CPT204 – Advanced OOP
Department of Computing

Xi’an Jiaotong-Liverpool University

Tutorial/Lab 5 - Lists, Stacks, Queues, and Priority Queues

Aim

This tutorial/lab aims to further the understanding of Week 5’s lecture on the topic of lists stacks, queues, and priority queues.

Exercise 5.1 List Iterator

During the lecture, we discussed that using an iterator would be more efficient than using get() method to traverse a list. Refer to lecture page 30 of lecture 5 for detail. In this exercise you are going to test the difference.
Guideline to the task:
(1) Create a LinkedList containing 100,000 Integer objects.
(2) Create a timer using System.currentTimeMillis().
(3) Traverse through the LinkedList using the get() method. Record the time consumed.
(4) Traverse through the LinkedList using an iterator. Record the time consumed.
Output the above two time value.

Exercise 5.2 Set on Priority Queues

Write a program that creates two priority queues, {"George", "Jim", "John", "Blake", "Kevin", "Michael"} and {"George", "Katie", "Kevin", "Michelle", "Ryan"} and displays their union, difference, and intersection.
Guideline to the task:
(1) Use PriorityQueue’s addAll() method to get the union of two priority queues.
(2) Use PriorityQueue’s removeAll() method to get the difference of two priority queues.
(3) Use PriorityQueue’s retainAll() method to get the intersection of two priority queues

Exercise 5.3 Stack

A program source-code can contain various pairs of grouping symbols, such as:
- Parentheses: (and)
- Braces: { and }
- Brackets: [and]

The grouping symbols cannot overlap. For example, (a{b)} is illegal.
Write a program to check whether a program source-code line has paired grouping symbols. You may assume that the program source-code line will be input from the keyboard. Below are two sample input and output.

Input: int a = (int) (“0.1”);
Output: Paired

Input: public void static main(String[] args){
Output: Unpaird

Created Spring 2024
Yihong Wang, Erick Purwanto