



OOSD CLASS

DESIGN PATTERN

MVC



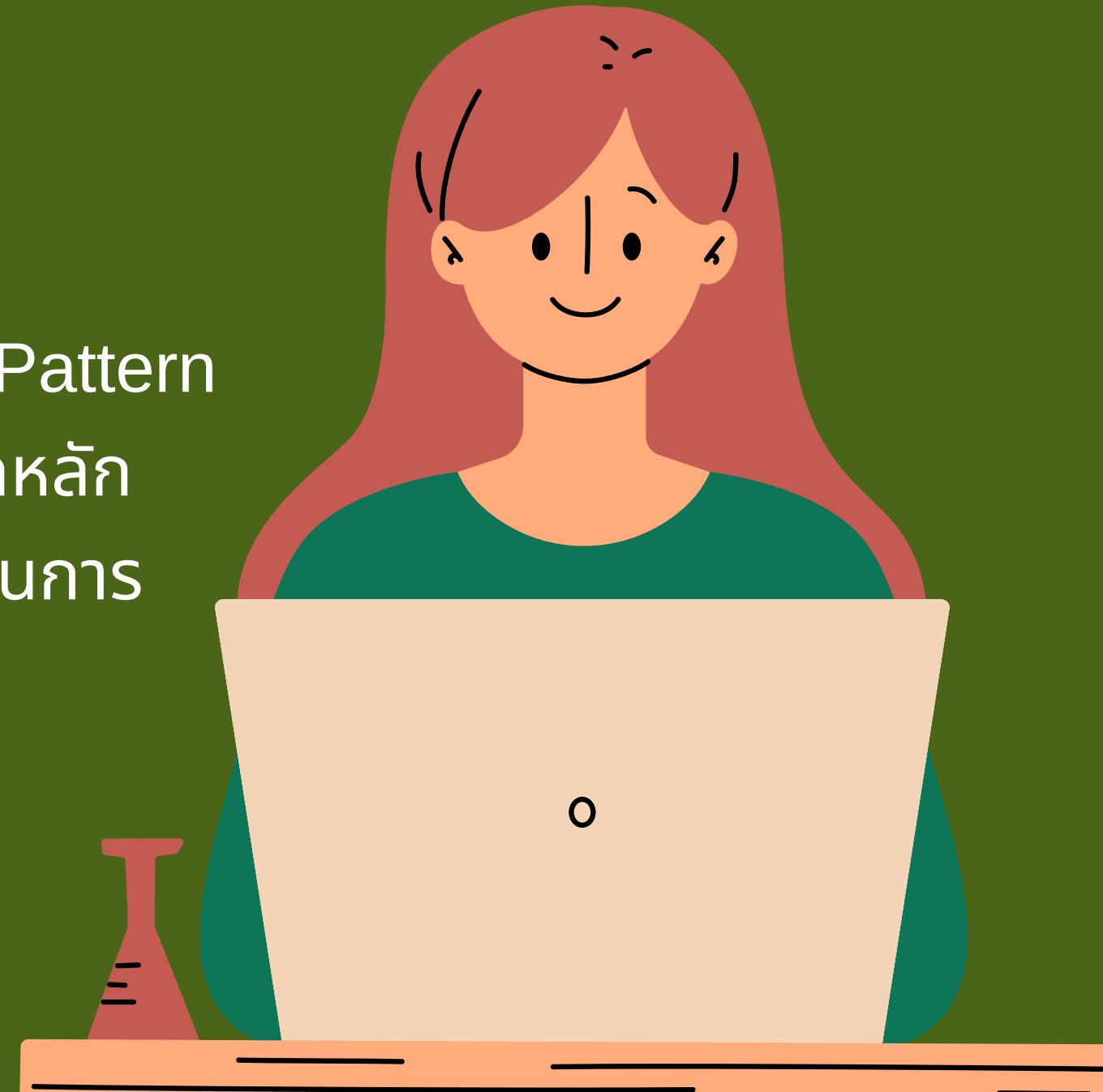
Next Page

w w w . c o p y a n d p a s t e . c o m

Presented By Group 24

MVC คืออะไร?

