

คลาส `ArrayQueue` และ `ArrayListQueue` ที่เรียนมาเป็นตัวอย่างการสร้างคลาสสำหรับโครงสร้างข้อมูลพื้นฐานที่คล้ายกับ interface `Queue` ในจาวา แบบฝึกหัดนี้เป็นการฝึก

- สร้างคลาสที่ทำงานเหมือน queue โดยใช้โครงสร้างข้อมูลแบบอื่น
- สร้างคลาสที่ทำงานคล้าย queue และ
- ใช้ queue ในการแก้ปัญหา

แบบฝึกหัดบางส่วนมาจาก “โครงสร้างข้อมูล : ฉบับจาวา” (<https://www.cp.eng.chula.ac.th/books/ds-vjiv/>) โดย รศ.ดร.สมชาย ประสิทธิ์จตุระกุล

สร้างคลาส `LinkedListQueue`

สร้างคลาส `LinkedListQueue` ที่ implement `Queue` โดยใช้คลาส `LinkedList` เพื่อเก็บข้อมูล และสร้าง test class ที่ใช้คลาส `LinkedListQueue` เพื่อทดสอบคลาสที่สร้างมา

จากนั้นเปรียบเทียบว่าการใช้คลาส `LinkedList` เพื่อสร้าง queue ต่างจากการใช้ `Array` และ `ArrayList` อย่างไร

เขียนโปรแกรมเล่นเกม Sudoku ขนาด 4x4

Sudoku เป็นเกมที่ให้ผู้เล่นเติมตัวเลขในช่องว่างของกระดาน โดยในทุกแนวตั้ง ทุกแนวนอน และ จัตุรัสที่กำหนดต้องมีตัวเลขไม่ซ้ำกันดังแสดงในรูปข้างล่าง เกม Sudoku มีหลายขนาด ในที่นี้ เราจะเขียนโปรแกรมให้เล่น Sudoku ขนาด 4x4

กระดานโจทย์	<table><tr><td>2</td><td>1</td><td></td><td></td></tr><tr><td></td><td>3</td><td>2</td><td></td></tr><tr><td></td><td></td><td></td><td>4</td></tr><tr><td>1</td><td></td><td></td><td></td></tr></table>	2	1				3	2					4	1				<table><tr><td>1</td><td></td><td>3</td><td>4</td></tr><tr><td></td><td>3</td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>1</td></tr></table>	1		3	4		3			2							1	<table><tr><td>4</td><td>2</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>2</td><td></td></tr><tr><td></td><td></td><td></td><td>3</td></tr></table>	4	2			1					1	2					3	<table><tr><td></td><td></td><td></td><td>3</td></tr><tr><td>4</td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>3</td><td></td></tr><tr><td>3</td><td></td><td>2</td><td></td></tr></table>				3	4					1	3		3		2	
	2	1																																																																		
		3	2																																																																	
				4																																																																
1																																																																				
1		3	4																																																																	
	3																																																																			
2																																																																				
			1																																																																	
4	2																																																																			
1																																																																				
	1	2																																																																		
			3																																																																	
			3																																																																	
4																																																																				
	1	3																																																																		
3		2																																																																		
เฉลย	<table><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr><tr><td>4</td><td>3</td><td>2</td><td>1</td></tr><tr><td>3</td><td>2</td><td>1</td><td>4</td></tr><tr><td>1</td><td>4</td><td>3</td><td>2</td></tr></table>	2	1	4	3	4	3	2	1	3	2	1	4	1	4	3	2	<table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>4</td><td>3</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr><tr><td>3</td><td>4</td><td>2</td><td>1</td></tr></table>	1	2	3	4	4	3	1	2	2	1	4	3	3	4	2	1	<table><tr><td>4</td><td>2</td><td>3</td><td>1</td></tr><tr><td>1</td><td>3</td><td>4</td><td>2</td></tr><tr><td>3</td><td>1</td><td>2</td><td>4</td></tr><tr><td>2</td><td>4</td><td>1</td><td>3</td></tr></table>	4	2	3	1	1	3	4	2	3	1	2	4	2	4	1	3	<table><tr><td>1</td><td>2</td><td>4</td><td>3</td></tr><tr><td>4</td><td>3</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>3</td><td>4</td></tr><tr><td>3</td><td>4</td><td>2</td><td>1</td></tr></table>	1	2	4	3	4	3	1	2	2	1	3	4	3	4	2	1
	2	1	4	3																																																																
	4	3	2	1																																																																
	3	2	1	4																																																																
1	4	3	2																																																																	
1	2	3	4																																																																	
4	3	1	2																																																																	
2	1	4	3																																																																	
3	4	2	1																																																																	
4	2	3	1																																																																	
1	3	4	2																																																																	
3	1	2	4																																																																	
2	4	1	3																																																																	
1	2	4	3																																																																	
4	3	1	2																																																																	
2	1	3	4																																																																	
3	4	2	1																																																																	

(มีกระดานโจทย์และเฉลยของเกม Sudoku ขนาด 4x4 ที่ http://www.sudoku-download.net/sudoku_4x4.php)

- เขียนโปรแกรมที่แก้ปัญหา Sudoku โดยวิธี breadth-first search คือ ลองใส่เลขทีละช่อง แล้วเอากระดานที่ไม่ขัดกับข้อกำหนดของเกม (คือ ใส่เลขที่ต้องการในช่องที่เราสนใจได้) ใส่ Queue แล้วดึงกระดานที่หัว Queue มาลองใส่เลขในช่องถัดไปเรื่อยๆ จนใส่เลขในกระดานครบทุกช่อง
- ลองเขียนโปรแกรมที่แก้ปัญหานี้โดยวิธี depth-first search คือ ลองใส่เลขทีละช่อง แล้วเอากระดานที่ไม่ขัดกับข้อกำหนดของเกม (คือ ใส่เลขที่ต้องการในช่องที่เราสนใจได้) ใส่ Stack แล้วดึงกระดานจาก Stack (บนสุดก่อน) มาลองใส่เลขในช่องถัดไปเรื่อยๆ จนใส่เลขในกระดานครบทุกช่อง

สร้างคลาส `Deque`

สร้างคลาส `Deque` หรือ Double-ended Queue ที่ให้เพิ่มและลบข้อมูลได้ที่ปลายทั้ง 2 ข้างของ queue และมี method เหมือน interface `Deque<E>` ในจาวา (<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/Deque.html>) และสร้าง test class ที่ใช้คลาส `Deque` เพื่อทดสอบคลาสที่สร้างมา