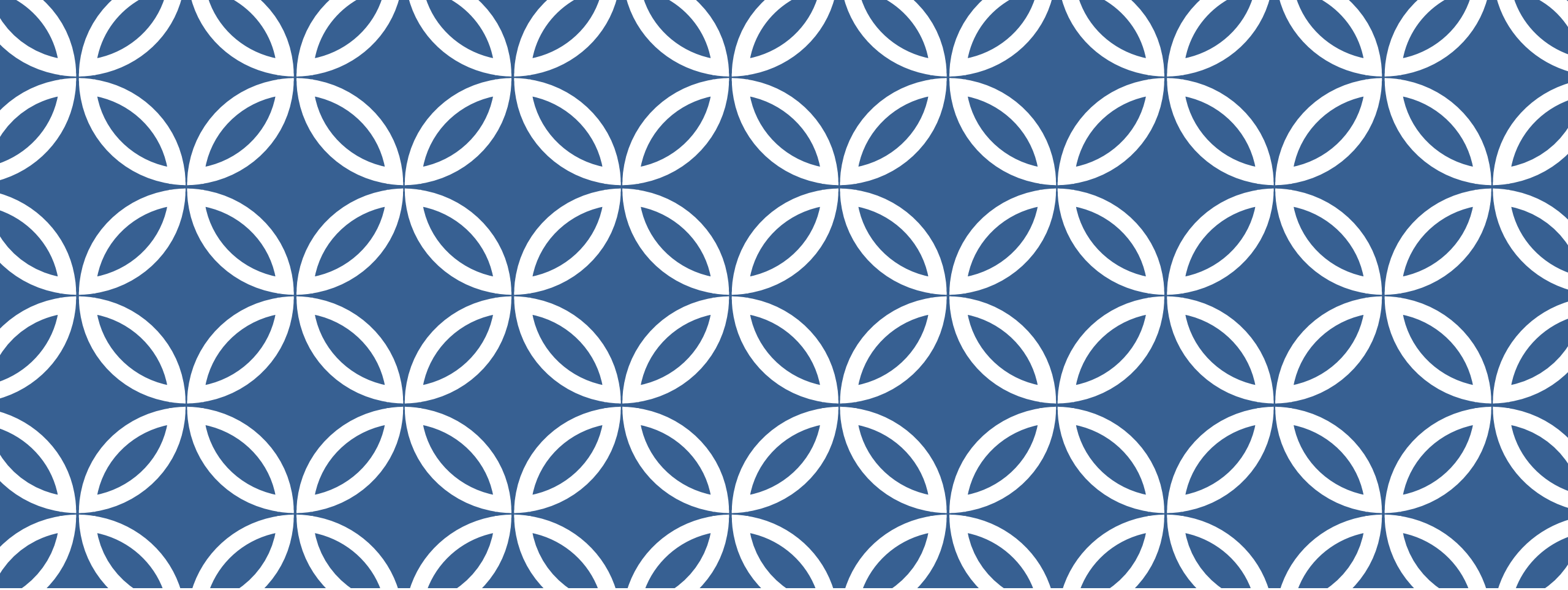


บทที่ 1
INTRODUCTION

อ.สกรณ์ บุษบง
สาขาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
มหาวิทยาลัยราชภัฏบุรีรัมย์

OBJECT-ORIENTED PROGRAMMING

- แนวคิดของการเขียนโปรแกรมเชิงวัตถุเริ่มตั้งแต่ปี 1960
- มีหลายภาษาในการเขียนโปรแกรมที่รองรับการเขียนโปรแกรมเชิงวัตถุนี้

OBJECTS AND THEIR INTERACTIONS IN THE REAL WORLD

สมมุติว่า...