MVC ย่อมาจาก Model-View-Controller เป็น Software Design Pattern รูปแบบหนึ่งที่ได้รับความนิยมในโลกของการพัฒนาซอฟต์แวร์ โดยยึดหลัก ของการทำงานที่แยกส่วนอิสระต่อกันชัดเจน จึงทำให้มีความสามารถในการ จัดการและบำรุงรักษาแอปพลิเคชันได้ดี



Next Page

M (Model)

คือ ส่วนที่เกี่ยวข้องกับการจัดการ Data และ Business Logic เช่น การเขียนโลจิก การจัดการกับข้อมูลและฐานข้อมูล โดยไม่เด่นนั้นจะแยกออกจากกันกับ Interface ชัดเจน

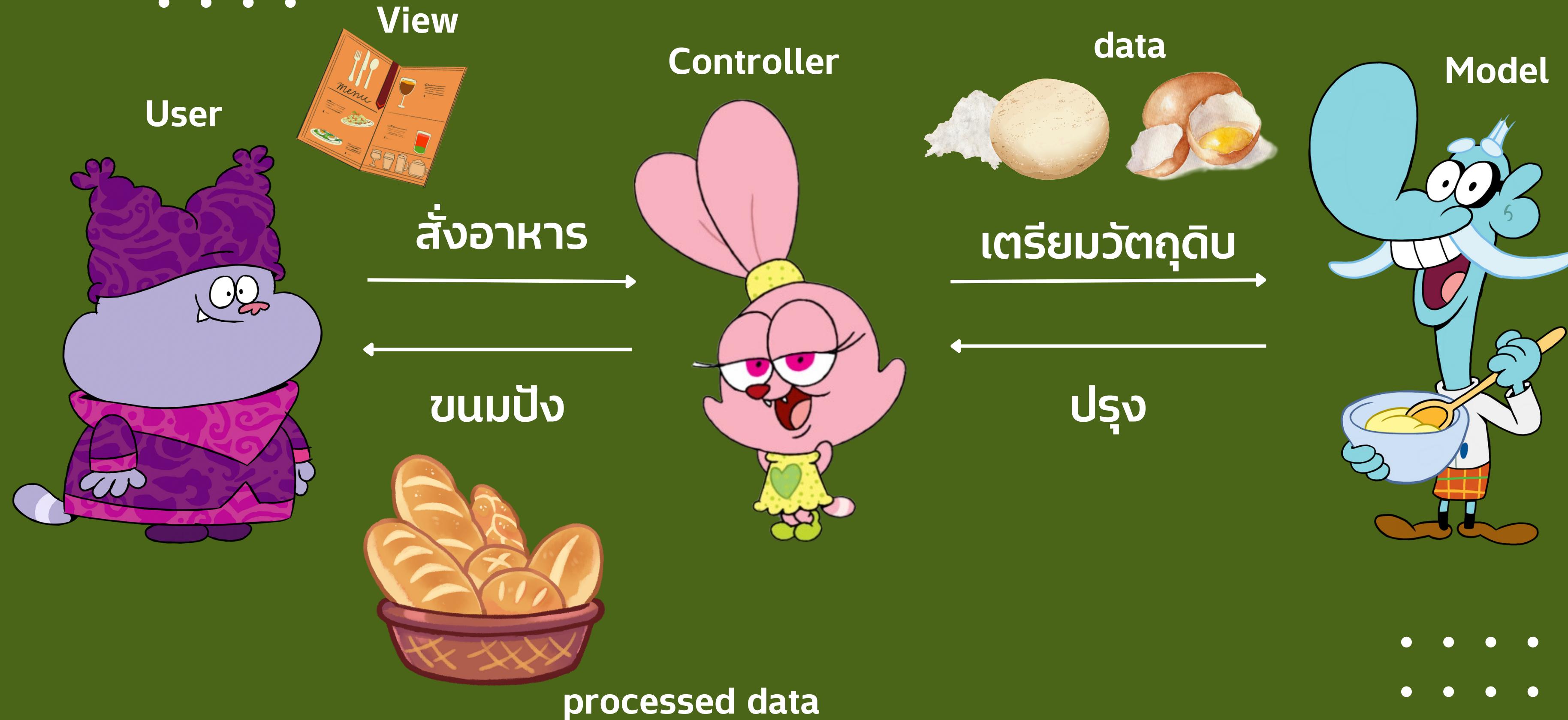
View (V)

