

แบบฝึกหัดคาบที่ 8: Software Design Pattern

1. ให้เลือกใช้ Design pattern ให้เหมาะสมกับสถานการณ์ต่อไปนี้

[วัดระดับน้ำท่วม]

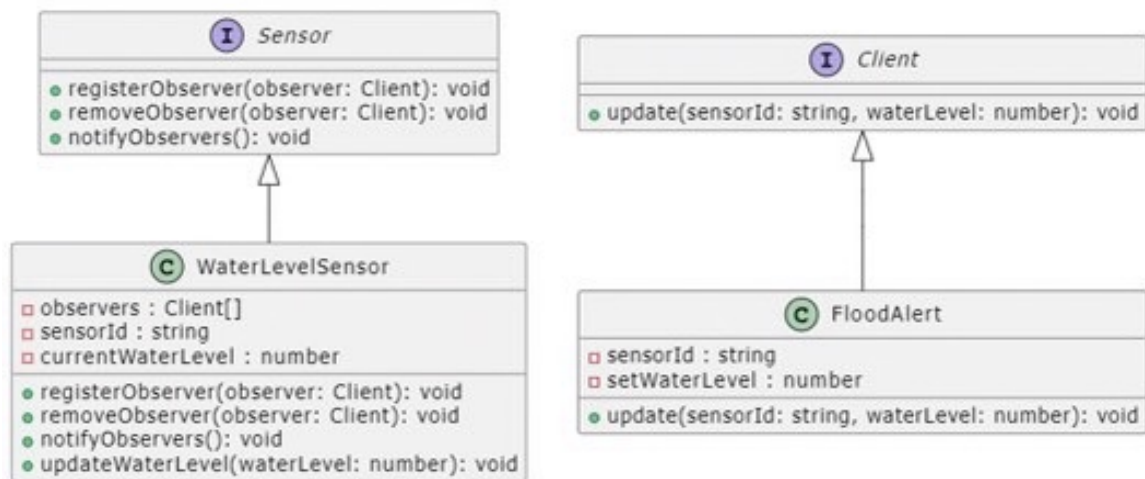
จากข่าวการเกิดเขื่อนแตกที่ประเทศเพื่อนบ้านเมื่อปีที่ผ่านมา ทางหน่วยงานที่เกี่ยวข้องมีความสนใจในการพัฒนาเครื่องมือและโปรแกรมสำหรับการแก้ปัญหาดังกล่าว จึงได้พัฒนาระบบตรวจวัดระดับน้ำของเขื่อนขึ้นทั่วประเทศ

โดยส่วนประกอบภายในระบบประกอบด้วยอุปกรณ์เซ็นเซอร์(Sensor)ที่มีโปรแกรมภายในสำหรับรับระดับน้ำ โดยเซ็นเซอร์แต่ละตัวจะมี รหัสประจำอุปกรณ์ ตำแหน่งที่ตั้งที่อุปกรณ์ติดตั้ง โดยหน้าที่ของอุปกรณ์คือรับข้อมูลของระดับน้ำทุก ๆ 10 วินาที โดยในทุก 1 นาทีจะส่งข้อมูลดังกล่าวออกไปให้ฐานข้อมูลที่เกี่ยวข้องต่อไป

เมื่อมีส่วนที่ต้องการจะรับข้อมูล(Client) สำหรับทำการแจ้งเตือนระดับน้ำและรับข้อมูลจากเซ็นเซอร์ดังกล่าวจำเป็นต้องมีการลงทะเบียนกับอุปกรณ์เซ็นเซอร์ไว้ก่อน โดยอุปกรณ์เซ็นเซอร์จะส่งข้อมูลให้กับส่วนที่ลงทะเบียนทุก 1 นาที โดยหากมีระดับน้ำที่เกินกว่าระดับน้ำที่ตั้งไว้จะส่งสัญญาณตัวอักษรเป็นการแจ้งเตือนออกมาด้วย

จากข้อกำหนดดังกล่าวให้นักศึกษาออกแบบคลาส แอทริบิวต์ และเมทอด และความสัมพันธ์ระหว่างคลาสดังกล่าวเพื่อจำลองการทำงานของระบบ โดยเขียนรายละเอียดการทำงานของเมทอดตามความเหมาะสม

Ans : Observer pattern



2. จากข้อกำหนดต่อไปนี้จงจับคู่ Design pattern ที่เหมาะสมกับข้อกำหนดดังกล่าว (บาง pattern ไม่ได้อธิบายในห้องให้นักศึกษาถูกละเอียดเพิ่มเติมเพื่อตอบคำถาม)

Abstract factory	1. แนวคิดที่ว่า Client ไม่จำเป็นต้องติดต่อกับ concrete Factory เพื่อสร้าง Object แต่ให้ติดต่อผ่าน abstract Factory แทน ซึ่งจะช่วยลด Dependency ระหว่าง Client และ Concrete Factory และจะส่งผลเราสามารถเพิ่มหรือเปลี่ยนแปลง Factory ได้ในภายหลัง โดยที่ไม่กระทบกับการทำงานของ client	A. Observer B. Mediator C. Iterator D. Facade E. Bridge F. Singleton
Singleton	2.เป็น design pattern ที่ใช้จำกัดจำนวน object ที่ถูกสร้าง ในขณะที่โปรแกรมทำงาน มีประโยชน์ในสถานการณ์ที่ทั้งระบบต้องมี object เพียงตัวเดียวเพื่อจะได้ไม่เกิดการทำงานซ้ำซ้อนกันเช่น class สำหรับ setting ค่าต่าง ๆ	G. Abstract factory H. Decorator I. Command
Observer	3.ตอนที่จำเป็นต้องกระจายข้อมูลเยอะให้กับผู้รับที่มีจำนวนเยอะมากๆ (ONE TO MANY) อาทิ และไม่ยากให้ Server รับภาระให้การ Push ข้อมูลไปที่ Client เมื่อมีการเปลี่ยนแปลง	
facade	4.เป็นดีไซน์แพตเทิร์นเพื่อช่วยให้การเรียกใช้งานคลาสต่างๆ ในระบบที่ซับซ้อนทำได้ง่ายขึ้น ได้ชื่อมาจากส่วนตกแต่งด้านหน้าของอาคาร ที่ทำเพื่อความสวยงามและซ่อนโครงสร้างของตัวตึก ตัวอย่างที่เห็นได้ในชีวิตประจำวันเช่น เกียร์รถยนต์แบบอัตโนมัติที่ผู้ใช้ไม่จำเป็นต้องกังวลกับรายละเอียดวิธีการเปลี่ยนเกียร์ เป็นต้น	
Command	5.ต้องการสร้างสื่อกลางระหว่าง Control (คนสั่ง) กับ Receiver(คนทำตาม) แต่ไม่ยากให้ Controller รู้จักกับ Receiver มากนัก สิ่งที่ต้องทำ คือ เพิ่มสื่อตรงกลางขึ้นมา ซึ่งมัน คือ คำสั่ง (Command)	