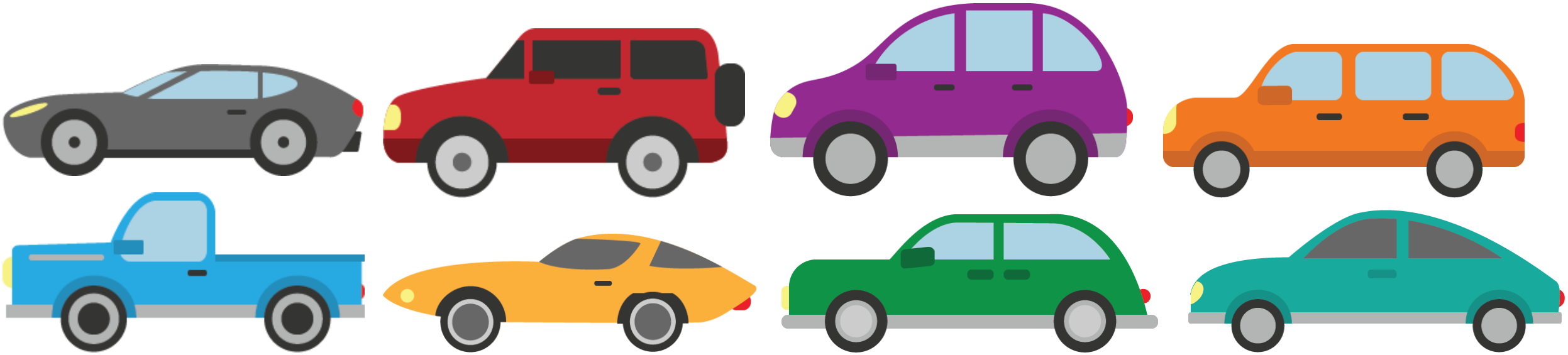
มีคนอยู่ 2 คน

Bill และ Sarah

OBJECTS AND THEIR INTERACTIONS IN THE REAL WORLD

Bill ต้องการซื้อรถยนต์ซักคัน จึงไปเลือกซื้อรถยนต์

รถแต่ละคัน จะมีบางส่วนที่เหมือนกัน และบางส่วนที่ต่างกัน



OBJECTS AND THEIR INTERACTIONS IN THE REAL WORLD

ส่วนที่เหมือนกัน

- ตัวถังรถ
- เครื่องยนต์
- เกียร์
- ล้อ
- ฯลฯ

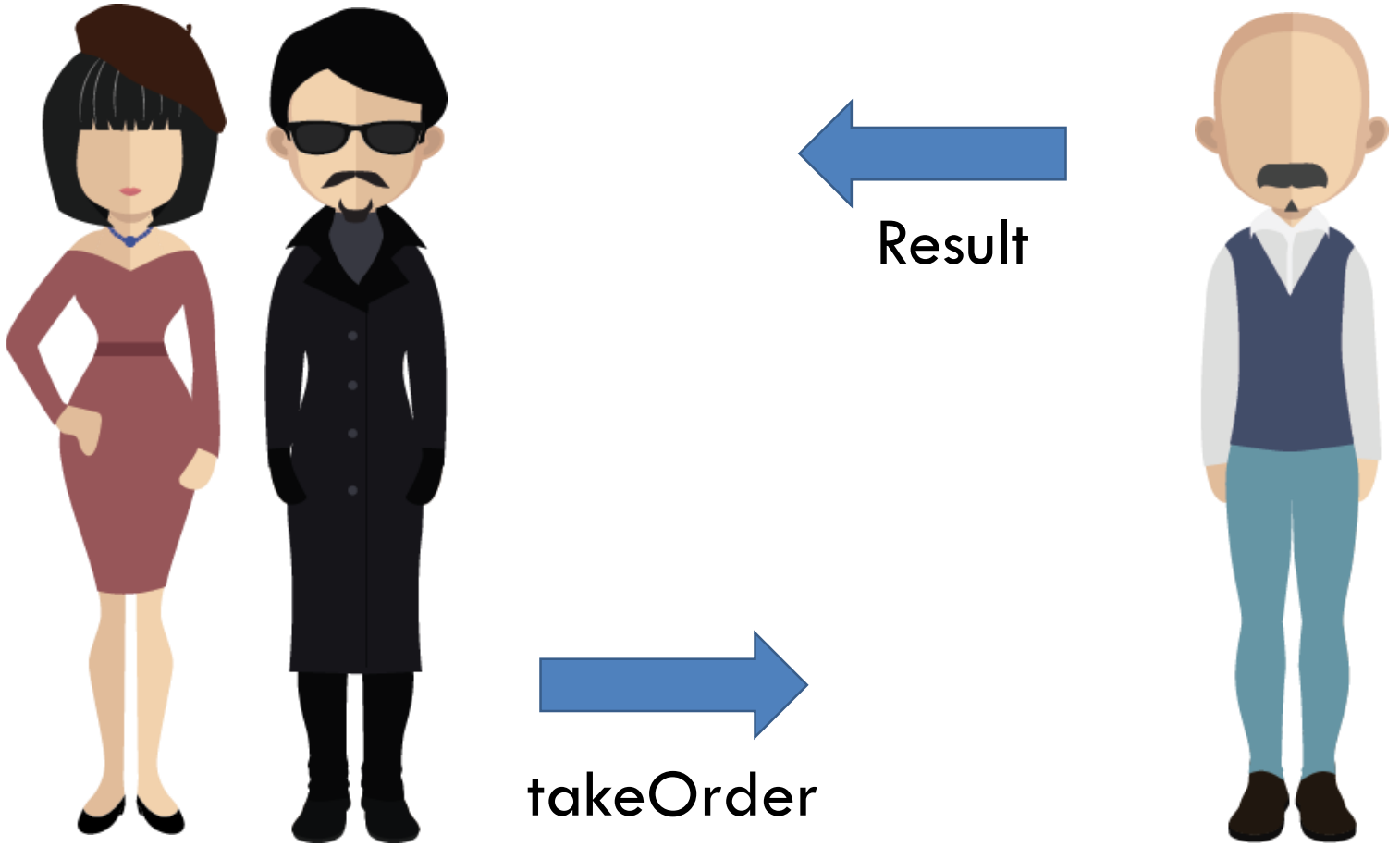