คือ ส่วนที่เกี่ยวข้องกับ Interface ที่เป็นส่วนที่จะรับข้อมูลหรือแสดงผลข้อมูลจากผู้ใช้

Controller (C)

ทำหน้าที่เป็นตัวกลางระหว่าง Model และ View โดยจะรับ request จากผู้ใช้เข้ามาแล้วนำมาประมวลผลหรือตีความว่า request นั้นคืออะไร แล้วส่งต่อไปที่ Model ให้จัดการกับข้อมูลต่อไป จากนั้น Controller ก็จะทำหน้าที่ response (ตอบกลับ) สิ่งที่ผู้ใช้ได้ร้องขอมาต่อไป

Mung Daal Catering Company



JAVA

C++

CONNECTED

MVC เมماะกับงานแบบไหน?

MVC เมماะสำหรับงานที่ต้องการการแยกส่วนของโค้ดเพื่อควบคุมการทำงานและการแสดงผลข้อมูลในแอปพลิเคชันหรือโปรแกรมคอมพิวเตอร์ต่าง ๆ โดยทั่วไปแล้ว การใช้ MVC ช่วยให้โค้ดง่ายต่อการบริหารและการแก้ไข เพิ่มความยืดหยุ่นและการรวมงานที่มีได้ดีขึ้นในโปรเจกต์ขนาดใหญ่และซับซ้อน เช่น

- การพัฒนาเว็บแอปพลิเคชัน
- การพัฒนาโปรแกรมแบบอินเทอร์เฟซกราฟิก (GUI)
- การพัฒนาระบบฐานข้อมูล



ทำไมMVCถึงอยู่CATEGORYนี้?

