

Generalization Abstraction คือ กระบวนการในการนำ class ที่มีลักษณะเหมือนกันหรือคล้ายกันหรือมีสมบัติบางอย่างใดอย่างหนึ่งร่วมกัน มาจัดหมวดหมู่ เป็น class เดียวกัน ในทำนองกลับกัน ถ้ามี class 1 class แล้วจำแนกออก เป็น class ย่อย ๆ เราเรียกวิธีการนี้ว่า Specialization

Visibility แบ่งออกได้เป็น 3 ประเภท ได้แก่

- Public เป็นการกำหนดสิทธิ์ในการเข้าถึง Attributes หรือ Methods ได้ทันทีโดยอิสระ แต่โดยทั่วไปแล้ว Visibility แบบ Public มักจะใช้กับ Methods มากกว่า Attributes และ สัญลักษณ์ที่ใช้แทน Visibility แบบ Public ใช้สัญลักษณ์ “+”

- Private เป็นการกำหนดสิทธิ์ในการเข้าถึง Attributes หรือ Methods แบบ ส่วนตัว คือ คลาสอื่นๆ ไม่สามารถเรียกใช้งาน อ่านค่า หรือเปลี่ยนค่า Attributes หรือ Methods ได้ แต่มีเพียงคลาสของตนเองเท่านั้นที่เรียกใช้งาน อ่านค่า เปลี่ยนค่า Attributes หรือ Methods ของ ตนเองได้ แต่โดยทั่วไปแล้ว Visibility แบบ Private มักจะใช้กับ Attributes มากกว่า Methods และ สัญลักษณ์ที่ใช้แทน Visibility แบบ Private ใช้สัญลักษณ์ “-”

- Protected เป็นการกำหนดสิทธิ์ในการเข้าถึง Attributes หรือ Methods เฉพาะ คลาสตนเองและ คลาสที่มีการสืบทอดเท่านั้น คลาสอื่นๆ จะไม่สามารถเข้าถึงได้ สัญลักษณ์ที่ใช้แทน Visibility แบบ Protected ใช้สัญลักษณ์ “#”

Association Abstraction คือ กระบวนการในการสร้างความสัมพันธ์ ระหว่าง class ต่าง ๆ ใน Problem domain ที่เราสนใจ โดยเป็นการอธิบายความสัมพันธ์ ของ class ในเชิงกิจกรรม

- ความสัมพันธ์ระหว่างอ็อบเจ็ค (Relationship) คือ ความสัมพันธ์ ระหว่างอ็อบเจ็ค 2 อ็อบเจ็คขึ้นไป เช่น ความสัมพันธ์ในลักษณะความเป็นเจ้าของ ความเป็น พ่อ แม่ ลูก เป็นต้น

- พิจารณา Class ต่างๆ ใน Class Diagram ว่ามีความสัมพันธ์แบบ Association หรือไม่ หากมีให้ใส่ สัญลักษณ์ Association กำหนดบทบาท (Role) ของความสัมพันธ์และใส่ Cardinality หรือ จำนวนสมาชิกของคลาสให้ถูกต้อง