ส่วนที่ต่างกัน

- สี
- ขนาดเครื่องยนต์
- ระบบเกียร์
- ลายล้อแม็ก
- ฯลฯ

OBJECTS AND THEIR INTERACTIONS IN THE REAL WORLD

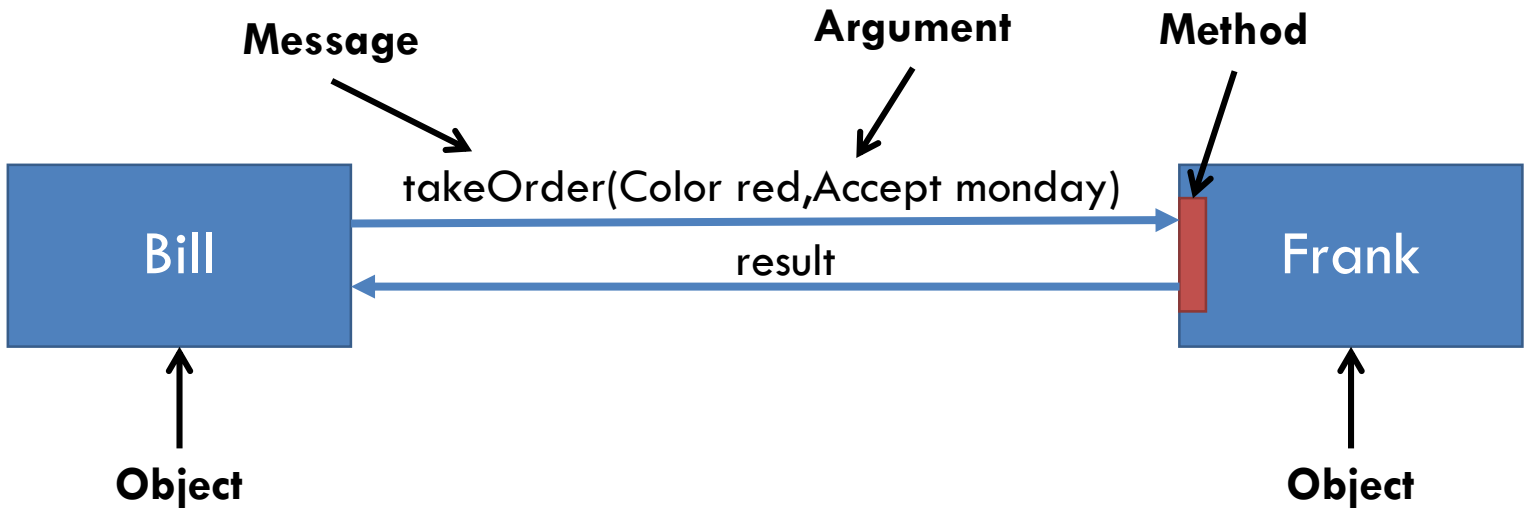
- Bill ต้องการสั่งซื้อรถยนต์สีแดง จึงได้สั่งซื้อกับพนักงานชื่อ Frank และต้องการรถวันจันทร์
- ซึ่ง Frank จะต้อง Confirm การซื้อรถของ Bill
- การสั่งซื้อถือว่าเป็นการส่ง message ระหว่าง Bill และ Frank