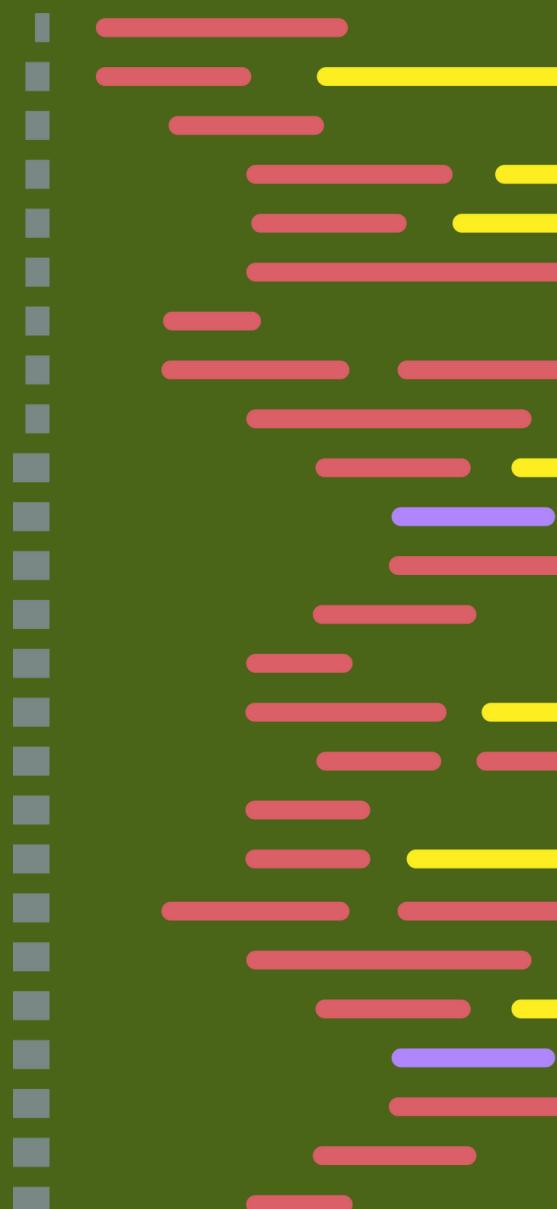
JAVA

Architectural Design Pattern เป็นรูปแบบการออกแบบที่เกี่ยวกับโครงสร้างและการจัดระเบียบของระบบโปรแกรมหรือแอปพลิเคชันทั้งหมด

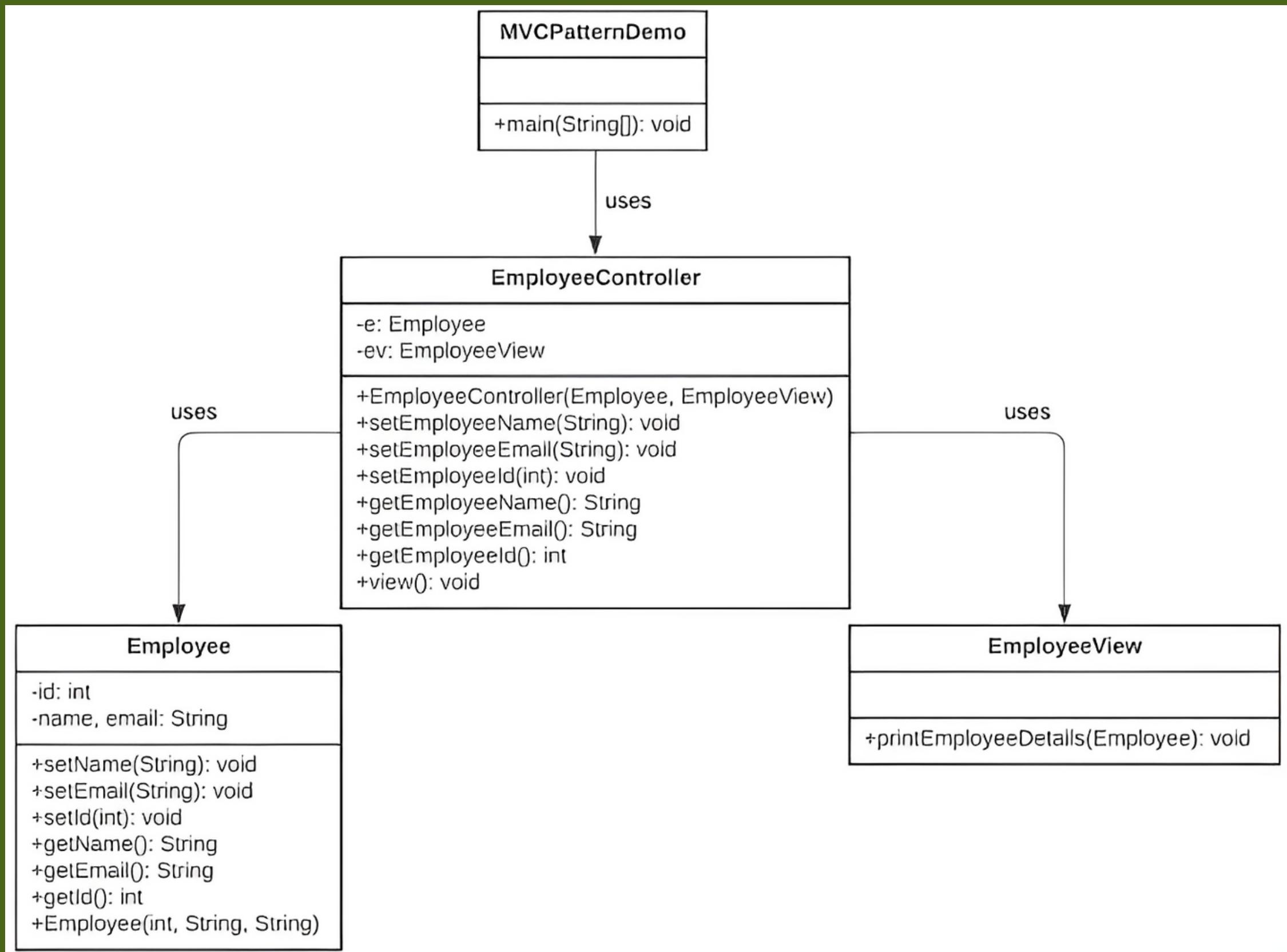
MVC ถูกออกแบบขึ้นเพื่อแยกส่วนหน้าที่และระบบข้อมูลของแอปพลิเคชันออกจากกัน ทำให้ง่ายต่อการบริหารและปรับปรุงระบบ นี่คือเหตุผลที่ MVC เรียกว่า "Architectural Pattern" ซึ่งช่วยกำหนดโครงสร้างและการจัดการกับความซับซ้อนของแอปพลิเคชันได้อย่างมีระเบียบและมีประสิทธิภาพ

