

Lab 2 : Encapsulation ในระบบธนาคารขนาดเล็ก (Mini Banking System)

หัวข้อ: Class, Object, Encapsulation

จุดประสงค์: ให้นักศึกษาเข้าใจการซ่อนข้อมูล (private) และเข้าถึงข้อมูลผ่าน method
(getter/setter/operation)

โจทย์ (Problem Statement)

ให้ ออกรูปแบบคลาส BankAccount ซึ่งใช้แทนบัญชีธนาคารของลูกค้า 1 คน โดยมีคุณสมบัติ:

ข้อมูลที่ต้องเก็บ (ต้องเป็น private)

เลขที่บัญชี (accountNumber)

ชื่อเจ้าของบัญชี (ownerName)

ยอดเงินคงเหลือ (balance)

คำจำกัดนักศึกษา: ห้ามประกาศเป็น public field เพื่อให้เข้าใจ encapsulation

เมธอดที่ต้องมี

deposit(double amount) → ฝากเงิน

withdraw(double amount) → ถอนเงิน

getBalance() → คืนยอดเงินคงเหลือ

printInfo() → แสดงข้อมูลบัญชี

เงื่อนไขทางธุรกิจ (Business Rules)

ฝากเงินต้อง > 0

ถอนเงินต้อง > 0 และเงินในบัญชีต้องพอ

ห้ามตั้งค่า balance จากภายนอก (ตรวจสอบใน main ได้)

หมาย! นักศึกษาได้เรียนรู้ว่าการแก้ค่าของ balance ต้องผ่าน method เท่านั้น → encapsulation

Skeleton Code (ให้นักศึกษาเติมเอง)

ไฟล์ 1: BankAccount.java

```
public class BankAccount {
```

```
    public BankAccount(String accountNumber, String ownerName, double initialBalance) {  
    }
```

```
// TODO: Deposit method
```

```
// ต้องตรวจสอบว่า amount > 0
```

```

public void deposit(double amount) {
    // ...
}

// TODO: Withdraw method
// ต้องตรวจสอบว่า amount > 0 และ amount <= balance
public void withdraw(double amount) {
    // ...
}

// TODO: Getter สำหรับ balance
public double getBalance() {
    return 0; // แก้ให้คืนค่าจริง
}

// TODO: printInfo() แสดงข้อมูลบัญชี
public void printInfo() {
    // แสดงเลขบัญชี ชื่อเจ้าของ และยอดเงินคงเหลือ
}

```

ไฟล์ 2: MainBankAccount.java

```

public class MainBankAccount {

    public static void main(String[] args) {

        // TODO: สร้าง BankAccount 1 บัญชี
        // TODO: ทดสอบฝากเงิน
    }
}

```

```

// TODO: ทดสอบตอนเงิน

// TODO: แสดงข้อมูลบัญชี

// ทดสอบการเข้าถึง balance โดยตรง (ต้อง error)
// acc.balance = -9999; // ต้องแก้ไม่ได้ เพราะ encapsulation
}

}

```

คำถามท้าย Lab

- ทำไมต้องประกาศ field เป็น private? เพื่อความปลอดภัยของข้อมูล และ จำกัดการเข้าถึงทั้งตัวข้อมูล และ การแก้ไข
- ถ้า balance เป็น public จะเกิดอะไรขึ้นในระบบจริง? อาจถูกแก้ไขได้โดยง่าย
- อธิบายประโยชน์ของ encapsulation ในบริบทของธุรกิจธนาคาร เนื่องจากธนาคารเป็นธุรกิจที่ต้องการความปลอดภัยสูงอันเนื่องมากจากเรื่องเงิน การ encapsulation จะช่วยเพิ่มความปลอดภัยทางธุรกิจ ให้น่าเชื่อถือมากยิ่งขึ้น และ ปลอดภัยต่อข้อมูลของลูกค้าด้วย
- ยกตัวอย่าง class อื่นในชีวิตจริงที่ควรใช้ encapsulation แล้วอธิบายเหตุผล ตัวอย่างเช่น class Citizen ข้อมูลทั่วไปของประชาชน เพราะ ข้อมูลส่วนบุคคลในทางกฎหมายเป็นข้อมูลที่เป็นข้อมูลส่วนบุคคล ไม่สามารถให้ใครรู้ได้ ดังนั้นการมีวิธี encapsulation ช่วยให้รักษาความลับของข้อมูลเหล่านี้ได้ดียิ่งขึ้น

Learning Outcomes

นักศึกษาจะ:

- เข้าใจความหมายของ encapsulation
- เขียนคลาสที่ซ่อนข้อมูลอย่างเหมาะสม
- ใช้วิธีควบคุมค่าด้วย method (deposit / withdraw)
- เห็นประโยชน์ของ OOP ในการป้องกันข้อมูลสำคัญในระบบจริง เช่น ธนาคาร