



แบบฝึกปฏิบัติการครั้งที่ 11

Generic Programming

จุดประสงค์ เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถ

1. เข้าใจหลักการของ Generic Programming
2. ใช้คำสั่งภาษาจาวา เพื่อสร้าง Method Template ได้
3. ใช้คำสั่งภาษาจาวา เพื่อสร้าง Class Template ได้

คำสั่ง

จงเขียนโปรแกรมเชิงวัตถุ โดยให้สร้าง Project ใหม่สำหรับแต่ละข้อ (1 Project ต่อ 1 ข้อ) ตามชื่อ Project ที่กำหนดให้ แล้วทำการบีบอัดไฟล์ 1 ไฟล์ต่อ 1 Project (บีบอัดจาก folder ของ Project) โดยให้ไฟล์ที่บีบอัดมีชื่อเดียวกับ Project แต่มีนามสกุลไฟล์เป็น .rar หรือ .zip ก่อนทำการ Upload ส่ง

การส่งงาน

- เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์
- ให้นักศึกษาส่งงานด้วยการ Upload ผ่านหน้าเว็บไซต์สำหรับส่งงาน <http://hw.cs.science.cmu.ac.th>
- ส่งงานโดยการ Upload ส่งไฟล์ที่บีบอัดแล้ว คือ Lab11_1_รหัสนักศึกษา.rar และ Lab11_2_รหัสนักศึกษา.rar ภายในวันพฤหัสบดี ที่ 30 กันยายน 2564

1. (ข้อนี้ไม่ต้องสร้างเป็น class ให้สร้างเป็น method template แล้วเรียกใช้จาก main เท่านั้น ดังเช่น ตัวอย่างโปรแกรมที่ 3-5 ของ lecture เรื่อง Generic Class) **(ตั้งชื่อ Project ว่า Lab11_1_รหัสนักศึกษา)**

จงสร้างแผ่นแบบเมทอด(Method Template) ชื่อ `getCenter()` ที่รับพารามิเตอร์ เป็นข้อมูลที่มีชนิดเดียวกัน 3 จำนวน แต่จะเป็นชนิดอะไรก็ได้ โดยเมทอดนี้จะทำการหาผลลัพธ์คือค่าที่อยู่ตรงกลางระหว่างข้อมูล 3 จำนวนนั้น แล้วส่งค่ากลับ (return) มาพิมพ์ใน `main()` เช่น 3 8 1 ค่ากลางคือ 3

9 9 6 ค่ากลางคือ 9

จากนั้นทำการเรียกใช้เมทอดดังกล่าว 3 ครั้ง ด้วยข้อมูลชนิด `int`, `char` และ `float` ตามลำดับ เช่น

```
getCenter(3,8,1)
```

```
getCenter('x','a','c')
```

```
getCenter(1.8,2.5,4.9)
```

ตัวอย่าง Input

Enter 3 integer numbers

3

8

1

Enter 3 characters

x

a

c

Enter 3 float numbers

1.8

2.5

4.9

ตัวอย่าง Output

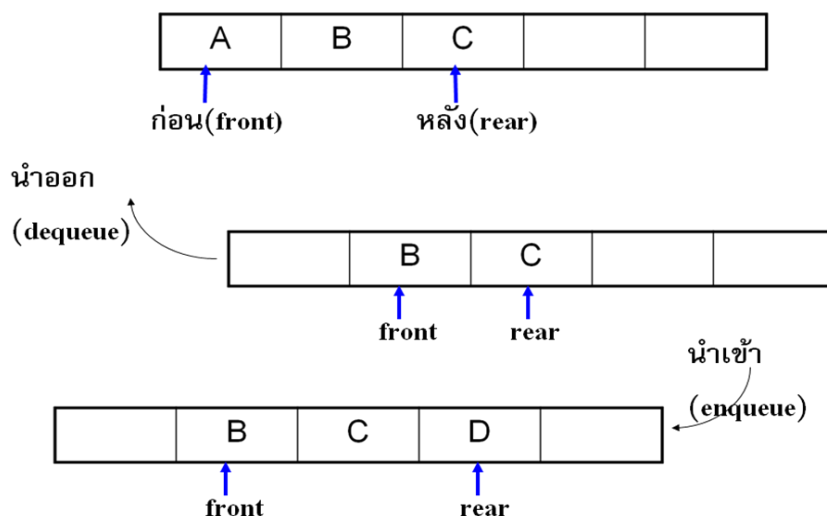
3

c

2.5

2. จงสร้างแผ่นแบบคลาส(Class Template) ชื่อ Queue เพื่อสร้างและดำเนินการกับข้อมูลในคิว ชนิด Linear แบบ Simple ดังภาพ (ตั้งชื่อ Project ว่า Lab11_2_รหัสนักศึกษา)