C++

CONNECTED



Class Diagram



JAVA

C++

CONNECTED

EmployeeView

```
1
2 public class EmployeeView {
3     public void printEmployeeDetails(Employee e)
4     {
5         System.out.println("Employee no: " + e.getId());
6         System.out.println("Name: " + e.getName());
7         System.out.println("Email: " + e.getEmail());
8     }
9 }
10
```



EmployeeController

Page 12

```
1 public class EmployeeController {  
2     private Employee e;  
3     private EmployeeView ev;  
4     public EmployeeController(Employee e, EmployeeView ev){  
5         this.e = e;  
6         this.ev = ev;  
7     }  
8     public void setEmployeeName(String name){  
9         e.setName(name);  
10    }  
11    public void setEmployeeEmail(String email){  
12        e.setEmail(email);  
13    }  
14    public void setEmployeeId(int id){  
15        e.setId(id);  
16    }  
17    public String getEmployeeName(){  
18        return e.getName();  
19    }  
20    public String getEmployeeEmail(){  
21        return e.getEmail();  
22    }  
23    public int getEmployeeId(){  
24        return e.getId();  
25    }  
26    public void view(){  
27        ev.printEmployeeDetails(e);  
28    }  
29 }
```



Next Page

Employee

Page 13

```
1 public class Employee {  
2     private int id;  
3     private String name;  
4     private String email;  
5     public int getId(){  
6         return id;  
7     }  
8     public void setId(int id){  
9         this.id = id;  
10    }  
11    public String getName(){  
12        return name;  
13    }  
14    public void setName(String name){  
15        this.name = name;  
16    }  
17    public String getEmail(){  
18        return email;  
19    }  
20    public void setEmail(String email){  
21        this.email = email;  
22    }  
23    public Employee(int id, String name, String email) {  
24        this.id = id;  
25        this.name = name;  
26        this.email = email;  
27    }  
28 }
```



Next Page

MVCPatternDemo

```
1 public class MVCPatternDemo
2 {
3     Run | Debug
4     public static void main(String[] args)
5     {
6         EmployeeController ec = new EmployeeController(retreiveEmployee(), new EmployeeView());
7         ec.view();
8         ec.setEmployeeName(name:"Employee B");
9         ec.setEmployeeEmail(email:"employee.b@company.com");
10        ec.view();
11    }
12    private static Employee retreiveEmployee(){
13        return new Employee(id:1, name:"Employee A", email:"employee.a@company.com");
14    }
}
```

MVCPatternDemo

Employee no: 1

Name: Employee A

Email: employee.a@company.com

Employee no: 1

Name: Employee B

Email: employee.b@company.com



THANK YOU



640710528 ຮິມາພຣ ກຣັບຍົນ
design



640710533 ນໂຣດມ ກຣັບຍົນ
progamming



630710037 ສູນະຮັສ ນ້ອຍເມືອງ
present