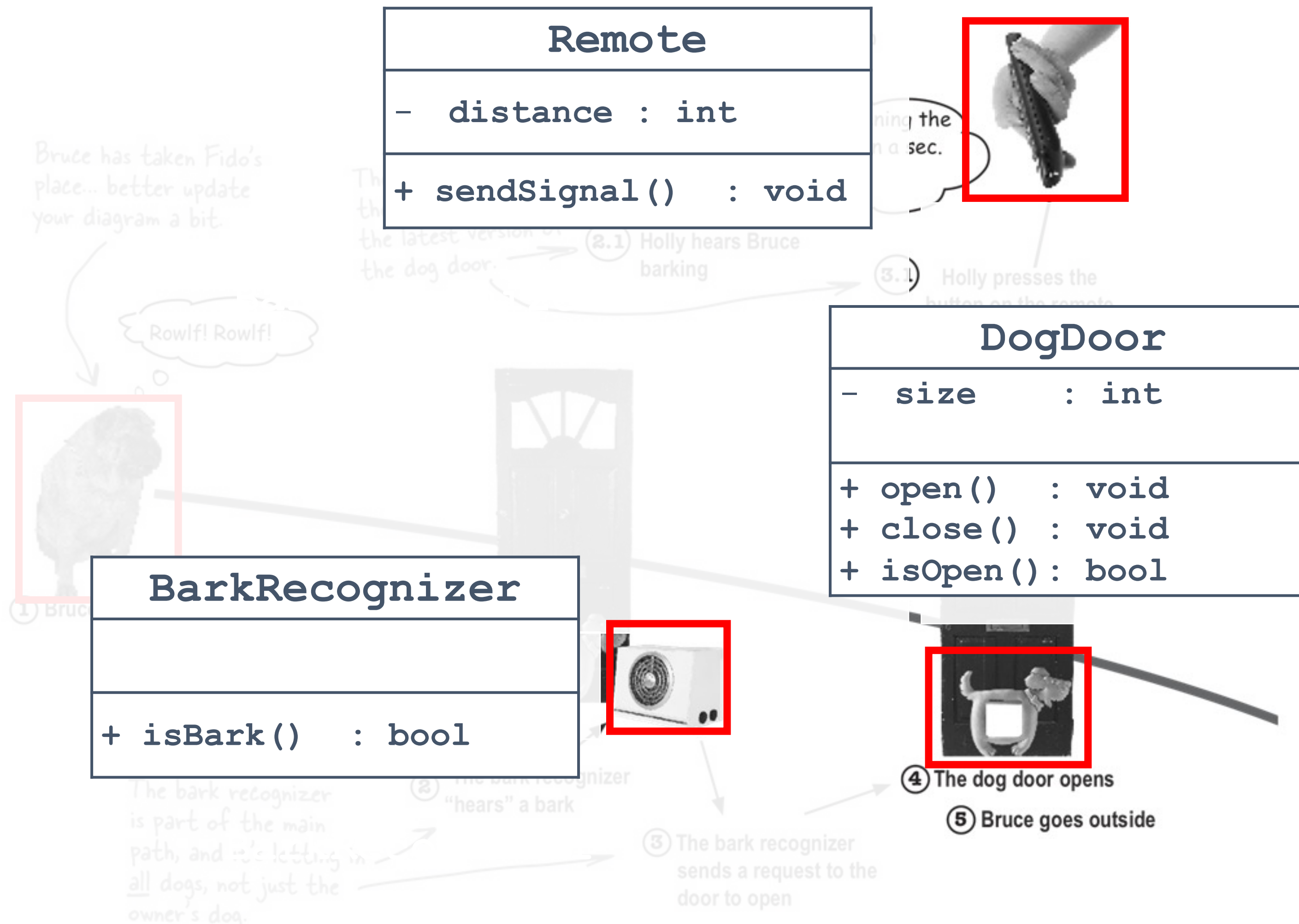


DogDoor

Remote

BarkRecognizer

หลักการทำงานของภาษาเชิงออบเจ็ค



