

## บทที่ 2

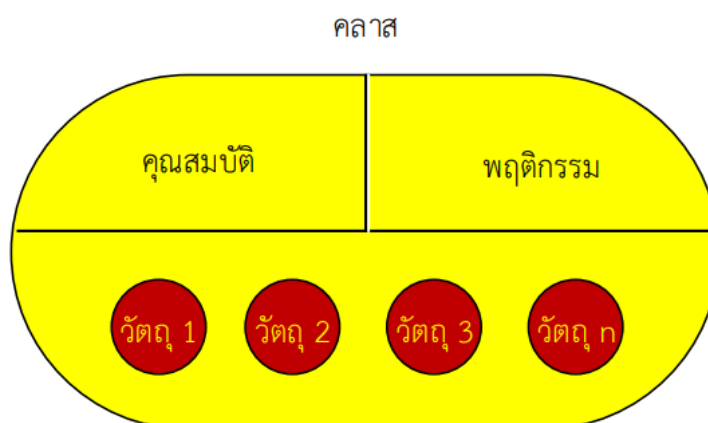
### เอกสารและงานวิจัยที่เกี่ยวข้อง

การจัดทำโครงการ โครงการพัฒนาระบบจัดการข้อมูลสาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏกาญจนบุรี ในการพัฒนาระบบจัดการข้อมูลนั้นต้องอาศัยหลักการและทฤษฎีต่าง ๆ รวมทั้งศึกษา แนวทางในการพัฒนาจากงานวิจัยที่เคยมีการพัฒนามาแล้วไปประยุกต์ใช้ในการพัฒนาระบบต่อไป ดังนั้นในบทนี้กล่าวถึงระเบียบวิธีการพัฒนาระบบสารสนเทศ การกำหนดขั้นตอนในการดำเนินงาน ทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบงาน และงานวิจัยที่เกี่ยวข้องกับระบบจัดการข้อมูล

#### 1. การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ [1] หมายถึง การมองการเขียนโปรแกรมเป็นลักษณะของวัตถุ มากกว่าการกระทำ การเขียนโปรแกรมเชิงวัตถุจะแยกปัญหาหรือระบบงานออกเป็นส่วน เพื่อลดความซับซ้อนให้น้อยลง โดยมุ่งที่จะเพิ่มประสิทธิภาพในการสร้างโปรแกรม ทำให้การพัฒนาโปรแกรมด้วยวิธีนี้ใช้เวลาสั้นลงและต้นทุนต่ำลง โดยการนำชุดคำสั่งที่สร้างไปแล้วกลับมาใช้งานใหม่ได้ และยังหาจุดที่ต้องการแก้ไขในโปรแกรมได้ง่ายขึ้น เป็นวิธีการเขียนโปรแกรมที่น่าแนวคิด ในความเป็นจริงที่อยู่รอบตัวมาใช้ในการเขียนโปรแกรม

การเขียนโปรแกรมเชิงวัตถุ มีแนวคิดและองค์ประกอบมาจากความหมายของเชิงวัตถุประกอบไปด้วย 4 ส่วนสำคัญ ได้แก่ คลาส วัตถุ คุณสมบัติ และพฤติกรรม โดยมีรายละเอียดดังภาพที่ 2.1



ภาพที่ 2.1 ส่วนประกอบของเชิงวัตถุ

จากภาพที่ 2.1 ส่วนประกอบของเชิงวัตถุ โดยแนวคิดประกอบไปด้วยคลาส ภายใน คลาสนั้นจะประกอบด้วยคุณสมบัติและพฤติกรรม กำหนดคุณลักษณะของวัตถุวัตถุใดที่มีคุณสมบัติ หรือพฤติกรรมเดียวกับคลาส ส่วนประกอบของเชิงวัตถุ มีรายละเอียดดังต่อไปนี้

### 1.1 คลาส (Class)

คลาส หมายถึงศัพท์ที่อธิบายโครงสร้างหรือพิมพ์เขียวที่กำหนดขอบเขตกลุ่มของวัตถุ มีความหมายและคำจำกัดความกลุ่มของวัตถุกลุ่มใดกลุ่มหนึ่งมีรายละเอียดดังต่อไปนี้

