

# การเขียนโปรแกรมคอมพิวเตอร์ แบบภาษาเชิงวัตถุ

การเขียนโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ เป็นการเขียนโปรแกรมเพื่อให้สามารถใช้อ็อบเจกต์ประกอบหรือการทำงานร่วมกัน แบ่งออกเป็นส่วนๆ สามารถใช้คุณสมบัติที่มีการพัฒนามาแล้ว เริ่มต้นผู้สนใจควรศึกษาถึงโครงสร้าง นิยาม ความหมายของภาษาเชิงวัตถุ



# ประกอบด้วย

---

- พื้นฐานโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
- ประโยชน์ของโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ
- การประยุกต์ใช้งานโปรแกรมเกมแบบภาษาเชิงวัตถุ



# พื้นฐานโปรแกรมคอมพิวเตอร์แบบ ภาษาเชิงวัตถุ

---

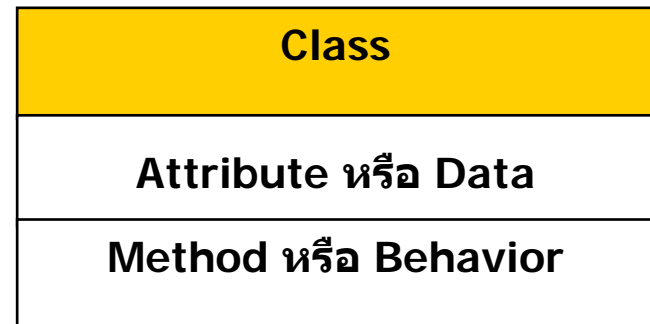
โปรแกรมแบบภาษาเชิงวัตถุ เป็นการนำ“วัตถุ” (object) มาประกอบกันเป็นโปรแกรมคอมพิวเตอร์ กำหนดให้องค์ประกอบในโปรแกรมคอมพิวเตอร์เป็นวัตถุ มีคุณสมบัติต่างๆ เพื่อใช้ในการประมวลผลและสามารถทำงานร่วมกับวัตถุอื่นๆ ได้ มีการสื่อสารกันได้



# คลาส(class)

---

คลาส(class) เปรียบเสมือนแม่พิมพ์บอกลักษณะว่าวัตถุในคลาสนี้มีลักษณะและพฤติกรรมเป็นอย่างไร ภายในคลาสประกอบด้วย แอททริบิวท์ (Attribute) หรือข้อมูล (Data) เป็นสิ่งที่ใช้บอกถึงลักษณะต่างๆของวัตถุในคลาสเมธอด(Method) หรือพฤติกรรม (Behavior) เป็นสิ่งที่ใช้อธิบายการทำงานของวัตถุในคลาส



