

**กิจกรรมที่ 3 : Class Diagram**

- ธนาคารให้บริการ User เปิดบัญชี โดยมีบัญชีประเภทเดียว คือ ออมทรัพย์ โดย User 1 คนจะเปิดกี่บัญชีก็ได้ โดยข้อมูลของ User ประกอบด้วย 1) เลขประจำตัวประชาชน (สมมติ) 2) ชื่อ-นามสกุล ส่วนข้อมูลของบัญชีประกอบด้วย 1) หมายเลขบัญชี 2) เจ้าของบัญชี (instance)
- ใน 1 บัญชี สามารถขอบัตร ATM ได้ 1 บัตร เพื่อใช้กับเครื่อง ATM ได้ โดยบัตร ATM จะมีหมายเลขบัตรและข้อมูลในบัตร ได้แก่ 1) หมายเลขบัญชี 2) PIN Number
- ธนาคารจะมีหลักเกณฑ์และอัตราต่างๆ ได้แก่ 1) บัตร ATM จะมีค่าธรรมเนียมรายปี 150 บาท 2) บัตร ATM จะเบิกเงินสูงสุดต่อวันไม่เกิน 40,000 บาท
- ผู้ใช้สามารถใช้บัตรกับเครื่อง ATM โดยตู้ ATM จะมีข้อมูลคือ 1) หมายเลขตู้ 2) เงินที่มีในตู้ กำหนดให้ค่าเริ่มต้นคือ 1,000,000 บาท ตู้ ATM จะไม่มีการเก็บข้อมูลใดๆ ยกเว้นเงินในตู้ และเพื่อให้ง่ายจะไม่แยกประเภทธนบัตร
- การทำกิจกรรมกับธนาคารให้ทำผ่านเครื่อง ATM เท่านั้น กิจกรรมที่ทำกับเครื่อง ATM ได้ประกอบด้วย 1) การฝากเงิน (Deposit) 2) การถอนเงิน (Withdraw) และการโอนเงิน (Transfer) โดยการถอนเงินและการโอนเงิน จะต้องไม่เกิน อัตราเบิกเงินสูงสุดต่อวัน
- ในการทำรายการแต่ละครั้งต้องมีการบันทึกลงในบัญชี ได้แก่ 1) ประเภทรายการ (D=Deposit, W=Withdraw, TD=Transfer Deposit โอนเข้า, TW=Transfer Withdraw โอนออก) 2) จำนวนเงิน 3) หมายเลขตู้ ATM 4) กรณีโอนจะเก็บเลขบัญชีที่โอนด้วย

**คำสั่ง**

- ให้เขียน Class Diagram ที่แสดงถึงระบบข้างต้น โดยยังไม่ต้องระบุ Multiplicity และให้เขียน Class Diagram ให้ Staff ตรวจสอบการเขียนโปรแกรม
- ให้เขียนโปรแกรม ที่ทำงานตาม Class Diagram ที่เขียนขึ้น โปรแกรมจะต้องเป็นไปตามข้อกำหนดดังนี้
  - ห้ามใช้ dictionary ในการเก็บข้อมูล
  - ห้ามเก็บข้อมูลนอกคลาส และ ห้ามเก็บข้อมูลซ้ำซ้อน และ ทุกข้อมูลต้องเป็น private
  - ในการสร้างคลาสให้กำหนดว่าจะเก็บข้อมูลใด และ ห้ามมิให้เก็บข้อมูลที่ไม่ใช่ข้อมูลที่เกี่ยวข้องกับคลาสนั้น หากไม่สามารถเก็บลงในคลาสใดได้เลย ให้พิจารณาสร้างคลาสใหม่ และ ให้พิจารณาให้ชื่อว่า object ใด ควรมี **พฤติกรรม** ใด
  - ในคลาสไม่ให้มีการ Input ค่าหรือ print ค่าโดยตรง ให้ส่งข้อมูล parameter เข้าไปและได้ข้อมูลกลับมาเท่านั้น (ให้มอง class เป็น service)
  - ข้อมูลที่เก็บในคลาสที่ไม่ใช่คลาสพื้นฐาน จะต้องเก็บข้อมูลเป็น Instance ของคลาสพื้นฐานเท่านั้น และ function ที่เกี่ยวข้องกับข้อมูลใน class ต้องเป็น method เท่านั้น
  - ข้อมูลที่จะเก็บสู่ Class ต้องมี Validation
  - ให้กำหนด Setter/Getter เฉพาะที่มีการอ้างถึงจากภายนอก คือ เมื่อมีการอ้างถึงค่อยมาใส่

3. ในการสร้าง instance ให้มีข้อมูลของ

- User จำนวน 2 คน กำหนดให้ Citizen ID เป็น 1-1101-12345-nn-0 โดย nn เป็นรหัส 2 ตัวสุดท้ายของรหัส นศ. สำหรับคนที่ 2 ให้ใช้ nn+1
- บัญชี จำนวน 1 บัญชีต่อคน ให้เลขบัญชี 10 หลัก
- บัตร ATM 1 บัตรต่อ 1 บัญชี สำหรับ Pin ให้ใช้เป็น 1234
- ตู้ ATM จำนวน 2 ตู้

4. Test Case จะมี 7 Test Case

- Test Case #1 : สอดบัตร เข้าตู้ ATM
- Test Case #2 : ฝากเงิน
- Test Case #3 : ฝากเงิน แต่ส่งค่าติดลบ
- Test Case #4 : ถอนเงิน
- Test Case #5 : ถอนเงิน มากกว่าเงินในบัญชี
- Test Case #6 : โอนเงิน
- Test Case #7 : แสดง statement
- Test Case #8 : ทดสอบการใส่ PIN ไม่ถูกต้อง
- Test Case #9 : ทดสอบการถอนเงินเกินวงเงินต่อวัน (40,000 บาท)
- Test Case #10 : ทดสอบการถอนเงินเมื่อเงินในตู้ ATM ไม่พอ

**การตรวจ**

1. เมื่อเขียน Class Diagram เสร็จ ให้ Staff ตรวจ
2. นำโปรแกรมมาเขียนให้ทำงานครบตาม Test Case แล้ว ให้ Staff ตรวจอีกครั้ง และ แจ้งด้วยว่ามีการแก้ไข Class Diagram หรือไม่

**คำแนะนำ:**

- ควรเริ่มจากการวาด class diagram ให้ครบถ้วนก่อนเริ่มเขียนโค้ด
- ให้ความสำคัญกับการ validate ข้อมูลในทุก method
- ระวังเรื่อง data encapsulation โดยใช้ private attributes
- ควรเพิ่ม type hints เพื่อความชัดเจนของโค้ด