#### แม่แบบ



คุณสมบัติ : แขน ขา ปาก หู ตา จมูก  
รหัสบัตร ชื่อ เพศ อายุ

พฤติกรรม : วิ่ง หัวเราะ ประหม่อ สายหัว พูด ชก กระโดด

### ภาพที่ 2.2 คลาส

จากภาพที่ 2.2 ตัวอย่างคลาสมนุษย์ เมื่อมองในความหมายขอบเขตของมนุษย์ ก็จะเป็นลักษณะของสิ่งมีชีวิตที่เป็นรูปธรรม เช่น แขน ขา หู ตา จมูก เป็นต้น หรือคุณสมบัติด้านนามธรรม เช่น ชื่อ อายุ เพศ น้ำหนัก ส่วนสูง เป็นต้น ในลักษณะของคลาสนั้นบ่งบอกลักษณะภายในตัวของคลาส และในส่วนที่สอง เป็นส่วนของการมองในส่วนของคลาสนั้นสามารถเกิดการกระทำ จากตัวอย่างสามารถที่จะกระทำหรือแสดงพฤติกรรมออกมาได้ เช่น วิ่ง หัวเราะ ประหม่อ พูด กระโดด เป็นต้น ซึ่งเมื่อมองในความหมายภาพรวมเห็นได้ว่าเป็นลักษณะของมุมมองที่เป็นนามธรรมที่กำหนดขอบเขตของกลุ่มใดกลุ่มหนึ่ง

### 1.2 คุณสมบัติ (Properties)

คุณสมบัติ หมายถึงองค์ประกอบภายในคลาส ใช้ในการกำหนดข้อมูลที่เก็บภายในคลาส หรือส่วนที่กำหนดคุณลักษณะ ข้อกำหนดของคลาสนั้นมีแต่ละคลาสจะกำหนดคุณสมบัติที่ซึ่งกำหนดสถานะของวัตถุที่เกิดจากคลาสนั้น มีรายละเอียดดังต่อไปนี้

แม่แบบ



คุณสมบัติ : แขน ขา ปาก หู ตา จมูก  
รหัสบัตร ชื่อ เพศ อายุ

ภาพที่ 2.3 คุณสมบัติ

จากภาพที่ 2.3 ตัวอย่างคุณสมบัติที่เกิดจากคลาสมนุษย์ มนุษย์จะมีคุณสมบัติภายในคลาสของมนุษย์นั้นมี เช่น แขน ขา ชื่อ รหัสบัตร ที่อยู่ อายุ เพศ เป็นต้น คุณสมบัติเป็นองค์ประกอบที่สำคัญของคลาส โดยจะกำหนดคุณลักษณะที่มีของคลาสนั้น คุณสมบัติบางอย่างจะส่งผลกระทบต่อเนื่องไปยังพฤติกรรมที่เกิดขึ้นของคลาส

### 1.3 พฤติกรรม (Behavior)

พฤติกรรม หรือการกระทำ (Action) หมายถึงองค์ประกอบที่อยู่ภายในคลาสใช้ในการกำหนดการกระทำบางอย่างของคลาส พฤติกรรมเป็นการกำหนดขอบเขตที่สามารถกระทำได้ของคลาส ส่วนมากจะเกิดจากคุณสมบัติที่มีอยู่ในคลาส พฤติกรรม มีรายละเอียดดังต่อไปนี้

แม่แบบ



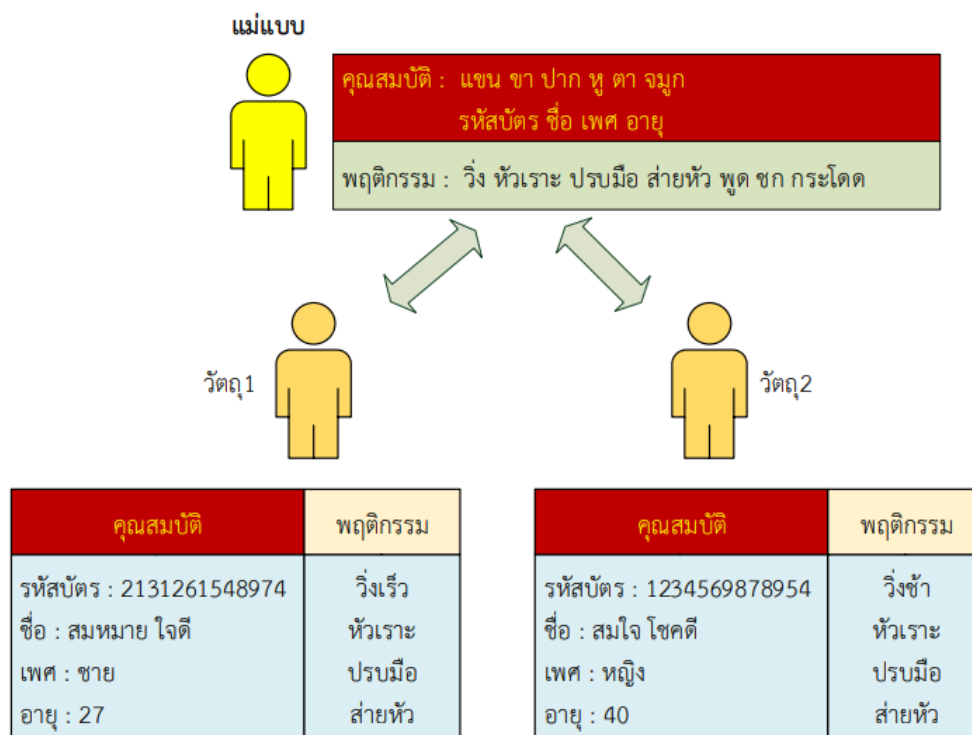
พฤติกรรม : วิ่ง หัวเราะ ปรบมือ ส่ายหัว พุด ชก กระโดด