OBJECTS AND THEIR INTERACTIONS IN THE REAL WORLD

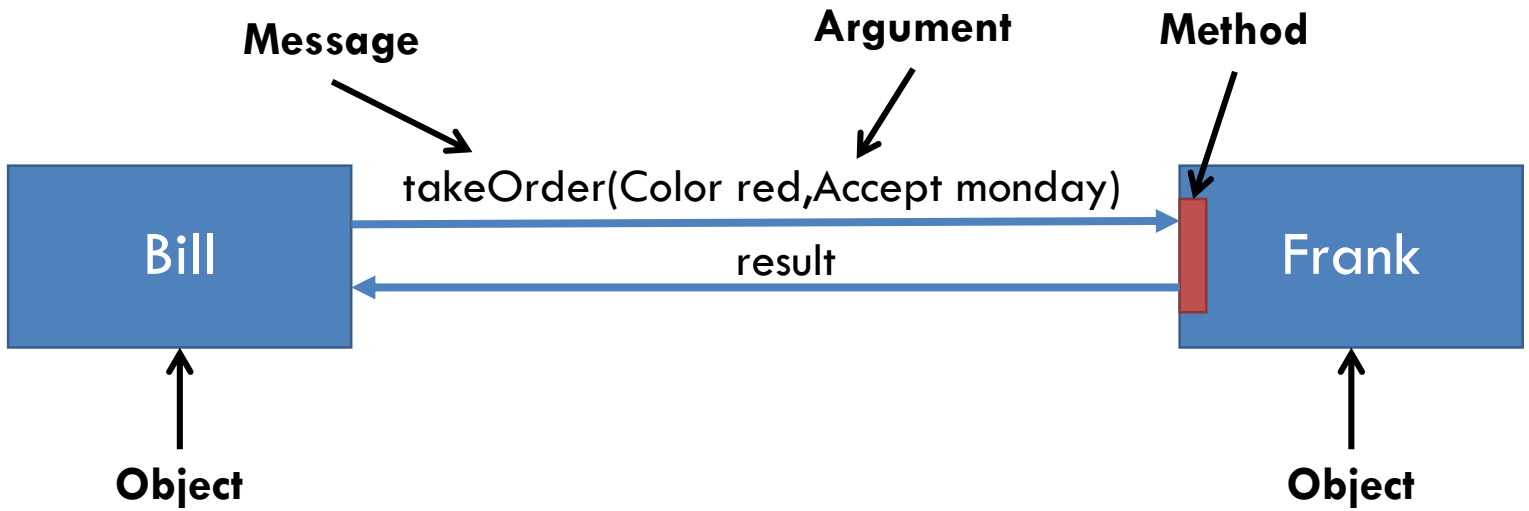


OBJECTS AND THEIR INTERACTIONS IN PROGRAMMING

การโต้ตอบกันระหว่าง Bill และ Frank สามารถแทนได้ด้วยการเขียนโปรแกรมได้ดังนี้



OBJECTS AND THEIR INTERACTIONS IN PROGRAMMING



Bill จะไม่รู้ Frank ทำงานอย่างไร รู้เพียงว่าผลลัพธ์ที่ได้จะเป็นรูปแบบใด ลักษณะนี้เรียกว่า information hiding

JAVA

- ภาษาจาวา ได้รับการเปิดตัวในช่วงปี 1995 เป็นภาษาเชิงวัตถุที่เรียบง่ายและปลอดภัย
- เป็นภาษาที่มีเอกลักษณ์ไม่เหมือนใครในยุคนั้น ทำให้ดึงดูดความสนใจกลุ่ม Computing Community
- 2 ปีหลังจากที่ภาษาจาวาเปิดตัว ได้มีโปรแกรมเมอร์ภาษาจาวาประมาณ 400k คน มีหนังสือจาวากว่า 100 เล่ม

JAVA

- ภาษาจาวาสามารถที่จะทำงานบนคอมพิวเตอร์เครื่องใดก็ได้
- ภาษาจาวาจะถูก Compile เป็น Binary File ที่สามารถรันได้โดย JVM
- JVM สามารถติดตั้งบนคอมพิวเตอร์เครื่องใดก็ได้
- บน internet สามารถรันจาวาได้โดยใช้ Applet ฝังลงใน HTML

OBJECTS AND CLASS

ขอแนะนำให้รู้จัก

David

OBJECTS AND CLASS

David เป็นลูกค้าเช่นเดียวกับ Bill

David

Bill

OBJECTS AND CLASS

ในฐานนะที่ Bill กับ David เป็นลูกค้าเหมือนกันทำให้ทั้งคู่มีคุณสมบัติต่าง ๆ ที่เหมือนกันเช่น

- ชื่อ
- ที่อยู่
- งบประมาณ

สิ่งเหล่านี้เป็นเหมือนสิ่งที่แสดงถึงตัวตนของ Bill และ David

David

Bill

OBJECTS AND CLASS

สิ่งเหล่านี้เป็นเหมือนสิ่งที่แสดงถึงตัวตนของ Bill และ David

เราเรียกสิ่งเหล่านี้ว่า Attribute ของ Object

David

Bill

OBJECTS AND CLASS

Attribute ของ Object ทำให้ Object ที่มาจาก Class เดียวกันแตกต่างกัน เช่น

- David อาจจะมีงบประมาณมากกว่า Bill

OBJECTS AND CLASS

นอกจากเรื่องของ Attribute แล้ว David และ Bill ยังมีพฤติกรรม (behavior) ตามรูปแบบของลูกค้า เช่น