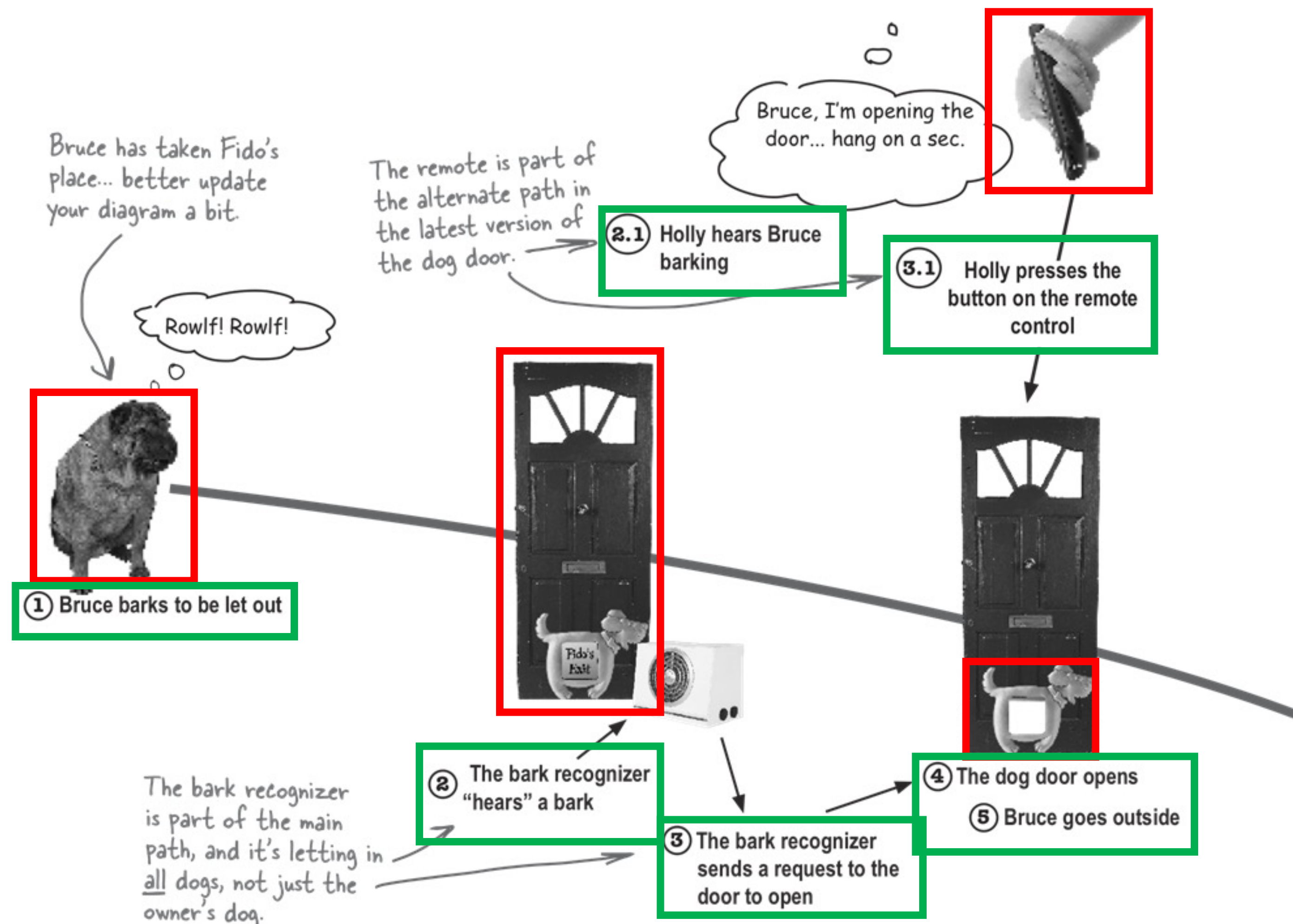
วิเคราะห์ปัญหาโดยมองปัญหาว่า
ประกอบไปด้วยออบเจ็คต่างๆ