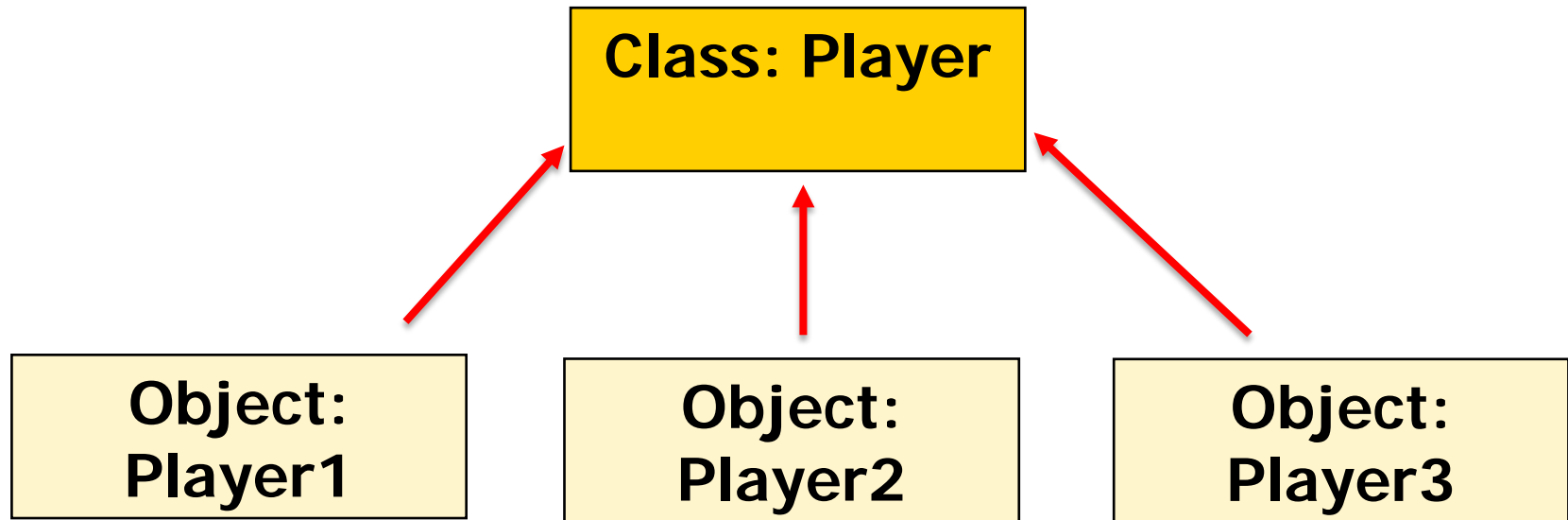


# คลาส(class)

---

คลาสเปรียบเสมือนโปรแกรมที่เขียนขึ้นเพื่อทำหน้าที่เป็นพิมพ์เขียวของวัตถุ(object) การสร้างวัตถุจากคลาสเรียกว่าการทำ **instantiation** ผลลัพธ์ที่ได้คือ **instance** ของวัตถุ เรียก **object** หนึ่ง **object** ว่าหนึ่ง **instance** สามารถสร้าง **object** ได้หลายๆ **instance** จากคลาสเพียงคลาสเดียว ส่วนในภาษา **C#** หากสร้างคลาสหนึ่งคลาส ยกตัวอย่างเช่น สร้างคลาสขึ้นมาเพื่อ नियาม **node** หลังจากนั้น สามารถเขียนโปรแกรมคอมพิวเตอร์สร้าง **binary tree** ขณะทำงานอาจจะสร้าง **object** จากคลาส **node** ได้หลายล้าน **instance** ภายในหนึ่งวินาที เป็นต้น

# คลาส(class)





# คลาส(class)

การใช้งานคลาสในโปรแกรมคอมพิวเตอร์แบบ  
ภาษาเชิงวัตถุ สามารถทำได้ดังนี้คือ

1) ทำการสร้าง **object** และใช้งานคลาสผ่าน  
**object**

2) การเรียกใช้งานโดยตรงโดยไม่ต้องสร้าง  
**object** เรียกวินี้ว่า **static class**

โครงสร้างรูปแบบของการสร้างคลาส

```
Class ชื่อคลาส
{
    method1(){
        //วิธีการทำงาน
    }
    method2(){
        //วิธีการทำงาน
    }
    attribute1;
    attribute2;
}
```



# คลาส(class)

---

วิธีการเข้าถึงหรือใช้ตัวแปร (**Variable**) หรือ ฟังก์ชัน (**Function**) สามารถประกาศการเข้าถึง ได้ดังนี้

- **public** คือ เมื่อประกาศตัวแปรหรือฟังก์ชันจะสามารถถูกอ้างถึงได้ทั้งหมด ทั้งภายในหรือภายนอกคลาส
- **protected** คือ จะถูกอ้างถึงได้เฉพาะในคลาสเดียวกันหรือคลาสลูก(คลาสที่มีการขยาย **extend** หรือสืบทอด **inherit** มาจากคลาสอื่น)
- **private** คือ เมื่อประกาศตัวแปรหรือฟังก์ชันจะสามารถใช้อ้างได้ภายในเฉพาะคลาสตัวเองเท่านั้น
- **static** จะถูกอ้างถึงได้โดยมีเงื่อนไขดังนี้ เมธอดที่เป็น **static** เช่น คลาส **in** สามารถเรียกได้เฉพาะเมธอดในคลาสเดียวกันที่เป็น **static** เท่านั้น เมธอดที่เป็น **static** จะถูกเรียกจากที่อื่นโดยอ้างผ่านทางชื่อคลาสได้ทันที ซึ่งต่างกับเมธอดตรงที่ไม่เป็น **static** ที่ต้องมีการสร้าง **instance** ก่อน





# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

```
using UnityEngine;
using System.Collections;

public class ScopeAndAccessModifiers : MonoBehaviour
{
    public int alpha = 5;
    private int beta = 0;
    private int gam = 5;
    private AnotherClass myOtherClass;

    void Start ()
    {
        alpha = 29;
        myOtherClass = new AnotherClass();
        myOtherClass.Fruitchine(alpha, myOtherClass.apples);
    }
}
```