- David และ Bill ตัดสินใจซื้อรถยนต์ สามารถเริ่มต้นการสั่งซื้อโดยใช้ เมธอด purchase()
- เมธอด purchase() จะรวบรวมกระบวนการต่างๆในการทำรายการซื้อเพื่อส่งข้อมูลไปยัง Frank ซึ่งเป็นพนักงานขาย

OBJECTS AND CLASS

โครงสร้างของ David และ Bill มีลักษณะดังนี้

David as an Object
Attributes:
- name = “David”
- address = “1, Robinson Road”
- budget = “2000”

Methods:
- purchase() {send a purchase request to a sales}
- getBudget() {return budget}

Bill as an Object
Attributes:
- name = “Bill”
- address = “18, Sophia Road”
- budget = “1000”

Methods:
- purchase() {send a purchase request to a sales}
- getBudget() {return budget}

David

Bill

OBJECTS AND CLASS

- name, address, budget เป็น Attribute
- purchase() และ getBudget() เป็น Method ของทั้ง 2 Object(David และ Bill)
- จะเห็นว่าทั้ง 2 Object มี Attribute และ Method เหมือนกัน
- ความจริง ลูกค้าทุกคนมี Attribute และ Method เหมือนกัน

OBJECTS AND CLASS

- การกำหนด Object เหล่านี้ จะต้องมาจากต้นแบบเดียวกัน จึงทำให้มีคุณสมบัติเหมือนกัน
- ต้นแบบของ Object เรียกว่า Class
- Class จะเป็น Template ของโครงสร้างของ Object
- Object ที่มาจาก Class เดียวกันจะมีโครงสร้างที่เหมือนกัน

OBJECTS AND CLASS

ดังนั้นโครงสร้างของ Class ลูกค้าจะมีรูปแบบดังนี้

Class Customer

Attributes:

name
address
budget

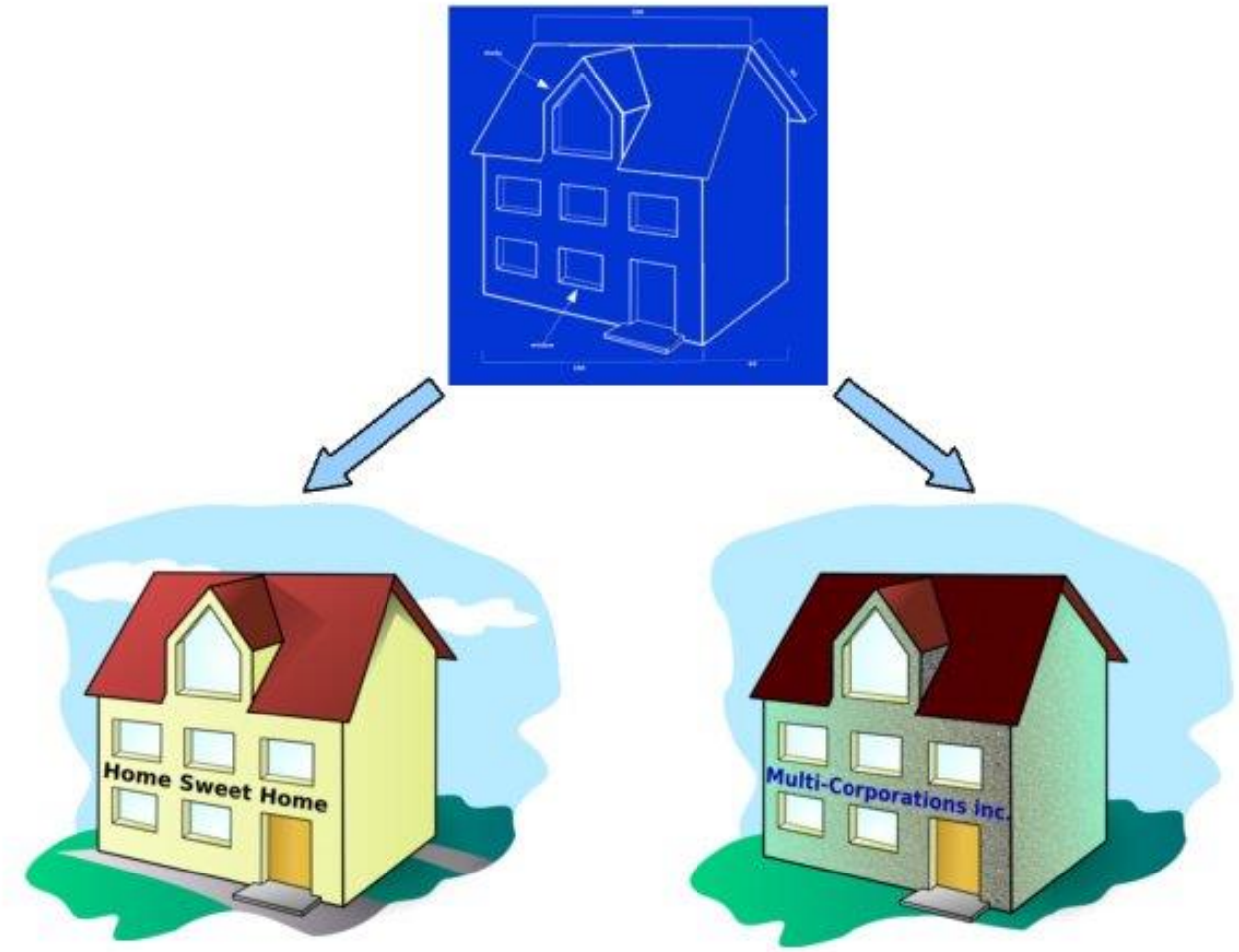
Methods:

purchase() {send a purchase request to a sales}
getBudget() {return budget}

OBJECTS AND CLASS

- ความแตกต่างหลักๆระหว่าง Class และ Object คือการกระทำต่อ Attribute และ Method
- Class เปรียบเสมือนต้นแบบของ Object ดังนั้น Attribute และ Method ของ Class จะไม่มี value
- Object ที่ถูกสร้างจากคลาสเปรียบเสมือนตัวแทนของ Class
- Object ที่สร้างจาก Class เดียวกัน อาจมี value ของ Attribute ที่แตกต่างกันได้

OBJECTS AND CLASS



OBJECTS AND CLASS

มาดูพนักงานขายบ้าง

- พนักงานขายก็มี Attribute และ Method เช่นกัน
- Jules และ Mary ก็เป็นพนักงานขาย
- พวกเขาจึงมีความสามารถของพนักงานขายทุกประการ
- พนักงานขายจะต้องทำการขาย คือ takeOrder()

OBJECTS AND CLASS

โครงสร้างของ Jules และ Mary

Mary as an Object

Attributes:

name = “Mary”

Methods:

```
takeOrder()    {
  check with warehouse on stock availability
check with warehouse on delivery schedule
  if ok
  then    {instruct warehouse to deliver stock(address, date)
      return ok}
  else return not ok
 }
```

Jules as an Object

Attributes:

name = “Jules”

Methods:

```
takeOrder()    {
  check with warehouse on stock availability
check with warehouse on delivery schedule
  if ok
  then    {instruct warehouse to deliver stock(address, date)
      return ok}
  else return not ok
 }
```

OBJECTS AND CLASS

โครงสร้างของ Jules และ Mary มี Attribute และ Method ที่เหมือนกัน ดังนั้น โครงสร้างของ Class พนักงานขายเป็นดังนี้

```
Class SalesPerson
    Attributes:
        name =
    Methods:
        takeOrder()   {
          check with warehouse on stock availability
          check with warehouse on delivery schedule
          if ok
          then    {instruct warehouse to deliver stock(address, date)
              return ok}
          else return not ok
        }
```

OBJECTS AND CLASS

- Class SalesPerson และ Class Customer คนละ Class กันนะ
- เนื่องจาก
"customers make orders and salespersons take orders."

MESSAGE AND METHOD

- Object มีการสื่อสารกันโดยการส่ง message
- message คือ method call จาก message-sending object to a message-receiving object
- message-sending object เรียกว่า sender
- message-receiving object เรียกว่า receiver

MESSAGE AND METHOD

- Object จะมีการตอบสนองต่อ message โดยการรัน Method
- ข้อมูลเพิ่มเติมที่ส่งมาพร้อมกับ message เรียกว่า arguments
- การกำหนด Parameter ช่วยให้เพิ่มความยืดหยุ่นของ message

MESSAGE COMPONENTS

Message ประกอบไปด้วยส่วนแระกอบทั้งหมด 3 ส่วนดังนี้

- การระบุ Object ที่เป็น receiver
- ชื่อ Method (ของ receiver)
- arguments (ข้อมูลเพิ่มเติมที่ส่งให้ Method ทำงาน)