จำลองคุณลักษณะและ
พฤติกรรมของออบเจ็ค

ออบเจ็คจะส่งข้อมูลกัน
โดยผ่านข่าวสาร (Message)

พิจารณาลำดับการทำงานและแบ่งการ
ทำงานของโปรแกรมตามหน้าที่

หลักการทำงานของภาษาเชิงออบเจ็ค



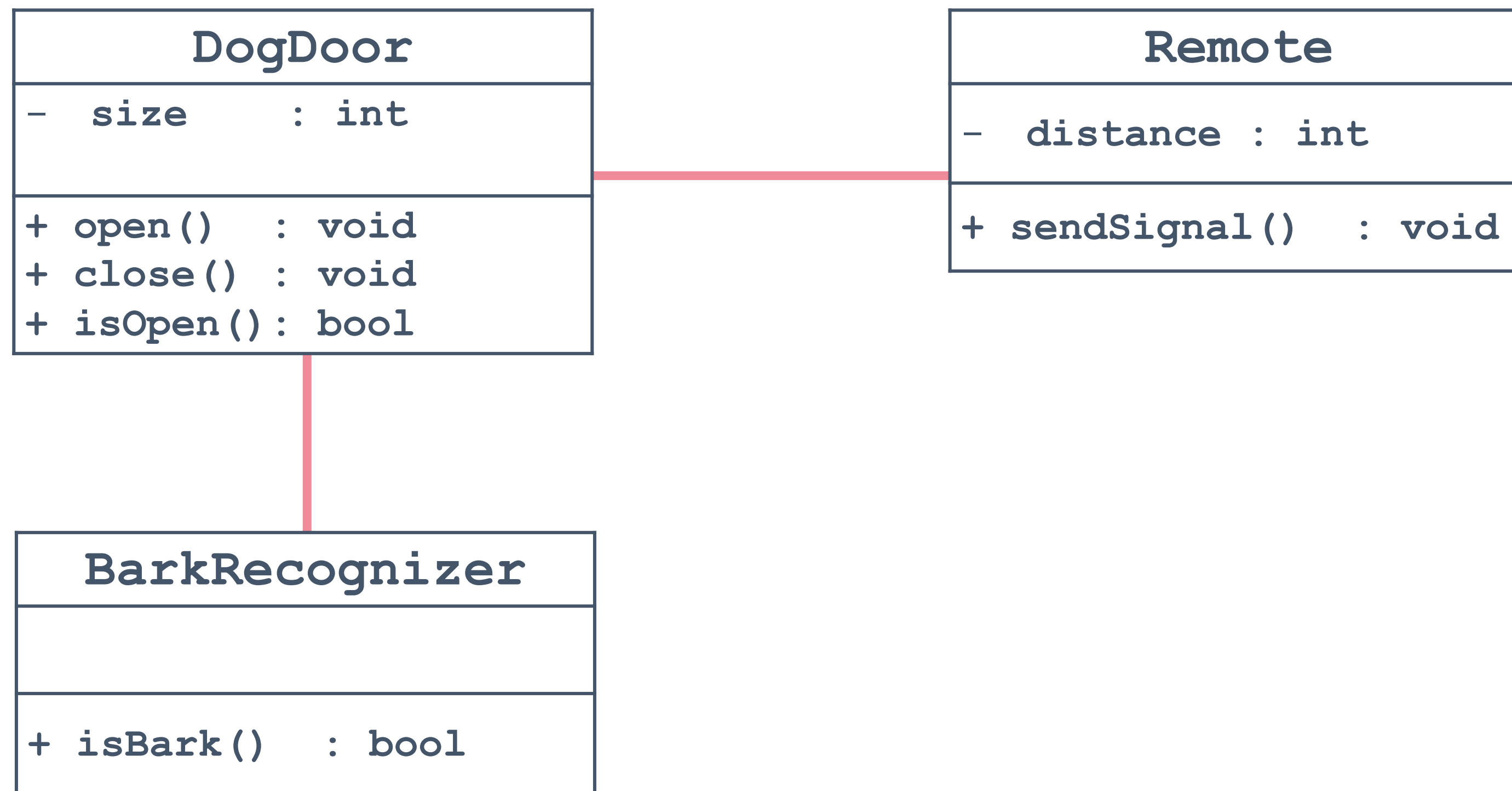
วิเคราะห์ปัญหาโดยมองปัญหาว่า
ประกอบไปด้วยออบเจ็คต่างๆ