ภาพที่ 2.4 พฤติกรรม

จากภาพที่ 2.4 ตัวอย่างพฤติกรรมที่เกิดจากคลาสมนุษย์ พฤติกรรมมักเกิดจากคุณสมบัติที่มีภายในคลาส เช่น คุณสมบัติมีขาจะสามารถวิ่งได้ มีปากก็สามารถพูดได้ เป็นต้น พฤติกรรมเป็นกระบวนการที่เกิดขึ้นตั้งแต่นำเข้าข้อมูล ประมวลผลและเกิดผลลัพธ์

#### 1.4 วัตถุ (Object)

วัตถุ หมายถึงการมองสิ่งที่อยู่รอบข้างเป็นวัตถุเป้าหมาย ภายในวัตถุจะประกอบไปด้วย คุณสมบัติของวัตถุและพฤติกรรมหรือการกระทำ ที่ตอบสนองต่อเหตุการณ์ที่จะทำให้เกิดขึ้นของวัตถุ มีรายละเอียดดังต่อไปนี้



ภาพที่ 2.5 วัตถุ

จากภาพที่ 2.5 วัตถุ1 และวัตถุ2 จะมีคุณสมบัติเหมือนกัน คือ รหัสบัตร ชื่อ เพศและอายุ แต่ข้อมูลที่อยู่ภายในคุณสมบัตินี้ จะมีความแตกต่างกันที่บ่งบอกคุณลักษณะเฉพาะของวัตถุ

## 2. ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล [2] คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่าง ผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไข

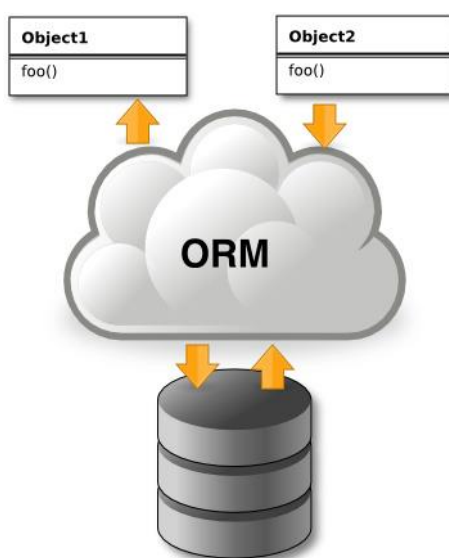
ฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล ประโยชน์ของฐานข้อมูลประกอบไปด้วย

- 1) ลดการเก็บข้อมูลที่ซ้ำซ้อน ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลายแห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ ความซ้ำซ้อนของข้อมูลลดน้อยลง
- 2) รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ในกรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องตามกันหมดโดยอัตโนมัติ
- 3) การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้นซึ่งก่อให้เกิดความปลอดภัยของข้อมูลด้วย

