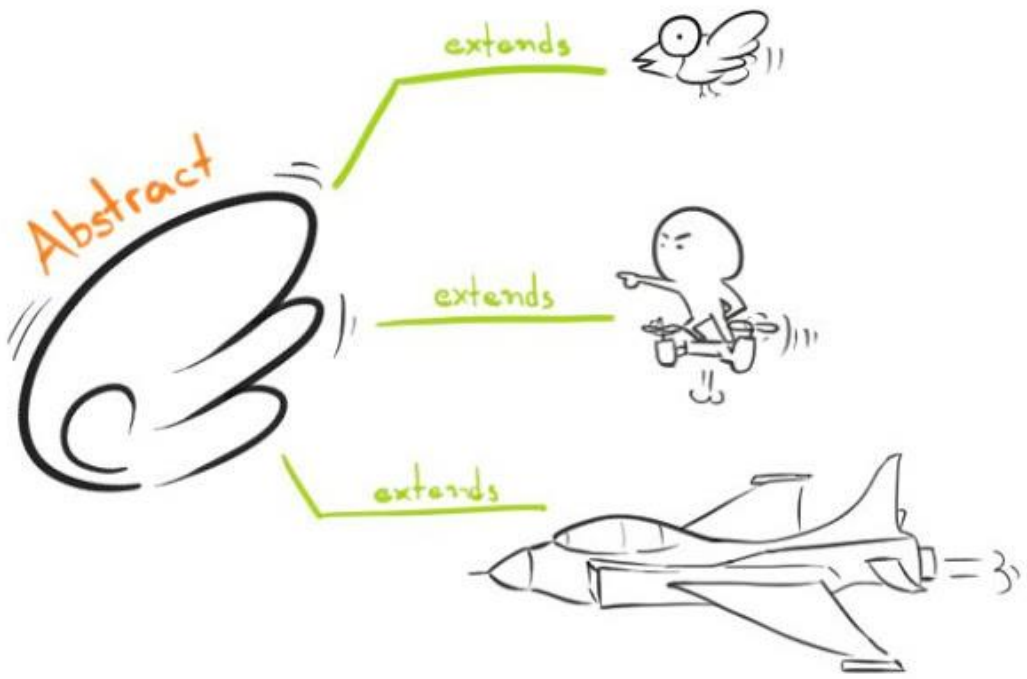
แนวคิดเชิงวัตถุ

ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน (tko@webmail.npru.ac.th)

สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี

เนื้อหาการเรียนรู้

- แนวคิดเชิงวัตถุ
- ตัวแทนวัตถุ
- ความสัมพันธ์เชิงวัตถุ
- มุมมองแบบนามธรรม
- ขอบเขตของปัญหาและความต้องการ
- การวิเคราะห์และออกแบบเชิงวัตถุ
- แนวคิดการเขียนโปรแกรมเชิงวัตถุ



https://www.tamemo.com/post/125/all-about-oop-3-polymorphism/

แนวคิดเชิงวัตถุ

- เทคนิคการจำลองสิ่งต่าง ๆ ด้วย “วัตถุ”
- แจกแจงรายละเอียดและอธิบายความสัมพันธ์ของเหตุการณ์
- แบบจำลองหรือต้นแบบ
- มุมมองแบบนามธรรมในการจำกัดรายละเอียดของวัตถุ
- นิยามคลาส เพื่อใช้เป็นตัวแทนหรือตัวแบบของวัตถุ
- โครงสร้างหรือแบบจำลองในการพัฒนาซอฟต์แวร์

(abstract)

(model)