- การใส่ข้อมูล จะใส่ลงด้านหลัง (ต่อจากตำแหน่ง rear)
- การนำข้อมูลออก จะนำข้อมูลด้านหน้า ในตำแหน่ง front ออกจากคิว
- กรณีคิวว่าง count=0
- กรณีคิวเต็ม rear=N-1 เมื่อ N คือขนาดของคิว



กำหนดให้ member data ประกอบด้วย

- count มีชนิดเป็น int สำหรับเก็บว่าปัจจุบันมีข้อมูลอยู่ในคิวกี่จำนวน
- front มีชนิดเป็น int สำหรับเก็บ index ของข้อมูลที่อยู่หน้า
- rear มีชนิดเป็น int สำหรับเก็บ index ของข้อมูลที่อยู่ด้านหลัง
- item[] มีชนิดเป็น array of อะไรก็ได้ สำหรับเก็บข้อมูลในคิว

กำหนดให้ member method ประกอบด้วย

- isEmpty() สำหรับตรวจสอบว่าคิวว่างหรือไม่ โดยส่งค่ากลับเป็นจริงหรือเท็จ
- isFull() สำหรับตรวจสอบว่าคิวเต็มหรือไม่ โดยส่งค่ากลับเป็นจริงหรือเท็จ
- enqueue(v) สำหรับนำข้อมูล v ใส่องในด้านหลังของคิวหากคิวยังไม่เต็ม (ตรวจสอบโดยเรียกใช้ isFull()) พร้อมทั้งปรับค่า count แล้วแสดงข้อความ “OK” มิฉะนั้นแล้วแสดงข้อความ “Queue is full.”
- dequeue() สำหรับนำข้อมูลส่วนหัวออกจากคิว โดยหากคิวไม่ว่าง (ตรวจสอบโดยเรียกใช้ isEmpty()) ให้นำข้อมูลออกจากคิว พร้อมทั้งปรับค่า count แล้วแสดงข้อมูลนั้นออกทางหน้าจอ มิฉะนั้นแล้วแสดงข้อความ “Queue is empty.”
- show() สำหรับแสดงข้อมูลทุกตัวในคิว

ให้นักศึกษาสร้างแผ่นแบบคลาสที่ใช้ TYPE เป็น ชนิดข้อมูลใน array ซึ่งมีขนาด N และ โปรแกรมสามารถ

- รับชนิดข้อมูลว่าเป็น int หรือ char แล้วสร้างวัตถุของคลาสที่สอดคล้องตามชนิดข้อมูล
- แสดงเมนู ต่อไปนี้ จนกว่าผู้ใช้จะเลือก 4 จึงจะจบการทำงานของโปรแกรม
 - 1) enqueue
 - 2) dequeue
 - 3) show
 - 4) exit

เมื่อผู้ใช้เลือกเมนูหมายเลขใด ให้ทำการรับ input และ/หรือ แสดง output ที่สอดคล้องกับเมนูและชนิดข้อมูลที่ผู้ใช้เลือก

Hint อัลกอริทึมบางส่วนสำหรับการดำเนินการกับ Queue

ให้นักศึกษานำไปปรับใช้ให้สอดคล้องกับภาษาจาวาตามที่โจทย์กำหนด

Queue(int x) { //สร้างคิว

```
    N = x;
    item = new Object[N];
    front=0
    rear= -1
    count=0
}
```

public void enqueue(TYPE v) { //เพิ่มข้อมูลเข้าสู่คิว

```
    if (rear < N-1) {
        rear++
        item[rear] = v
        count++
    }
}
```

TYPE dequeue() { //นำข้อมูลออกจากคิวแล้วส่งค่ากลับเป็นค่านั้น

```
    if (!q.isEmpty())
        queueFront=item[front]
        front++
        count--
        return queueFront
}
```