

### ปัญหา บัญชีธนาคาร 3 [BankAccount3]

หลังจากมีคลาส Account สำหรับบัญชีเงินฝากแล้ว เราต้องการสร้างคลาสสำหรับบัญชีเพิ่มเติมอีก 2 ประเภท คือ 1) บัญชีฝากประจำ (FixedDeposit) เป็นบัญชีที่ต้องฝากเงินด้วยยอดที่ไม่ต่ำกว่ายอด fixAmount ที่ระบุไว้ในตอนเปิดบัญชี และ 2) บัญชีกระแสรายวัน (CurrentAccount) เป็นบัญชีที่สามารถถอนเงินเกินยอดที่มีเหลือในบัญชีได้ แต่ต้องไม่เกินวงเกิน overDraft ที่ระบุไว้ในตอนเปิดบัญชี ทั้งสองบัญชีนี้จะสืบทอดจากคลาส Account ที่สร้างไว้แล้ว โดยแต่ละคลาสจะเก็บข้อมูลเพิ่มเติม และ override เมธอด ดังนี้

คลาส FixedDeposit มีลักษณะเพิ่มเติม คือ

1. เก็บข้อมูลเพิ่ม คือ ยอดเงินฝากประจำ (fixedAmount)
2. ตัวสร้างรับข้อมูลเพิ่มจาก Account 1 ค่า เป็น double เพื่อนำค่าไปเก็บในตัวแปร fixedAmount และพิมพ์ "FixedDeposit account with " ตามด้วยค่า fixedAmount
3. Override เมธอด deposit โดยยอดเงินฝากจะต้องมากกว่า fixedAmount จึงจะฝากเงินสำเร็จ (ต้องใช้เมธอดในคลาสแม่) และเมธอด return ค่า true ถ้าไม่สำเร็จ จะพิมพ์ "deposit less than " ตามด้วยค่า fixedAmount และ return false

คลาส CurrentAccount มีลักษณะเพิ่มเติม

1. เก็บข้อมูลเพิ่ม คือ วงเงินที่ถอนเกินได้ (overDraft)
2. ตัวสร้างรับข้อมูลเพิ่มจาก Account 1 ค่า เป็น double เพื่อนำค่าไปเก็บในตัวแปร overDraft และพิมพ์ "Current account with overdraft " ตามด้วยค่า overDraft
3. Override เมธอด withdraw โดยยอดเงินที่ถอนได้จะติดลบได้ไม่เกินยอด overDraft และสถานะบัญชี active จึงจะถอนเงินสำเร็จ พร้อมทั้งพิมพ์ยอดเงินคงเหลือ ถ้าเป็นค่าติดลบจะตามด้วยข้อความว่า "overdraft " และเมธอดคืนค่า true ถ้าไม่สำเร็จจะคืนค่า false

จงสร้างคลาสทั้งสองตามข้อกำหนดข้างต้น และสร้างคลาส BankAccount3 ให้สมบูรณ์ โดยคลาส BankAccount3 จะมีเมธอด main ที่อ่านข้อมูลสำหรับสร้างสร้างบัญชี โดยสร้างบัญชีจากคลาสลูกทั้ง 3 ประเภท และแสดงผลการทำงานหรือพิมพ์ค่าที่คืนจากเมธอดดังกล่าวที่ได้ให้

## รูปแบบข้อมูลเข้า

รับข้อมูลเปิดบัญชีในลักษณะเดียวกัน 3 ครั้งดังนี้

บรรทัดแรก	เป็นข้อมูลชื่อผู้เปิดบัญชี
บรรทัดที่สอง	เป็นเลขบัญชี
บรรทัดที่สาม	เป็นยอดเงินฝากเริ่มต้น
บรรทัดที่ 4	เป็นยอดเงิน fixedAmount กรณีที่เป็นบัญชี fixedDeposit เป็นยอดเงิน overDraft กรณีที่เป็นบัญชี CurrentAccount
บรรทัดถัดจากข้อมูลข้างบน	ค่า N เป็นจำนวนครั้งที่ทำรายการ
อีก N บรรทัด	ประเภทบัญชี (1-Saving, 2-FixedDeposit, 3-CurrentAccount) ตามด้วยรายการที่ทำกับบัญชี เหมือนข้อ BankAccount1

## รูปแบบผลลัพธ์

N บรรทัด	เป็นผลจากการสร้างบัญชี และเรียกใช้เมธอดที่ตรงกับรายการที่ระบุในโค้ดที่ให้
----------	---

**ตัวอย่าง** (การเว้นบรรทัดเพื่อให้ดูผลการแสดงที่ตรงกับข้อมูลเข้าได้ง่ายเท่านั้น)

ข้อมูลเข้า	ผลลัพธ์	ความหมาย
Harry Potter 001-1200 500	Open account Saving account	เปิดบัญชี Saving เงินเปิดบัญชี 500 บาท
Hermione Granger 002-1100 500 1000.0	Open account FixedDeposit account with 1000.0	เปิดบัญชี FixedDeposit เงินเปิดบัญชี 500 บาท ฝากทุกครั้งที่ต้อง >= 1000 บาท
Ron Weasley 003-1111 500 1500.00	Open account Current account with overdraft 1500.0	เปิดบัญชี CurrentAccount เงินเปิดบัญชี 500 บาท วงเงินเกินบัญชี 1500 บาท
7		มี 7 รายการ
1 1	Harry Potter, 001-1200, 500.0, active no PromptPay	บัญชี Saving แสดงข้อมูลบัญชี
2 4 500.00	deposit less than 1000.0 false	บัญชี FixedAccount ฝาก 500 บาท ไม่สำเร็จ
2 4 1000.00	1500.0 true	บัญชี FixedAccount ฝาก 500 บาท สำเร็จ
2 1	Hermione Granger, 002-1100, 500.0, active	บัญชี FixedAccount แสดงข้อมูลบัญชี
3 5 500	0.0 true	บัญชี CurrentAccount ถอน 500 บาท สำเร็จ
3 5 1500	-1500.0 overdraft true	บัญชี CurrentAccount ถอน 1500 บาท สำเร็จ
3 1	Ron Weasley, 003-1111, -1500.0, active	บัญชี CurrentAccount แสดงข้อมูลบัญชี

## โครงสร้างโค้ดและเกณฑ์การให้คะแนน

```
import java.util.Scanner;
class FixedDeposit // your code here {
    // Enter Your Code Here.
}
class CurrentAccount // your code here {
    // Enter Your Code Here.
}
public class BankAccount3 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Saving sAcc = null;
        FixedDeposit fAcc = null;
        CurrentAccount cAcc = null;
        for (int t = 1; t <= 3; t++) {
            String name = scan.nextLine();
            String code = scan.nextLine();
            double money1 = scan.nextDouble();
            if (t == 1) {
                scan.nextLine();
                sAcc = new Saving(name, code, money1);
            } else if (t == 2) {
                double money2 = scan.nextDouble();
                scan.nextLine(); //Skip the remainder of the double line.
                fAcc = new FixedDeposit(name, code, money1, money2);
            } else if (t == 3) {
                double money2 = scan.nextDouble();
                scan.nextLine(); // Skip the remainder of the double line.
                cAcc = new CurrentAccount(name, code, money1, money2);
            }
        }
        int N = scan.nextInt();
        for (int i = 0; i < N; ++i) {
            double amount;
            //your code to scan input
            switch(menu) {
                case 1 -> {
                    //your code
                }
                //your code case 2-4
            }
            case 5 -> {
                //your code
            }
        }
    }
}
```