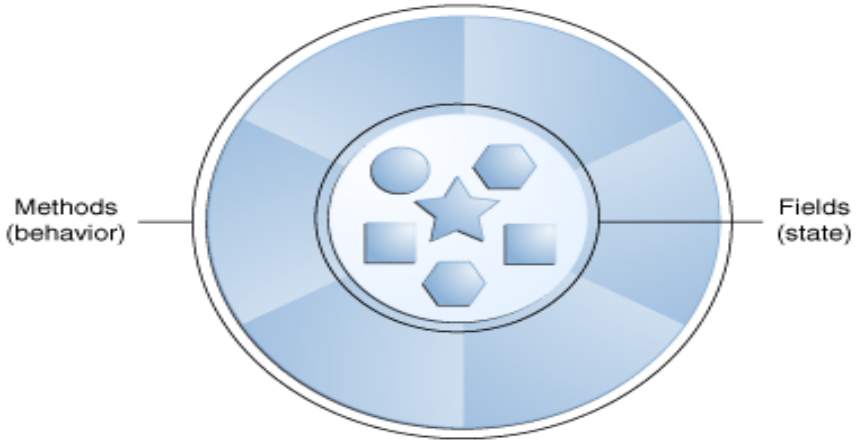
(class)

(object)

(software)

ตัวแทนวัตถุ

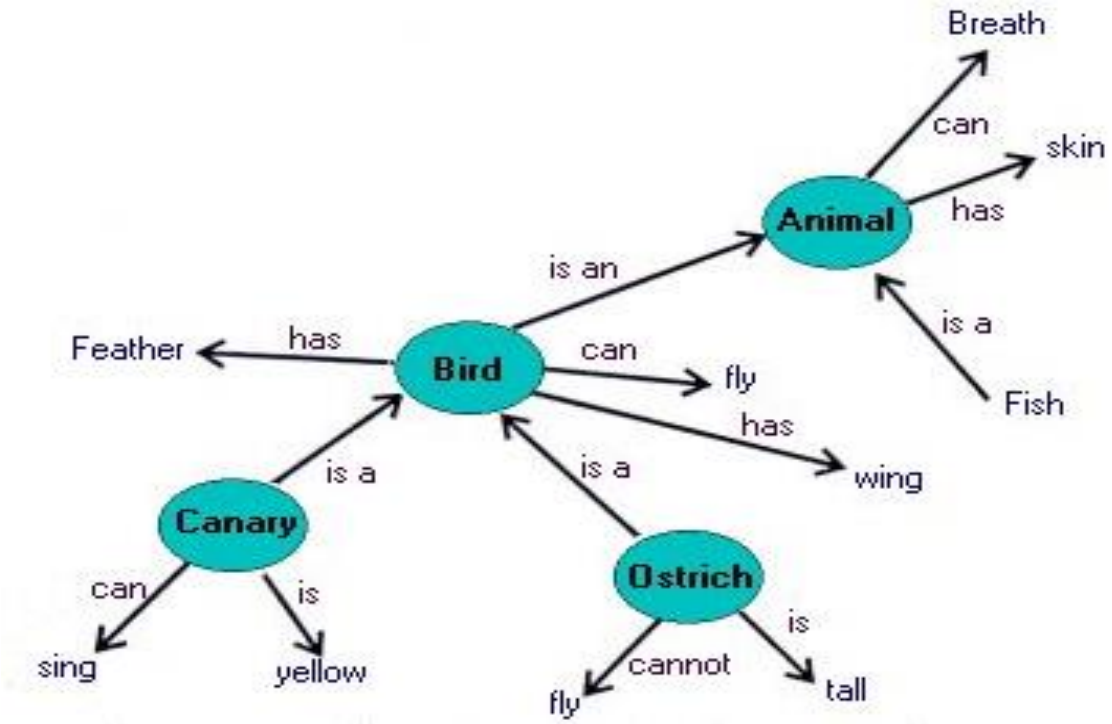
- การพัฒนาซอฟต์แวร์และการสร้างออบเจ็กต์ที่ใช้เป็นตัวแทนของสิ่งต่าง ๆ
- ชนิดข้อมูลแบบนามธรรม
  - สถานะหรือคุณสมบัติ (property)
  - พฤติกรรม (method)



https://www.tamemo.com/post/125/all-about-oop-3-polymorphism/

ความสัมพันธ์เชิงวัตถุ

- การสร้างแบบจำลองแนวคิดของสิ่งต่าง ๆ ตามเหตุการณ์ที่เกิดขึ้น



ที่มา: (ยืน ภู่วรวรรณ, 2542)