### 3. โออาร์เอ็ม (ORM)

ORM (Object / Relational Mapping) [3] หมายถึง การ Map ข้อมูลในตารางข้อมูลของฐานข้อมูลให้อยู่ในรูปของ Object-Oriented Language ซึ่งจะเป็นการสร้าง Database แบบเสมือนขึ้นให้มาอยู่ในรูป Language Programming ทำให้ไม่ต้องไปยุ่งกับ SQL Statement ซึ่งถ้ามีการเปลี่ยนแปลงแก้ไขต้องแก้ไขที่ตัวโปรแกรมแทน และการกระทำต่าง ๆ ยังคงเป็นแบบ Relational แบบเดิม เช่น มีตารางชื่อ Product ก็จะแปลงมาเป็น Class Product คุณลักษณะ (Attributes) ของตาราง Product ก็จะถูแปลงมาเป็น Object ใน Class

เราไม่จำเป็นต้องยุ่งในส่วนของ SQL (SELECT, INSERT, UPDATE, DELETE) แต่เราจะใช้ Framework มาช่วยจัดการแปลงจากภาษาโปรแกรมมิ่ง ไปเป็น SQL หรือ Database ให้เราแทน



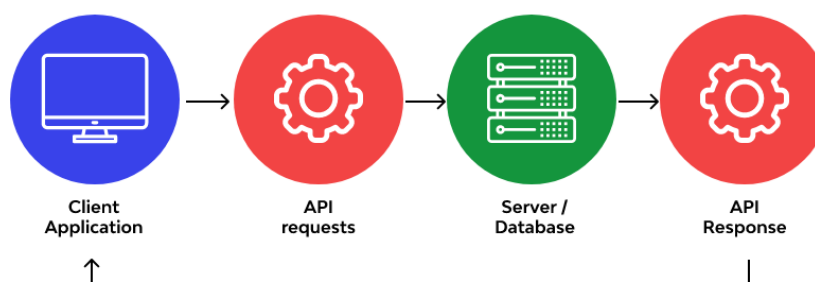
ภาพที่ 2.6 ORM: Object Relational Mapping

จากภาพที่ 2.6 ORM [4] เป็นตัวเอาไว้สลับระหว่าง ข้อมูลที่อยู่ในรูปแบบ Object แปลงไปเป็น ข้อมูลที่อยู่ในรูปแบบ ความสัมพันธ์ และดึงข้อมูลที่อยู่ในรูปแบบความสัมพันธ์ กลับไปเป็น ข้อมูลที่อยู่ในรูปแบบ Object และมีเครื่องมือที่พัฒนาตัว ORM นี้กันอยู่หลายตัวอยู่แล้ว ไม่จำเป็นต้องไม่ สร้างขึ้นมาเอง

#### 4. เอฟพีไอ (API: Application Programming Interface)

##### 4.1 API [5]

เป็นบริการช่องทางการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลจากระบบหนึ่งไปสู่ระบบอื่น ๆ ที่มีความรวดเร็ว ปลอดภัย หน้าที่หลักของ API คือ คอยรับคำสั่งจากฝั่งลูกข่าย (Client ) ซึ่งก็คือ Application ต่าง ๆ เช่น Web Application, Mobile Application, Desktop Application เป็นต้น เมื่อฝั่งลูกข่ายส่งคำสั่ง จะเรียกว่าการ Request จากนั้น ตัว API จะรับคำสั่งดังกล่าว นำไปประมวลผล และสรุปเป็นข้อมูลที่ตรงกับ Request และส่งข้อมูลเหล่านั้นกลับไปให้ฝั่งลูกข่าย หรือ Application เพื่อนำไปใช้งานต่อไป



ภาพที่ 2.7 การทำงานของ API

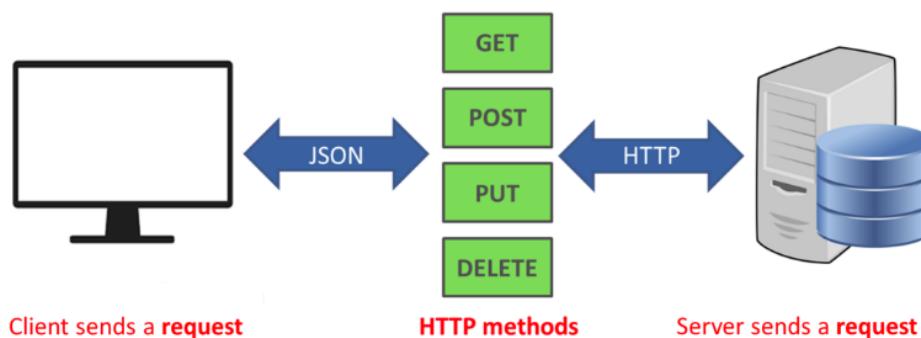
จากภาพที่ 2.7 API เป็นช่องทางในการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลจากระบบหลักไปสู่ระบบ Client อื่นๆ หน้าที่หลักของ API จะคอยรับคำสั่งจากฝั่ง Client ซึ่งก็คือแอปพลิเคชันต่าง ๆ จะมีการรับคำสั่งไปประมวลผลและสรุปข้อมูลให้ตรงกับ Request ที่ต้องการ ก่อนจะส่งข้อมูลนั้นกลับไปให้ Client หรือตัวแอปพลิเคชันนั้น ๆ เพื่อใช้งานต่อไป