จำลองคุณลักษณะและ
พฤติกรรมของออบเจ็ค

ออบเจ็คจะส่งข้อมูลกัน
โดยผ่านข่าวสาร (Message)

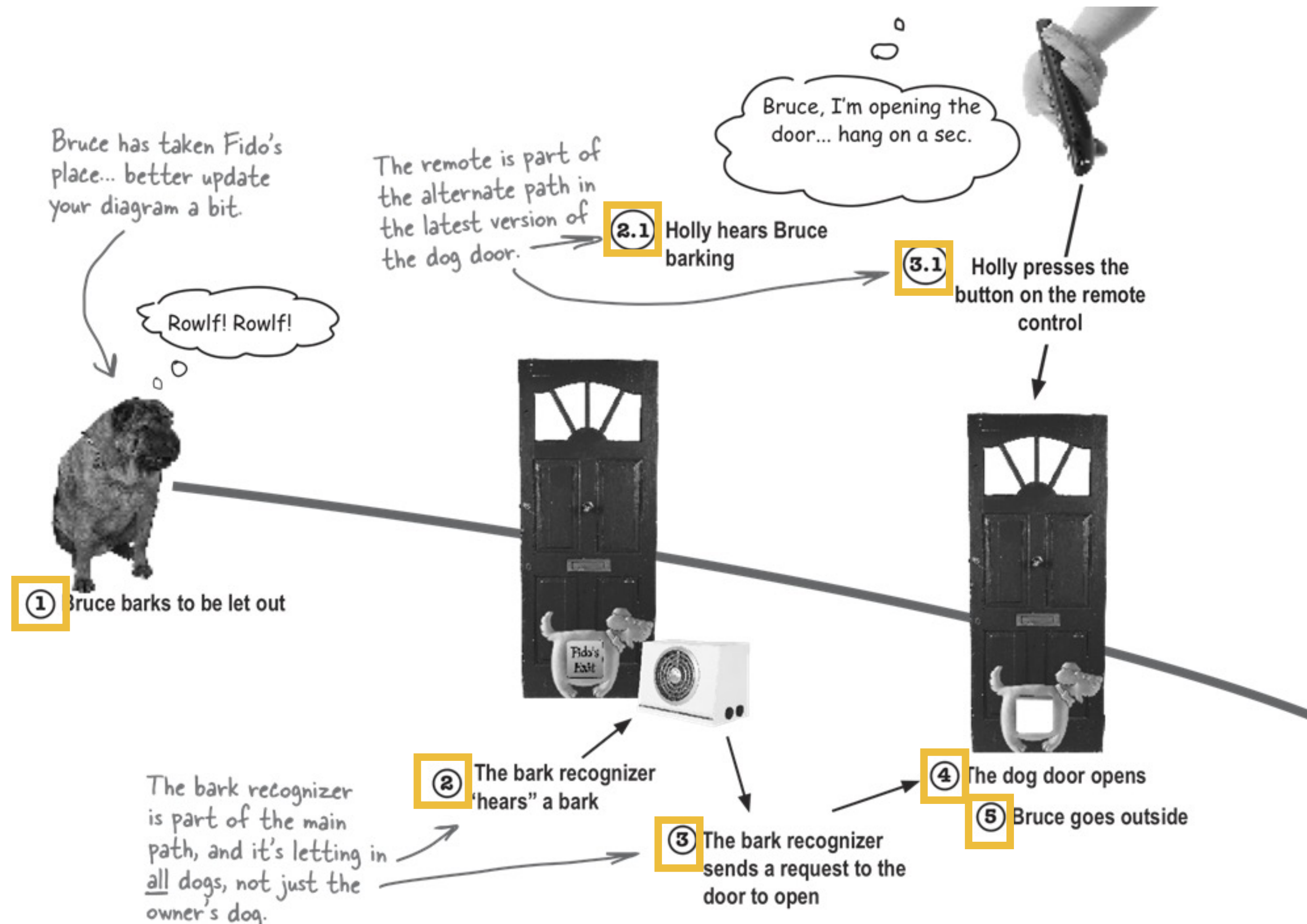
พิจารณาลำดับการทำงานและแบ่งการ
ทำงานของโปรแกรมตามหน้าที่