มุมมองแบบนามธรรม

- วัตถุแต่ละตัว มีพฤติกรรมและข้อมูลเฉพาะที่สัมพันธ์กัน
  1) แนวคิดการจัดกลุ่ม (classification abstraction)
  2) แนวคิดเชิงสัมพันธ์ (association abstraction)
  3) แนวคิดการแยกส่วน (aggregation abstraction)
  4) แนวคิดพื้นฐาน (generalization abstraction)

ที่มา : (Grady Booch, 1980)

ขอบเขตของปัญหาและความต้องการ

- ขอบเขตของปัญหาที่สนใจ ครอบคลุมปัญหาทั้งหมด
- รายละเอียดของปัญหาได้ชัดเจน
- เข้าใจปัญหาและวิธีการแก้ไขปัญหา
- ข้อกำหนดความต้องการ ข้อตกลงเบื้องต้นในการพัฒนาโปรแกรม
- การวิเคราะห์และออกแบบเชิงวัตถุ
- ภาพรวมการทำงานของโปรแกรมของทั้งระบบ

https://pixabay.com/th/illustrations กรอบ-กรอบรูป-เดคคอร์-photoshop-2015283/

การวิเคราะห์ความต้องการเชิงวัตถุ

1) การวิเคราะห์หาขอบเขตของปัญหา
   - การกำหนดขอบเขตในการพัฒนาซอฟต์แวร์
   - เข้าใจองค์ประกอบของปัญหา
- 2) การกำหนดขั้นตอนและวิธีการแก้ปัญหา
   - กำหนดรายละเอียดในการเขียนโปรแกรม

(problem space)

(solution space)

https://www.jobbkk.com/variety/detail/4129

กระบวนทัศน์เชิงวัตถุ

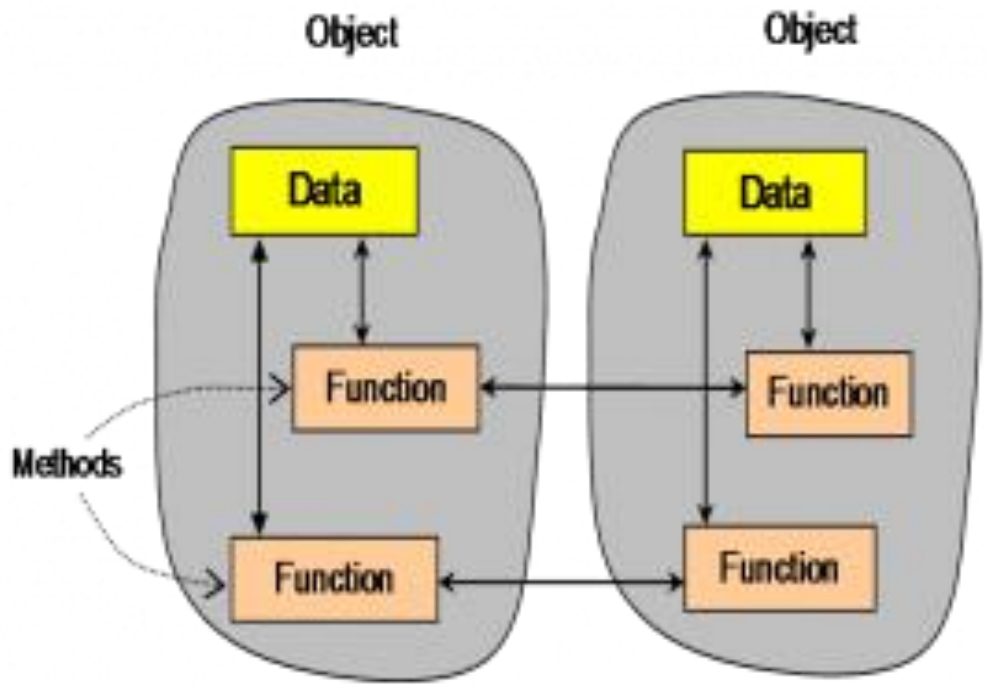
- **กระบวนทัศน์การเขียนโปรแกรม**
- แนวคิดในการพัฒนาโปรแกรมที่ได้รับความนิยม
- การแปลงแบบจำลองเชิงวัตถุไปเป็นโปรแกรมคอมพิวเตอร์
- **การเขียนโปรแกรมเชิงวัตถุ**
  - การนำต้นแบบมาใช้สร้างออบเจ็กต์
  - การนำออบเจ็กต์มาเป็นส่วนประกอบของโปรแกรม