##### 4.2 REST API [6]

REST (Representational State Transfer) เป็นรูปแบบการส่งข้อมูลระหว่าง Server-Client รูปแบบหนึ่งซึ่งอยู่บนพื้นฐานของ HTTP Protocol เป็นการสร้าง Web Service เพื่อแลกเปลี่ยนข้อมูลกันผ่าน Application วิธีหนึ่ง ซึ่งส่งข้อมูลได้หลายชนิด ไม่ว่าจะเป็น Text, XML, JSON โดย

ส่วนใหญ่แล้วจะเลือกชนิด JSON ด้วยความที่รองรับได้ทั้งหลายรูปแบบไม่ว่าจะเป็น Browser หรือ Mobile เพื่อระบุและทำงานกับทรัพยากร โดยที่เมธอดที่สำคัญใน REST API ประกอบด้วย

- 1) GET: ใช้สำหรับเรียกดูข้อมูลทรัพยากร
- 2) POST: ใช้สำหรับสร้างข้อมูลใหม่ทรัพยากร
- 3) PUT: ใช้สำหรับอัปเดตข้อมูลของทรัพยากรที่มีอยู่แล้ว
- 4) DELETE: ใช้สำหรับลบทรัพยากร



ภาพที่ 2.8 REST API

จากภาพที่ 2.8 REST API อาศัยรูปแบบของ HTTP Method เช่น GET POST PUT DELETE ในการทำงาน และจะส่งค่ากลับมาเป็น JSON ส่งผลให้สามารถรับส่งข้อมูลไปมาข้าม Platform ได้

## 5. เว็บแอปพลิเคชัน (Web application)

เว็บแอปพลิเคชัน คือ โปรแกรมประยุกต์ที่จะเข้าถึงด้วยโปรแกรม Internet Browser ซึ่งทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time จะพบข้อดีของเว็บแอปพลิเคชัน คือ ข้อมูลต่างๆ ที่อยู่ในระบบที่มีการไหลเวียนในแบบ Online จึงสามารถโต้ตอบกับผู้ใช้บริการแบบ Real Time ทำให้เกิดความประทับใจ รวมทั้งสามารถใช้งานได้ง่ายโดยไม่จำเป็นต้องติดตั้ง Client Program จะทำให้ไม่ต้อง Upgrade Client Program และสามารถใช้งานผ่าน Internet Connection ที่มีความเร็วต่ำกว่า ส่งผลให้ผู้ใช้บริการสามารถใช้โปรแกรมได้จากทุกที่ ตัวอย่างระบบออนไลน์ที่เหมาะสมกับเว็บแอปพลิเคชัน เช่น ระบบการจองสินค้าหรือบริการต่าง ๆ ระบบงานบุคลากร ระบบงานแผนการตลาด ระบบการสั่งซื้อแบบพิเศษ และระบบงานในโรงเรียน เป็นต้น

## 6. รีแอคต์ (React)

React คือ JavaScript Library [7] โดยเป็นเครื่องมือที่ใช้สร้างเว็บสำหรับสร้าง UI (User Interface) หรือ หน้าจอแอปพลิเคชัน จุดเด่นของ React ที่นำมาใช้นั้นก็คือ ระบบแคชในตัว ทำให้หน้าเว็บของเรามีการตอบสนองที่เร็ว การเขียน React เรายังสามารถแยกองค์ประกอบของ

หน้าเว็บเราออกเป็นส่วน ๆ เรียกว่าเป็น Component แล้วนำมาประกอบกันเป็นหน้าเว็บได้ ซึ่งทำให้เราสามารถนำ Component ของเราไปใช้ซ้ำที่อื่นได้ โดยไม่ต้องเขียนขึ้นใหม่ โดย React จะแบ่งส่วนประกอบหลัก ๆ มีทั้งหมด 3 ส่วน ดังนี้คือ

