Завдання 2

Dynamic Immutable Array (ArrayList) and Immutable Linked List

В архіві знаходиться шаблон проекту, який містить інтерфейс ImmutableList і класи, які необхідно реалізувати.

Інструкція:

1. Скачайте архів з шаблоном проекту *ImmutableCollections* та розархівуйте його
2. Додайте його в локальний *Git* репозиторій
    a. Перейдіть в командному рядку до директорії *ImmutableCollections*
    b. Виконайте команди:
    > git init
    *> git add **
    > git commit -m "ImmutableCollections initial commit"
3. Створіть на своєму GitHub новий репозиторій з іменем за наступним шаблоном apps20<surname>-hw2
4. Знову перейдіть в директорію *ImmutableCollections* і виконайте наступну команду
    *> git remote add origin https://github.com/**<user_name>/apps20<surname>-hw2.git***
5. Виконайте команду
    > git push -u origin master

Після цього може відкрити проект в IDE і почати виконувати завдання

Важливо:

- не змінюйте структуру шаблоного проекту та не видаляйте файли *checkstyle.xml, pom.xml*
- при коміті проекту в локальний репозиторій (та на GitHub) необхідно щоб виключно були закомічені: *src/, checkstyle.xml, pom.xml*. Директорію *target/,* яка буде створена під час компіляції комітити не треба.

Завдання:

1. Реалізуйте класи *ImmutableLinkedList* (зв'язаний список) і *ImmutableArrayList* (динамічний масив), що реалізують інтерфейс *ImmutableList*.
    Колекції є Immutable, тому не можна змінювати стану існуючої колекції. Тому всі операції які можуть змінювати стан мають повертати нову колекції.
    Тобто після виконання наступного фрагменту коду:
    ImmutableList immutableList = …;
    ImmutableList newImmutableList = immutableList.add(“abc”);
    змінна immutableList не буде містити новий доданий елемент.

    При реалізації інтерфейсу *ImmutableList* класами *ImmutableLinkedList або ImmutableArrayList*, методи інтерфейсу що повертають *ImmutableList* мають повертати колекції конкретного типу (тобто *ImmutableLinkedList або ImmutableArrayList*)

2. ImmutableLinkedList, додатково до методів з п.1 повинен мати наступні методи:

- *public* ImmutableLinkedList *addFirst(Object e) - додає елемент у початок зв'язаного списку*
- *public* ImmutableLinkedList *addLast(Object e) - додає елемент у кінець зв'язаного списку*
- *public Object getFirst()*
- *public Object getLast()*
- *public* ImmutableLinkedList *removeFirst() - видаляє перший елемент*
- *public* ImmutableLinkedList *removeLast() - видаляє останній елемент*

3. Використовуючи ImmutableLinkedList реалізуйте класи Queue та Stack - дані класи не мають бути Immutable (!!!)

Queue:
- Object peek() - Returns the object at the beginning of the Queue without removing it
- Object dequeue() - Removes and returns the object at the beginning of the Queue.
- void enqueue(Object e) - Adds an object to the end of the Queue.

Stack:
- Object peek() - Returns the object from the top of the Stack without removing it
- Object pop() - Removes and returns the object from the top of the Stack
- void push(Object e) - Adds an object to the the top of the Stack

4. Необхідно написати модульні тести
5. Аналогічно до попереднього завдання має бути створена нова Build Job на системі **CI Hudson/Jenkins** з іменем ***apps20<surname>-hw2*** та виконані вимоги до якості коду та його покриття тестами