# ตัวอย่างโปรแกรมเกมคอมพิวเตอร์ ในการกำหนดการเข้าถึง

```
void Example (int pens, int crayons)
{
    int answer;
    answer = pens * crayons * alpha;
    Debug.Log(answer);
}

void Update ()
{
    Debug.Log("Alpha is set to: " + alpha);
}
}
```



# ประโยชน์ของโปรแกรม คอมไพเตอร์แบบภาษาเชิงวัตถุ

โปรแกรมคอมไพเตอร์แบบภาษาเชิงวัตถุ มีประโยชน์ในการนำวัตถุ(object) ที่อยู่ในโปรแกรมคอมไพเตอร์ที่พัฒนาแบบภาษาเชิงวัตถุ สามารถนำมาใช้ร่วมกันได้ สามารถทำการสืบทอดคลาส เป็นการสร้างสิ่งใหม่ขึ้นด้วยการสืบทอดหรือรับเอาคุณสมบัติบางอย่างมาจากสิ่งเดิมที่มีอยู่แล้วและสร้างเพิ่มเติมอีกได้ ทำให้ผู้พัฒนาโปรแกรมคอมไพเตอร์ไม่ต้องเริ่มใหม่ทั้งหมด

การสร้างคลาสใหม่โดยการสืบทอดมาจากคลาสเก่า มีวิธีการดังนี้

- 1) สร้างคลาสใหม่ซึ่งสืบทอดมาจากคลาสเก่า เรียก คลาสใหม่ ว่าเป็น **ซับคลาส ของ คลาสเก่า** และ เรียก คลาสเก่า ว่าเป็น **ซุเปอร์คลาส ของคลาสใหม่**
- 2) วัตถุของคลาสใหม่ มาจากคลาสเก่า **แต่วัตถุทั้งหมดของคลาสเก่าไม่จำเป็นต้องเป็นของคลาสใหม่**

# ตัวอย่างโปรแกรมคอมไพเตอร์แสดง วิธีการสืบทอดคลาสที่มีชื่อว่า nager มา จากคลาสที่มีชื่อว่า Employee

```
public class nager : Employee
{
    // Employee fields, properties, methods and events are inherited

    // New nager fields, properties, methods and events go here...
}
```



# ประโยชน์อื่นๆ ของการเขียนโปรแกรม คอมพิวเตอร์แบบภาษาเชิงวัตถุ

---

- 1) การลดระยะเวลาในการพัฒนาโปรแกรมคอมพิวเตอร์โดยสามารถนำโค้ดโปรแกรมที่พัฒนาไว้กลับมาใช้งานร่วมกัน
- 2) การนำเอาโปรแกรมคอมพิวเตอร์ที่พัฒนาอยู่มาใช้งาน(reuse-ability)
- 3) การพัฒนาต่อยอดจากวัตถุ(object) ที่มีอยู่แล้ว
- 4) การดูแลแก้ไขทำได้ง่าย สามารถทำการแก้ไขเฉพาะ object ที่ต้องการเท่านั้น จะไม่กระทบต่อส่วนอื่นๆ ของโปรแกรมที่พัฒนาขึ้น
- 5) การพัฒนาแบบภาษาเชิงวัตถุ มีความเหมาะสมสำหรับระบบที่มีความซับซ้อน
- 6) การพัฒนาแบบภาษาเชิงวัตถุสามารถใช้คำหรือชื่อ ต่างๆ ที่มีอยู่ในระบบงานมาใช้ในการออกแบบโปรแกรม ทำให้เกิดความเข้าใจง่ายขึ้น
- 7) การพัฒนาแบบภาษาเชิงวัตถุมีความยืดหยุ่น สามารถปรับเปลี่ยนการทำงานของเดิมที่มีอยู่แล้วได้โดยไม่ต้องแก้ไขโปรแกรมคอมพิวเตอร์เดิม



# ใบงาน

ให้นักศึกษาแบ่งกลุ่มละ 2-3 คน อธิบายการพัฒนาโปรแกรมคอมพิวเตอร์แบบภาษาเชิงวัตถุ พร้อมยกตัวอย่างประกอบ และนำเสนอหน้าชั้นเรียน