MESSAGE COMPONENTS

- ก่อนหน้านี้ Bill ได้ส่ง Message ไปหา Frank เพื่อสั่งซื้อรถยนต์
- Object Bill มี Method ชื่อ purchase() ที่จะใช้ในการสั่งซื้อ

MESSAGE COMPONENTS

Bill as an Object

Attributes:

name = “Bill”

address = “18, Sophia Road”

budget = “1000”

Methods:

purchase() {

Frank.takeOrder(“Bill”, “sedan”, “18, Sophia Road”, “Monday”) }

getBudget() {return budget}

MESSAGE COMPONENTS

Message
Frank.takeOrder(“Bill”, “sedan”, “18, Sophia Road”, “Monday”)
หมายความว่า

- Frank เป็น receiver
- takeOrder เป็น method ของ Frank
- “Bill”, “sedan”, “18, Sophia Road”, “Monday” เป็น arguments ของ message

METHOD

- message จะถูกต้องเมื่อ receiver มี method ตรงกับ message และมี argument ที่เหมาะสม
- message ที่ถูกต้องท่านั้นที่จะถูกดำเนินการโดย receiver

METHOD

- takeOrder() เป็น message ที่ถูกต้อง เนื่องจาก Frank มี method ที่ตรงกันและ argument ที่ต้องการสอดคล้องกับ message (who, model, address, date)
- ดังนั้น Method takeOrder() ของ Frank จะมีโครงสร้างดังนี้

METHOD

Frank as an Object

Attributes:

name = “Frank”

Methods:

```
takeOrder(who, model, address, date)   {
  check with warehouse on stock availability
  check with warehouse on delivery schedule
  if ok then {
    instruct warehouse to deliver stock to address on date
    return ok
  } else return not ok
}
```

METHOD

- method ของ Frank จะมีเพียง Frank เท่านั้นที่รู้ขั้นตอนการทำงาน
- Bill หรือลูกค้าคนอื่นๆ จะไม่รู้เลยว่า Frank ทำงานอย่างไร
- Bill หรือลูกค้าคนอื่นๆ จะรู้เพียงว่า Frank จะตอบสนองต่อ message ของตนเองได้

METHOD

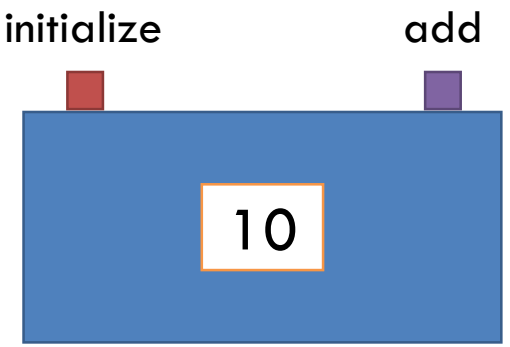
- ในแนวคิดของ Object-Oriented programming Bill และ Frank ได้แสดงถึงหลักการสำคัญของ OO คือ
- Information Hiding
- Bill จะรู้เพียงว่าต้องส่ง message และ argument อะไรไปให้ Frank บ้าง เพื่อให้ Frank ทำงานตามที่ request ได้
- Frank มีอิสระในการเลือกวิธีในการทำงานเพื่อตอบสนอง request

CREATING OBJECTS

- ในการเขียนโปรแกรมเชิงวัตถุนั้น object จะถูกสร้างจาก class
- Object ของ customer ทั้งหมดจากถูกสร้างจาก class customer
- Object ของ salePerson ทั้งหมดจากถูกสร้างจาก class salePerson

CREATING OBJECTS

- Object ที่ถูกสร้างขึ้นมาจาก class นั้นจะมี state ของตัวเอง
- เช่น counter เป็นอุปกรณ์ที่ใช้นับ มันมี 2 ปุ่ม
  - initialize ใช้ในการ reset ค่ากลับมาเป็น 0
  - add ใช้ในการบวกค่า 1 เข้าไปเก็บไว้



CREATING OBJECTS

โครงสร้างของ counter

First Counter Object

Attributes:

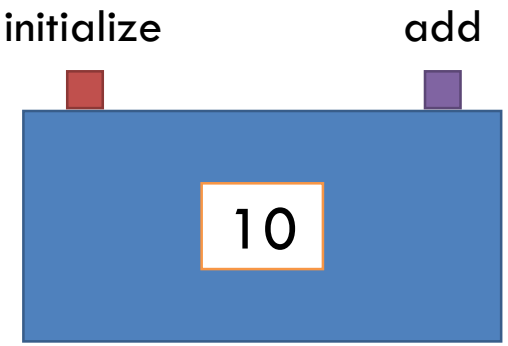
number = 10

Methods:

add() {number = number + 1}

initialize() {number = 0}

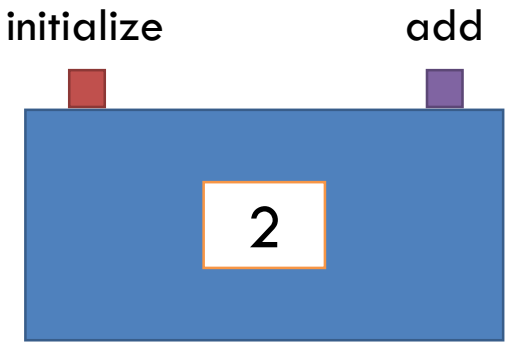
getNumber() {return number}

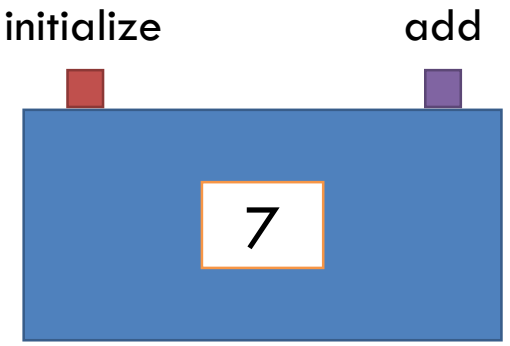


CREATING OBJECTS

เช่นเดียวกับ counter ตัวที่แล้ว counter 2 ตัวนี้มี 2 ปุ่ม เหมือนกัน ทำงานเหมือนกัน

CREATING OBJECTS





Second Counter Object

Attributes:

number = 2

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

Second Counter Object

Attributes:

number = 7

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

CREATING OBJECTS

- จะเห็นว่า ทั้ง 3 counter นั้นมีโครงสร้างต่างๆ เหมือนกัน แต่ต่างกันที่ ค่าที่เก็บ เราเรียกว่า object ทั้ง 3 มี state ที่ต่างกัน
- ดังนั้น โครงสร้างของ class ที่เป็นต้นแบบของ object ทั้ง 3 จะต้องมีโครงสร้างดังนี้

CREATING OBJECTS

Class Counter

Attributes:

number

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

- Attribute : number
- initialize() เป็น method ทำหน้าที่ reset ค่าของ number ให้เป็น 0
- Add() เป็น method ทำหน้าที่ บวก 1 ไปที่ number
- getNumber() เป็น mether ที่จะ return ค่าปัจจุบันของ number

CREATING OBJECTS

Object ที่เพิ่งถูกสร้างขึ้นใหม่จะมี attribute และ method เหมือนกัน

Fourth Counter Object

Attributes:

number = 0

Methods:

add() {number = number + 1}

initialize() {number = 0}

getNumber() {return number}

SUMMARY

- Object สร้างมาจาก Class
- Object ที่สร้างมาจาก Class เดียวกันจะมี attribute และ method เหมือนกัน
- บางครั้ง Object ที่สร้างมาจาก Class attribute อาจมีค่าต่างกันได้

SUMMARY

- Object ที่สร้างมาจาก Class ที่ต่างกันไม่สามารถแบ่งปัน attribute หรือ method ได้
- Message ประกอบไปด้วย 3 ส่วน object identifier, method name, arguments

EXERCISES

1. ให้นักศึกษาจำลองสถานการณ์การไปซื้อ อาหารกลางวันที่โรงอาหารจากนั้นออกแบบ object และ message ที่ object แลกเปลี่ยนกัน

EXERCISES

2. ให้นักศึกษาแต่ละกลุ่มวิเคราะห์สถานการสมมุติต่อไปนี้แล้วจำลองการโต้ตอบของ Object

- คนขับรถที่กำลังขับรถ
- ลูกค้าที่กำลังถอนเงินจากเครื่อง ATM
- ลูกค้าที่กำลังซื้อเครื่องเล่น DVD จากร้านค้า
- อาจารย์ที่กำลังสอนนักศึกษาในชั้นเรียน
- ตำรวจจราจรที่กำลังจับหมวกกันน็อคหน้ามหาวิทยาลัย
- นักศึกษากำลังขึ้นสอบโปรเจค 1
- เด็กนักเรียนกำลังขออนุญาตผู้ปกครองไปร้านเกมส์

EXERCISES

3. ให้นักศึกษาจำลองโครงสร้าง class รถยนต์ที่สามารถ start, move forward, move backward, stop และ off

รถยนต์สามารถ return ค่าตำแหน่งปัจจุบันได้

ตำแหน่งเริ่มต้นคือ 0

```
class Car {
  Attributes:
      ...
   Methods:
      ...
}
```

สรุป