https://www.wynnsoftstudio.com/Mobile_Application_Design_Tips

https://sites.google.com/site/4146anantaya/xngkh-prakxb-khxng-kar-xxkbaeb

โครงสร้างข้อมูลกับวิธีการเชิงวัตถุ

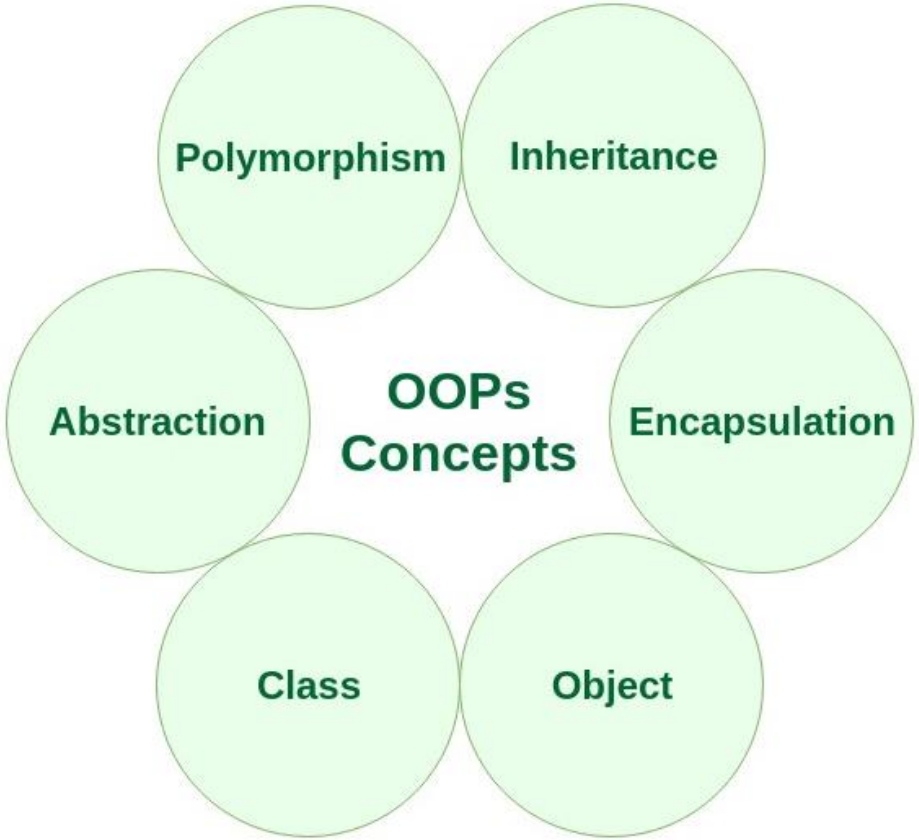
- การออกแบบวัตถุตาม**หน้าที่และความรับผิดชอบ**
- การสื่อสารข้อมูลทั้ง**ภายในและภายนอกวัตถุ**
- การเขียนโปรแกรมด้วยภาษาระดับสูง
- จัดการกับ**ความซับซ้อน**ตาม**หน้าที่ของวัตถุ**
- การนำโค้ดกลับมาใช้ผ่าน**คลาส**
- การปรับปรุงต้นแบบของ**ออบเจ็กต์**



ที่มา : (Ravinder, 2015)