### 6.1 คอมโพเนนท์ (Component)

คอมโพเนนท์ เป็นส่วนที่ใช้สร้างตัวอย่างของ UI ใน React โดยจะเป็นฟังก์ชันหรือคลาสที่จะคืนค่า UI ที่กำหนดรูปแบบและสิ่งที่จะแสดงบนหน้าเว็บ เราสามารถสร้าง Component ใหม่ ๆ และนำมาใช้ซ้ำในแอปพลิเคชันได้

```
function MyComponent(props) {
  return (
    <div>
      <h1>Hello, {props.name}!</h1>
      <p>{props.message}</p>
    </div>
  );
}
```

ภาพที่ 2.9 React-Component

จากภาพที่ 2.9 ตัวอย่างของฟังก์ชัน Component ในรูปแบบของ Function Component ใน React โดยมีชื่อว่า MyComponent ซึ่งรับพารามิเตอร์ props ภายในฟังก์ชันในส่วนของการแสดงผล ของ Component นี้ มี HTML และ JSX ที่กำหนดรูปแบบและเนื้อหาที่จะแสดงบนหน้าเว็บ

### 6.2 สเตท (State)

สเตท ใน React เป็นข้อมูลที่ถูกเก็บไว้ภายใน Component แต่ละ Component สามารถมี สเตทของตัวเองได้โดยอิสระ ซึ่งสเตทนี้สามารถเปลี่ยนแปลงได้และจะทำให้ React รีเรนเดอร์หรืออัปเดตส่วนของ UI ที่เกี่ยวข้อง

```
function Counter() {
  const [count, setCount] = useState(0);

  const increment = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={increment}>Increment</button>
    </div>
  );
}
```



### ภาพที่ 2.10 React-State

จากภาพที่ 2.10 ตัวอย่างนี้เราใช้ useState hook เพื่อสร้าง State ที่ชื่อ count และ setCount สำหรับอัปเดตค่า count เมื่อมีการคลิกที่ปุ่มจะเพิ่มค่า count ผ่านฟังก์ชันที่ชื่อ increment

#### 6.3 พรอปส์ (Props: Properties)

พรอปส์ ใน React คือข้อมูลที่ถูกส่งจากคอมโพเนนต์ หนึ่งไปยัง คอมโพเนนต์อื่น โดยใช้ ตัวแปรที่สามารถรับค่าได้เป็นพารามิเตอร์ของฟังก์ชันคอมโพเนนต์ นั้น ๆ ซึ่งพรอปส์ถูกใช้เพื่อส่ง ข้อมูลระหว่างคอมโพเนนต์แม่ และคอมโพเนนต์ลูก เพื่อให้ คอมโพเนนต์ลูกสามารถใช้ข้อมูลจาก คอมโพเนนต์แม่ได้

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

function App() {
  return <Greeting name="John" />;
}
```

### ภาพที่ 2.11 React-Props

จากภาพที่ 2.11 ตัวอย่างนี้เราสร้าง Function Component ชื่อ Greeting ที่มี พรอปส์ ชื่อ name เพื่อ แสดงข้อความ โดยข้อความจะขึ้นอยู่กับค่า name ที่ถูกส่งมาในพรอปส์ของคอมโพเนนต์นี้และใน ตัวอย่างนี้เราสร้าง Function Component ชื่อ App ซึ่งนำคอมโพเนนต์ Greeting มาใช้และส่ง พรอปส์ name ให้กับคอมโพเนนต์ Greeting โดยในตัวอย่างนี้เราส่งค่า "John" ในพรอปส์ name

## 7. ภาษาซีชาร์ป (C# Programming Language)

ภาษาคอมพิวเตอร์ประเภท Object-Oriented Programming [8] พัฒนาโดย Microsoft โดยมี จุดมุ่งหมายในการรวมความสามารถการคำนวณของภาษาซีพลัสพลัส (C++) ด้วยการโปรแกรมง่าย กว่าของ Visual Basic โดยภาษาซีชาร์ปมีพื้นฐานจากภาษาซีพลัสพลัส และเก็บส่วนการทำงาน คล้ายกับภาษาจาวา (Java) โครงสร้างของภาษาซีชาร์ป ประกอบด้วยส่วนต่าง ๆ ดังนี้

#### 7.1 คลาส

คลาส ในภาษาซีชาร์ปใช้สำหรับกำหนดโครงสร้างและพฤติกรรมของอ็อบเจกต์ ที่จะถูก สร้างขึ้นตามแม่แบบที่กำหนดไว้ โดยสามารถขยายและเพิ่มคุณสมบัติตามความต้องการ ตัวอย่างดัง ภาพที่ 2.12