หลักการทำงานของภาษาเชิงออปเจ็ค



หลักการทำงานของภาษาเชิงออปเจ็ค

11



วิเคราะห์ปัญหาโดยมองปัญหาว่า
ประกอบไปด้วยออปเจ็คต่างๆ

จำลองคุณลักษณะและ
พฤติกรรมของออปเจ็ค

ออปเจ็คจะส่งข้อมูลกัน
โดยผ่านข่าวสาร (Message)

พิจารณาลำดับการทำงานและแบ่งการ
ทำงานของโปรแกรมตามหน้าที่

ข้อดีของการพัฒนาโปรแกรมเชิงออบเจ็ค

- แนวคิดการวิเคราะห์ปัญหาใกล้เคียงกับธรรมชาติของมนุษย์
- ระบบจริง (Real Life) แบ่งตามออบเจ็ค ไม่ได้ขึ้นอยู่กับฟังก์ชันการทำงาน **(เก่าแล้ว)**
- นำโปรแกรมกลับมาใช้ใหม่ได้ง่าย
- ขบวนการพัฒนาโปรแกรมทำได้รวดเร็วขึ้น **(บางมิติ)**
- ง่ายต่อการพัฒนาและแก้ไข

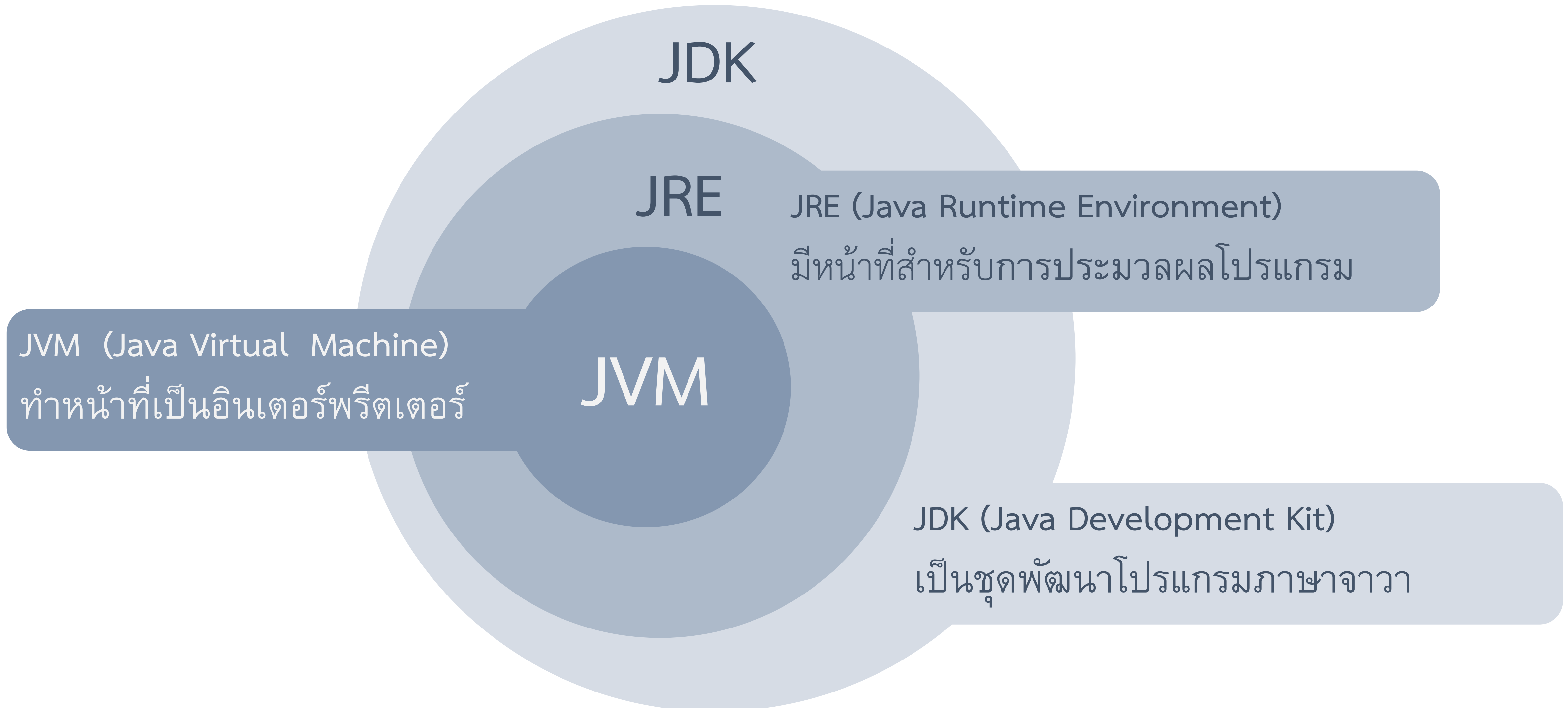


ประวัติภาษาจาวา



- พัฒนาขึ้นโดยทีมวิจัยของบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems)
- ปัจจุบันเป็นของ Oracle
- พัฒนามาจากโครงการที่ต้องการพัฒนาระบบซอฟต์แวร์เพื่อควบคุมเครื่องใช้ไฟฟ้าขนาดเล็กภายในบ้าน
- ชื่อเดิมคือภาษา Oak ต่อมาเปลี่ยนชื่อเป็นภาษาจาวา
- ภาษาจาวาเริ่มเป็นที่นิยมแพร่หลายในปี ค.ศ. 1995
- ภาษาจาวาเป็นภาษาที่**ไม่ขึ้นกับแพลตฟอร์ม (platform independent)**
- JDK 1.0 ประกาศใช้เมื่อปี 1996
- JDK เวอร์ชันปัจจุบันคือ 18 (March, 22nd 2022)