แนวคิดการเขียนโปรแกรมเชิงวัตถุ

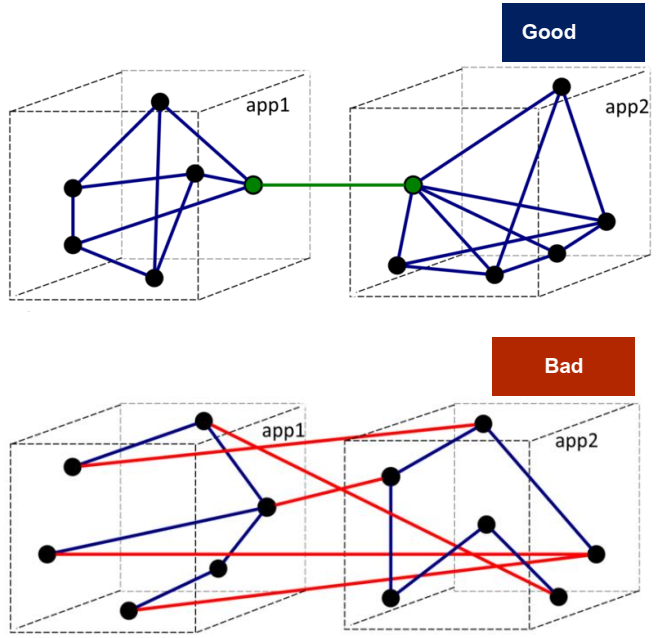
- ภาวะหลายรูปแบบ (Polymorphism)
- การถ่ายทอด (Inheritance)
- การห่อหุ้ม (Encapsulation)
- นามธรรม (Abstraction)
- คลาส **(Class)**
- ออบเจ็กต์หรือวัตถุ **(Object)**
  - เมธอดหรือการกระทำ (Method)
  - ข้อความที่ใช้สื่อสาร (Message Passing)



https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/

ประโยชน์ของแนวคิดเชิงวัตถุ

- การออกแบบจำลองจากมุมมองนามธรรม
  - เข้าใจและสามารถพัฒนาได้ตามที่ออกแบบ
  - ปกปิดกระบวนการทำงานไว้ภายในและเปิดให้ใช้งานจากภายนอก
  - **ความยึดเหนี่ยวภายในสูง (higher cohesion)**
  - **ความเกี่ยวข้องกับภายนอกต่ำ (lower coupling)**
- การเขียนโปรแกรมเชิงวัตถุ
  - ความเป็นธรรมชาติ (naturalness)
  - การใช้ซ้ำ (reusability)
  - ความง่ายในการบำรุงรักษา (maintainability)



https://en.wikipedia.org/wiki/Cohesion_(computer_science)#/media/File:Good,_bad_apps.png

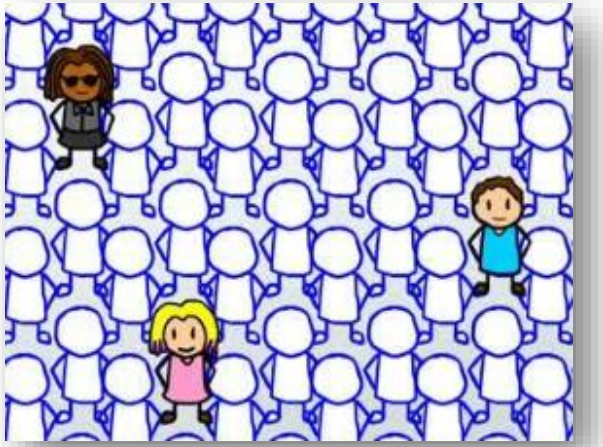
การวิเคราะห์และออกแบบเชิงวัตถุ

A กำหนดสิ่งที่ทำให้ชัดเจนจากการวิเคราะห์ความต้องการ

D ออกแบบให้ตอบสนองความต้องการมากกว่าการนำไปใช้

สรุปท้ายบท

- แนวคิดเชิงวัตถุ เทคนิคการจำลองสิ่งต่าง ๆ ด้วย “วัตถุ”
- ในการพัฒนาซอฟต์แวร์จากออบเจ็กต์ด้วยต้นแบบที่นิยามไว้
  - แบบจำลองข้อมูล
  - แบบจำลองการทำงาน
  - แบบจำลองความสัมพันธ์
- มุมมองแบบนามธรรม->ให้เกิดเป็นรูปธรรม
- แนวคิดการเขียนโปรแกรมเชิงวัตถุ
- ช่วยให้สามารถปรับเปลี่ยนและพัฒนาซอฟต์แวร์ได้ง่าย



https://study.com/academy/lesson/oop-object-oriented-programming-objects-classes-interfaces.html