คลาส `ArrayStack` และ `ArrayListStack` ที่เรียนมาเป็นตัวอย่างการสร้างคลาสสำหรับโครงสร้างข้อมูลพื้นฐานที่คล้ายกับคลาส `Stack` ในจาวา แบบฝึกหัดนี้เป็นการฝึก

- เขียน method เพิ่มในคลาส `ArrayStack` และ `ArrayListStack`
- สร้างคลาสที่ทำงานเหมือน stack โดยใช้โครงสร้างข้อมูลแบบอื่น และ
- ใช้ stack ในการแก้ปัญหา

แบบฝึกหัดบางส่วนมาจาก “โครงสร้างข้อมูล : ฉบับวาจาจาวา” (https://www.cp.eng.chula.ac.th/books/ds-vjjv/) โดย รศ.ดร.สมชาย ประสิทธิ์จูตระกูล

เขียน method เพิ่มในคลาส `ArrayStack` และ `ArrayListStack`

เขียน method ต่อไปนี้เพิ่มสำหรับคลาส `ArrayStack` และ `ArrayListStack`

- `Constructor` ที่รับพารามิเตอร์เป็น `ArrayStack` (หรือ `ArrayListStack`) แล้วสร้าง stack ที่มีค่าใน stack เหมือนกัน
- `Object secondTop()` ที่คืนข้อมูลตัวถัดจากตัวบนสุดใน stack ที่เป็น implicit parameter (ไม่เปลี่ยนค่าใน stack)
- `void clear()` ที่ลบข้อมูลทั้งหมดใน stack ที่เป็น implicit parameter
- `void popTo(Stack d)` ที่ pop ข้อมูลทุกตัวจาก stack ที่เป็น implicit parameter ไป push ใส่ใน stack `d`
- `void pushFrom(Stack s)` ที่ pop ข้อมูลทุกตัวจาก stack `s` ไป push ใส่ใน stack ที่เป็น implicit parameter

และสร้าง test class ที่ใช้คลาส `ArrayStack` และ `ArrayListStack` เพื่อทดสอบ method ที่สร้างมา

สร้างคลาส `LinkedListStack`

สร้างคลาส `LinkedListStack` ที่ implement `Stack` โดยใช้คลาส `LinkedList` เพื่อเก็บข้อมูล และสร้าง test class ที่ใช้คลาส `LinkedListStack` เพื่อทดสอบคลาสที่สร้างมา
จากนั้นเปรียบเทียบว่าการใช้คลาส `LinkedList` เพื่อสร้าง stack ต่างจากการใช้ Array และ `ArrayList` อย่างไร

เขียนโปรแกรมเล่นเกม Tower of Hanoi

Tower of Hanoi เป็นเกมที่ให้ย้ายแผ่นดิสก์ทั้งหมดจากเสาหนึ่งไปยังอีกเสาหนึ่งตามกติกาที่อธิบายใน Wikipedia พร้อมทั้งวิธีแก้ปัญหานี้แบบ recursive (https://en.wikipedia.org/wiki/Tower_of_Hanoi#Recursive_solution)
เขียนโปรแกรมที่แก้ปัญหา Tower of Hanoi ตามวิธี recursive ใน Wikipedia โดยให้เสา (peg) แต่ละเสาเป็น stack ดังนั้นการย้ายดิสก์จากเสาหนึ่งไปยังอีกเสาคือการ pop ค่าจาก stack หนึ่งไป push ใส่อีก stack

คำนวณค่าของ postfix expression

เขียนโปรแกรมที่รับ postfix expression แล้วคำนวณค่าที่ได้จาก expression นั้นโดยใช้ `stack`