```

class MyClass
{
    // ตัวแปรสมาชิก (Member Variables)
    private int myVariable;

    // คอนสตรัคเตอร์ (Constructor)
    public MyClass()
    {
        // การกำหนดค่าเริ่มต้นสำหรับตัวแปรสมาชิก
        myVariable = 0;
    }

    // เมท็อด (Methods)
    public void MyMethod()
    {
        // การดำเนินการต่าง ๆ
        Console.WriteLine("Hello, World!");
    }
}

```

ภาพที่ 2.12 โครงสร้างคลาสในภาษาซีชาร์ป

จากภาพที่ 2.12 รายละเอียดโครงสร้างดังนี้

- 1) MyClass คือชื่อของคลาส
- 2) myVariable คือตัวแปรสมาชิกที่เป็นตัวแปรที่ถูกเก็บเฉพาะภายในคลาส
- 3) MyClass() คือคอนสตรัคเตอร์ (Constructor) ที่ใช้สร้างอ็อบเจกต์ของคลาส ในตัวอย่างนี้จะกำหนดค่าเริ่มต้นของ myVariable เป็น 0
- 4) MyMethod() คือเมท็อด (Method) ที่ใช้สำหรับการดำเนินการต่าง ๆ ภายในคลาส

## 7.2 อ็อบเจกต์

อ็อบเจกต์ เป็นตัวแทนของคลาสที่ถูกสร้างขึ้นจากแม่แบบคลาส โดยมีคุณสมบัติและพฤติกรรมตามที่กำหนดไว้ในคลาสนั้น โดยการสร้างอ็อบเจกต์จะใช้คีย์เวิร์ด new ตามด้วยชื่อของคลาสและวงเล็บว่างเปล่า () ที่เรียกว่าคอนสตรัคเตอร์ของคลาส ตัวอย่างดังภาพที่ 2.14

```

class Person
{
    public string name;
    public int age;

    public Person(string n, int a)
    {
        name = n;
        age = a;
    }

    public void SayHello()
    {
        Console.WriteLine("Hello, my name is " + name + " and I am " + age
    }
}

Person person1 = new Person("John", 25);

person1.SayHello();

```

ภาพที่ 2.14 การสร้างอ็อบเจกต์ในภาษาซีชาร์ป

จากภาพที่ 2.14 ตัวอย่างคลาสชื่อ Person ที่มีคุณสมบัติ name และ age และเมทอด SayHello() เมื่อเราสร้างอ็อบเจกต์ของคลาส Person โดยใช้คอนสตรัคเตอร์ Person() แล้วกำหนดค่า name และ age ตามลำดับ จากนั้นจะสามารถเรียกใช้เมทอด SayHello() ของอ็อบเจกต์ person1

### 7.3 เมทอด

เมทอด เป็นส่วนที่ใช้ในการกำหนดการกระทำหรือการดำเนินการที่คลาสหรืออ็อบเจกต์สามารถทำได้ โดยสามารถรับพารามิเตอร์เข้ามาเพื่อประมวลผลและส่งค่ากลับ ดังภาพที่ 2.15

```

class Calculator
{
    public int Add(int a, int b)
    {
        int result = a + b;
        return result;
    }