องค์ประกอบของเทคโนโลยีจาวา

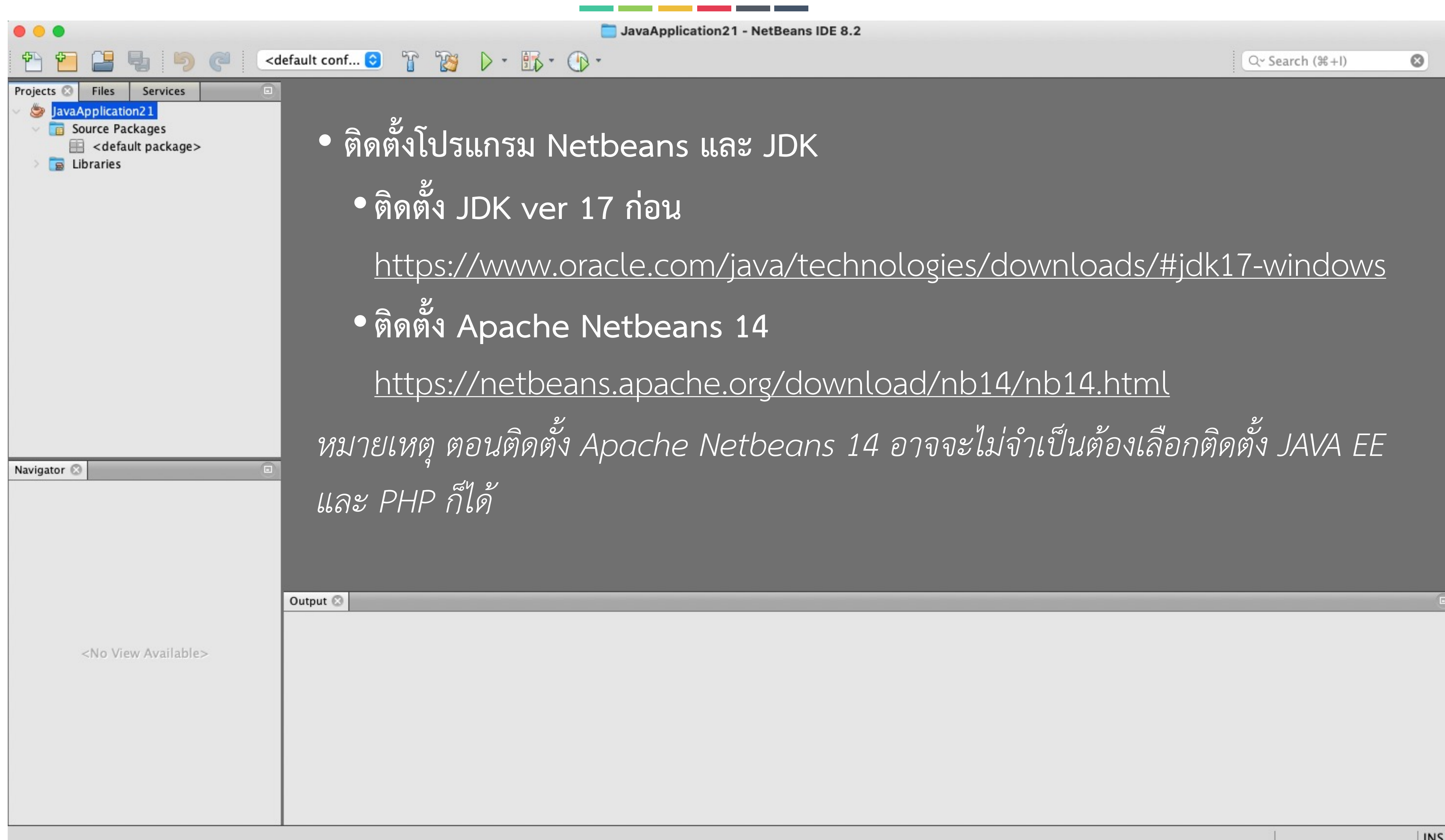


จุดเด่นของภาษาจาวา



- ความง่าย (simple)
- ภาษาเชิงออบเจ็ค (object oriented)
- การกระจาย (distributed)
- การป้องกันการผิดพลาด (robust)
- ความปลอดภัย (secure)
- สถาปัตยกรรมกลาง (architecture neutral)
- เคลื่อนย้ายง่าย (portable)
- อินเทอร์พรีต (interpreted)
- ประสิทธิภาพสูง (high performance)
- มัลติเธรด (multithreaded)
- พลวัต (dynamic)

ตัวอย่างการใช้ Java IDE



JavaApplication21 - NetBeans IDE 8.2

Search (⌘+I)

- ติดตั้งโปรแกรม Netbeans และ JDK
 - ติดตั้ง JDK ver 17 ก่อน
<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>
 - ติดตั้ง Apache Netbeans 14
<https://netbeans.apache.org/download/nb14/nb14.html>

