TCP1201 Object-Oriented Programming and Data Structures

Lab09-10 Generics, Implementing Stacks and Queues

Exercise 1: Generic Method with Comparable

1) Write a program that asks user input for 5 integers, then defines and uses the following method to check whether the elements in the array are in ascending order.

```
public static boolean isAscending (int[] array)
```

2) Update your program to add asking user input for 5 strings, then update the `isAscending` method to a generic version so that it can also check whether the elements in the string array are in ascending order.

Sample Run 1:
```
Enter 5 integers: 1 2 4 6 8
[1, 2, 4, 6, 8] is IN ascending order.

Enter 5 strings: a d f h j
[a, d, f, h, j] is IN ascending order.
```

Sample Run 2:
```
Enter 5 integers: 2 4 6 8 1
[2, 4, 6, 8, 1] is NOT in ascending order.

Enter 5 strings: a s d f h
[a, s, d, f, h] is NOT in ascending order.
```

Exercise 2: Implementing Stack Using Composition

Without copy and paste the code provided by the lecture, define a generic stack named **GenericStackComposition** that uses an appropriate list via composition.

Write a test program to test your generic stack.

Sample run:
```
GenericStack: []
Size: 0
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 1

GenericStack: [56]
Size: 1
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
```

```
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 1

GenericStack: [56, 20]
Size: 2
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 1

GenericStack: [56, 20, 31]
Size: 3
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 2
31

GenericStack: [56, 20, 31]
Size: 3
1 - Add a random integer to stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 3
31

GenericStack: [56, 20]
Size: 2
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 4
false

GenericStack: [56, 20]
Size: 2
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
```

```
5 - Clear stack
0 - Exit
Command > 5

GenericStack: []
Size: 0
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 4
true

GenericStack: []
Size: 0
1 - Push a random integer into stack
2 - Peek stack
3 - Pop stack
4 - Is stack empty?
5 - Clear stack
0 - Exit
Command > 0
```

Exercise 3: Implementing Stack Using Inheritance

Define a new generic stack class named **GenericStackInheritance** that extends an appropriate list.

Test your new GenericStackInheritance class with your test program in Exercise 2.

Exercise 4: Implementing Queue Using Composition

Without copy and paste the code provided by the lecture, define a generic queue named GenericQueueComposition that uses an appropriate list via composition. Write a test program to test your generic queue.

Sample run:

```
GenericQueue: []
Size: 0
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 1

GenericQueue: [70]
Size: 1
1 - Enqueue a random integer into queue
2 - Dequeue queue
```

```
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 3
false

GenericQueue: [70]
Size: 1
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 1

GenericQueue: [70, 43]
Size: 2
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 1

GenericQueue: [70, 43, 86]
Size: 3
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 2
70

GenericQueue: [43, 86]
Size: 2
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 4

GenericQueue: []
Size: 0
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 3
true

GenericQueue: []
```

```
Size: 0
1 - Enqueue a random integer into queue
2 - Dequeue queue
3 – Is queue empty?
4 - Clear queue
0 - Exit
Command > 0
```

Exercise 5: Implementing Queue Using Inheritance

Define a new generic queue class named GenericQueueInheritance that extends an appropriate list.

Test your new GenericQueueInheritance class with your test program in Exercise 4.