    public int Multiply(int a, int b)
    {
        int result = a * b;
        return result;
    }
}

```

ภาพที่ 2.15 เมทอดในภาษาซีชาร์ป

จากภาพที่ 2.15 ตัวอย่างเมทอดในคลาส Calculator ที่มีเมทอด Add และ Multiply ในตัวอย่างนี้ จะมีคลาส Calculator ที่มีเมทอด Add และ Multiply ซึ่งรับพารามิเตอร์ a และ b เพื่อประมวลผลและส่งค่าผลลัพธ์กลับ โดยสามารถเรียกใช้เมทอดเหล่านี้ได้ดังภาพที่ 2.16

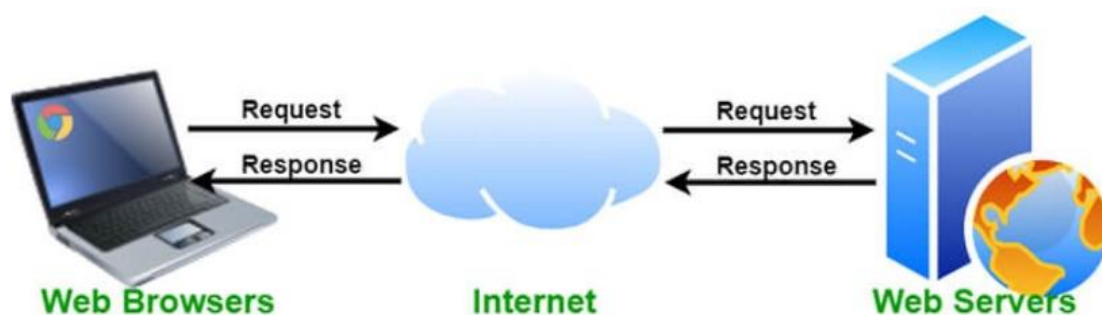
```
Calculator calculator = new Calculator();
int sum = calculator.Add(5, 3);
int product = calculator.Multiply(4, 2);
```

ภาพที่ 2.16 ตัวอย่างการใช้งานเมทอด

จากภาพที่ 2.16 สร้างอ็อบเจกต์ calculator จากคลาส Calculator และเรียกใช้เมทอด Add และ Multiply เพื่อทำการบวกและคูณตัวเลข กลับมาเก็บผลลัพธ์ในตัวแปร sum และ product ตามลำดับ

## 8. เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์ [9] คือ เครื่องคอมพิวเตอร์ ที่ติดตั้งโปรแกรมคอมพิวเตอร์ ซึ่งทำหน้าที่ให้บริการข้อมูลแก่ เครื่องคอมพิวเตอร์ที่ขอรับบริการ ในรูปแบบสื่อผสม ผ่านระบบเครือข่าย โดยผู้ใช้งานติดต่อสื่อสารกับเว็บเบราว์เซอร์ โดยผ่านทางโปรแกรมเว็บเซิร์ฟเวอร์ และมีการเชื่อมต่อผ่านอินเทอร์เน็ต



ภาพที่ 2.13 การทำงานของเว็บเซิร์ฟเวอร์

จากภาพที่ 2.8 เป็นการแสดงขั้นตอนการทำงานของระบบ โดยเริ่มจากผู้ใช้งานได้ทำการร้องขอบริการด้านข้อมูลกับเซิร์ฟเวอร์ซึ่งผู้ใช้งานสามารถติดต่อสื่อสารผ่านทางระบบอินเทอร์เน็ต และในปัจจุบันสามารถเรียกใช้เว็บเซิร์ฟเวอร์ผ่านทางโทรศัพท์เคลื่อนที่ได้

## บรรณานุกรม

- [1] อาจารย์ ดร.เอกราช ธรรมษา. (2559). การเขียนโปรแกรมเชิงวัตถุ. [http://www.com-tech.ubru.ac.th/~ekarach/book/pcp/09\\_OOP.pdf](http://www.com-tech.ubru.ac.th/~ekarach/book/pcp/09_OOP.pdf).
- [2] Mindphp.com. (2565). ระบบฐานข้อมูลคืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2055-database-คืออะไร23.html>.
- [3] ณัฐดนัย อุปทอง. (2563). ความหมายของ ORM. <https://www.mindphp.com/คู่มือ/73-คืออะไร/3247-orm-doctrine.html>.
- [4] Box Single. (2556). ORM (Object Relational Mapping). [https://www.boxsingle.com/3/ORM-\(Object-Relational-Mapping\)-คืออะไร](https://www.boxsingle.com/3/ORM-(Object-Relational-Mapping)-คืออะไร)
- [5] Hayatee e-pong. (2565). API คืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/8082-what-is-an-api.html>.
- [6] Settawat Janpuk. (2562). ทำความรู้จักกับ REST API. <https://medium.com/@settawatjanpuk/beginner-มาทำความรู้จักกับ-rest-api-กันดีกว่า-มีตัวอย่าง-c855d93beaf3>.
- [7] Mos Programmer. (2562). รู้จัก React. <https://mosprogramer.medium.com/มารู้จัก-react-กันเถอะ-eb04ab8b8ec1>
- [8] Mindphp.com. (2560). ซีชาร์ป คือ ภาษาคอมไพเตอร์ ซึ่งพัฒนามาจากภาษา C++. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2184-c-ชาร์ป-คืออะไร.html>.
- [9] Mindphp.com. (2560). Web Server คืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2053-web-server-คืออะไร.html>.