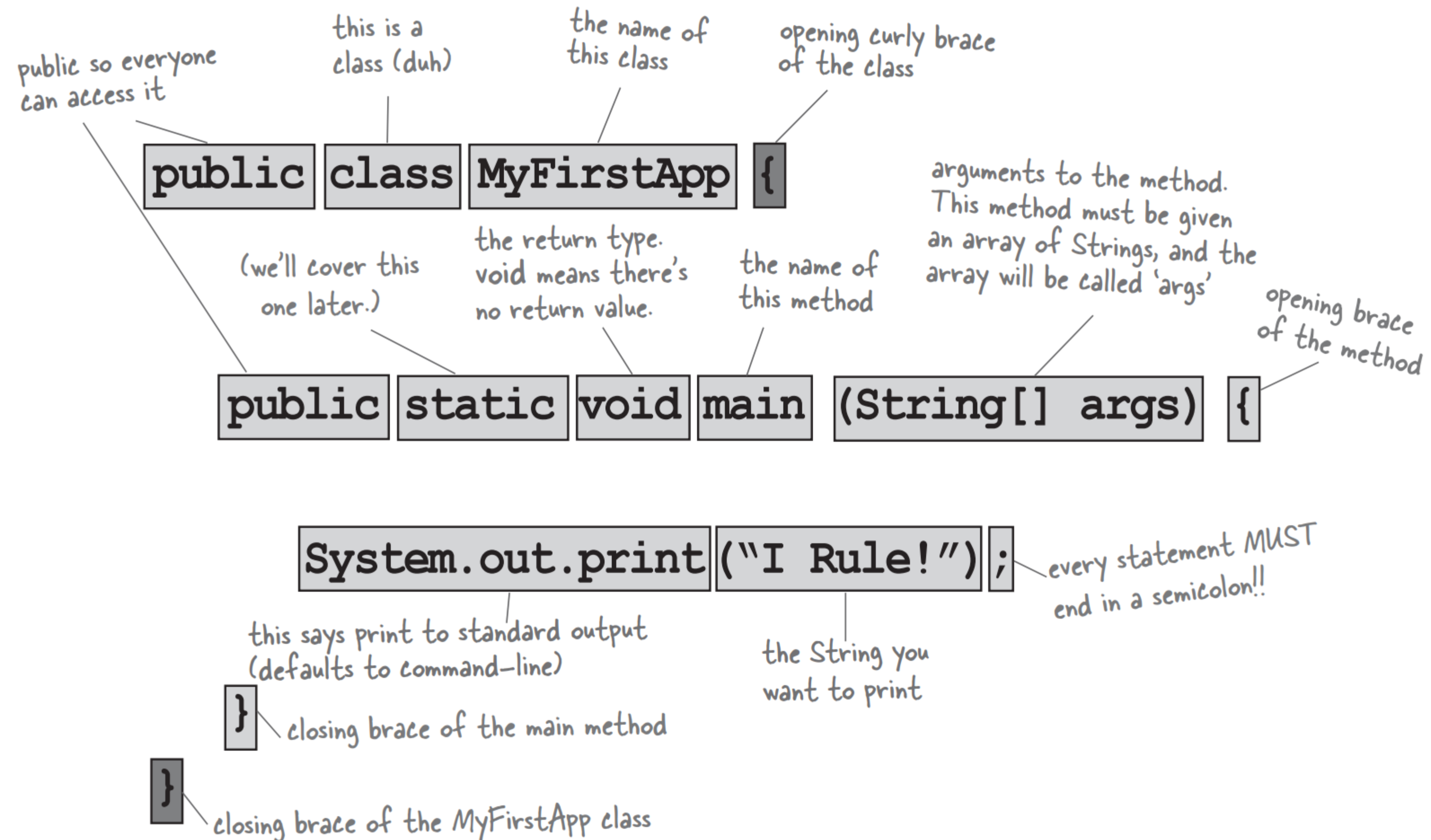
หมายเหตุ ตอนติดตั้ง Apache Netbeans 14 อาจจะไม่จำเป็นต้องเลือกติดตั้ง JAVA EE และ PHP ก็ได้

<No View Available>

Output

INS

MyFirstApp



ขั้นตอนการทำงาน

Editor

- เขียนซอร์สโค้ด

→ **MyFirstApp.java**



```
public class MyFirstApp {  
    public static void main  
    (String[] args) {  
        System.out.println("I Rule!");  
    }  
}
```

MyFirstApp.java

javac.exe
(Compiler)

- คอมไพล์โปรแกรม

javac MyFirstApp.java

→ **MyFirstApp.class**



```
Method Party() 0 aload_0 1  
invokespecial #1 <Method  
java.lang.Object>  
4 return  
Method void  
main(java.lang.String[])  
0 getstatic #2 <Field
```

MyFirstApp.class

java.exe
(Interpreter)

- รันโปรแกรม

java MyFirstApp