## ข้อมูลพื้นฐานในระบบ 2 มิติ และ 3 มิติ

---

**มิติ** ความหมายโดยทั่วไปหมายถึง สิ่งที่บอกคุณสมบัติของวัตถุ ได้แก่ ความกว้าง ความยาว และ ความสูง ส่วนในทางคณิตศาสตร์ มิติ หมายถึงจำนวนตัวเลขที่ต้องการเพื่อระบุตำแหน่งและคุณสมบัติของวัตถุใด ๆ ในปริภูมิ ในศาสตร์ต่าง ๆ อาจนิยามความหมายของคำว่า มิติ แทนจำนวนพารามิเตอร์ที่เกี่ยวข้อง เช่น ต้นทุน และ ราคา ในทางเศรษฐศาสตร์

ตัวอย่างในทางภูมิศาสตร์เช่น จุดบนพื้นผิวโลก สามารถกำหนดได้โดยตัวเลขค่าละติจูดและลองจิจูด ทำให้แผนที่ดังกล่าวมีสองมิติ (ถึงแม้ว่าโลกจะมีรูปร่างเกือบทรงกลมซึ่งมีสามมิติก็ตาม) ในการกำหนดตำแหน่งเครื่องบินหรืออากาศยานอื่น นอกจากละติจูดและลองจิจูดแล้ว ยังมีอีกตัวแปรหนึ่งคือค่า ความสูงจากพื้นดิน ทำให้พิกัดของเครื่องบิน เป็นสามมิติ

เวลา สามารถใช้เป็นมิติที่สามหรือที่สี่ (เพิ่มจากพื้นที่สองหรือสามมิติเดิม) ในการกำหนดตำแหน่งได้



# มิติในทางคณิตศาสตร์และฟิสิกส์

---

ศูนย์มิติ:

จุด

ปริภูมิช่องว่างศูนย์มิติ ในทางคณิตศาสตร์, ทอพอโลยี

จำนวนเต็ม

หนึ่งมิติ:

เส้น

กราฟ

จำนวนจริง

สองมิติ:

จำนวนเชิงซ้อน

ระบบพิกัดคาร์ทีเซียน (ระบบพิกัดสี่เหลี่ยมผืนผ้า)

ยูนิฟอร์ม ไทลิ่ง ในคณิตศาสตร์, เรขาคณิตยูคลิด

พื้นผิว ในทางคณิตศาสตร์, ทอพอโลยี





# มิติในทางคณิตศาสตร์และฟิสิกส์

---

สามมิติ:

ทรงตันเพลโต (ทรงนูนหลายหน้าปรกติ), เรขาคณิต

Ph สเตอริโอสโคปี (ภาพ 3 มิติ)

Ph มุมของออยเลอร์ (ทรงตันในอุดมคติ)

ช่องว่าง 3 มิติ ในคณิตศาสตร์นามธรรม, ทอพอโลยี

ปม ในทางคณิตศาสตร์

สี่มิติ:

Ph, ปริภูมิ-เวลา (ทฤษฎีสัมพัทธภาพ)

Ph, เทสเซอร์แควคต์ (ช่องว่าง 4 มิติ; มิติที่ 4 ของอวกาศ)

พอลิโครอน (พอลิโทป 4 มิติ) คณิตศาสตร์, ทอพอโลยี

ควอเทอร์เนียน (จำนวนเชิงซ้อน ในคณิตศาสตร์ทฤษฎีจำนวน)

ช่องว่าง 4 มิติ ในคณิตศาสตร์นามธรรม, ทอพอโลยี



# มิติในทางคณิตศาสตร์และฟิสิกส์

มิติที่สูงขึ้นไปในทางคณิตศาสตร์:

ออกโคโตเนียน (จำนวนจริง และ พีชคณิต)

เวกเตอร์

ช่องว่าง (ช่องว่างในทาง คณิตศาสตร์นามธรรม)

ช่องว่างกาลาไ-บาย

มิติที่สูงขึ้นไปในทางฟิสิกส์:

Ph ทฤษฎีควาลูซ่า-ไคลน์ (แรงพื้นฐานของ แรงโน้มถ่วง, แรงแม่เหล็กไฟฟ้า)

Ph ทฤษฎีสตริง (ฟิสิกส์ทฤษฎีและ คณิตศาสตร์ฟิสิกส์)

Ph ทฤษฎีเอ็ม (ฟิสิกส์ทฤษฎี)

มิติที่ไม่มีที่สิ้นสุด (มิติที่เป็นอนันต์) :

Ph ช่องว่างของ ฮิลแบร์ท

ช่องว่าง ฟังก์ชัน