Αναφορά Παράδοσης

Εργασία: Δομές Δεδομένων - Εργασία 1

Από:

Βλάχος Νίκος p3140306

Σπύρος Διαμαντής p3180048

Πίνακας περιεχομένων

α. Συνοπτική επεξήγηση για “Υλοποίηση των διεπαφών στο μέρος Α”.

Το πρόγραμμα merosA.java είναι η main (προαιρετικό αρχείο) που τρέχει δοκιμαστικά τις κλάσεις IntQueueImpl.java και StringStackImpl.java οι οποίες είναι οι υλοποιήσεις των διεπαφών που δίνονται για το μέρος Α, IntQueue.java και StringStack.java αντίστοιχα.

Η υλοποίηση περιλαμβάνει **τροποποιήσεις στους αρχικούς κώδικες IntQueue.java και StringStack.java** ώστε να είναι δυνατή η **υλοποίηση με generics**, όπως ζητείται στο bonus υποερώτημα του μέρους Α. Έτσι οι κώδικες φαίνονται ως εξής:

```java
package merosA;
import java.io.PrintStream;


public interface IntQueue<T> {

    /**
     * @return true if the queue is empty
     */
    public boolean isEmpty();

    /**
     * insert an integer to the queue
     */
    public void put(T item);

    /**
     * remove and return the oldest item of the queue
     * @return oldest item of the queue
     * @throws NoSuchElementException if the queue is empty
     */
    public T get() throws NoSuchElementException;

    /**
     * return without removing the oldest item of the queue
     * @return oldest item of the queue
     * @throws NoSuchElementException if the queue is empty
     */
    public T peek() throws NoSuchElementException;


```

```java

import java.io.PrintStream;

/**
 * Defines the methods for a Stack that handles String items
 */
public interface StringStack<T> {

        /**
     * @return true if the stack is empty
     */
    public boolean isEmpty();

    /**
     * insert a String item to the stack
     */
    public void push(T item);

    /**
     * remove and return the item on the top of the stack
     * @return the item on the top of the stack
     * @throws a NoSuchElementException if the stack is empty
     */
    public T pop() throws NoSuchElementException;

        /**
     * return without removing the item on the top of the stack
     * @return the item on the top of the stack
     * @throws a NoSuchElementException if the stack is empty
     */
    public T peek() throws NoSuchElementException;
```

β. Συνοπτική επεξήγηση για τον τρόπο χρήσης του μέρους Α στο μέρος Β.

Το μέρος Β υλοποιείται στο TagMatching.java στο οποίο υπάρχει δική του static main συνάρτηση για να τρέχει ξεχωριστά, και πρέπει να του περάσουμε running parameter το αρχείο html που περιέχει τα tags για τα οποία θα κάνουμε tag matching.

Η συνάρτηση σε αυτό το αρχείο με το όνομα isHTMLMatched που δέχεται ως όρισμα ένα array από Tag στοιχεία, αρχικοποιεί ένα **StringStackImpl<String> S**. Αυτό είναι η υλοποίηση της στοίβας του μέρους Α, στο οποίο κάνουμε push αν ισχύει πως το tag είναι isOpening και pop αν όχι, συγκρίνοντας το με την αντίστοιχη θέση στο tag array.

Αν η λίστα έχει αδειάσει τότε επιστρέφουμε true. Άν όχι τότε δεν μπορεί να θεωρηθεί matching το έγγραφο.

γ. Συνοπτική επεξήγηση για τον τρόπο χρήσης του μέρους Α στο μέρος Γ.

Το μερος Γ υλοποιείται στο NetBenefit.java, όπως και στο μέρος Β χρειαζόμαστε command argument το αρχείο από το οποίο θα διαβάσουμε τις πληροφορίες αγοράς/πώλησης μετοχών, όπως ζητείται στο ερώτημα.

Έπειτα ορίζουμε ένα IntQueueImpl<Integer> iq. Το οποίο θα λειτουργήσει ως το FIFO queue μας. Στην περίπτωση που συναντάμε στο αρχείο εντολή αγοράς, κάνουμε put την τιμή στην στοίβα, quantity φορές. Αλλιώς αν είναι sell, κάνουμε κατευθείαν πρόσθεση στο netBenefit το οποίο έχει ήδη αρχικοποιηθεί ως 0, την εξής αριθμητική πράξη

netBenefit += (price - (int) iq.get());

Έτσι μέσω του get λαμβάνουμε το πρώτο buy και το αφαιρούμε από την τιμή πώλησης, έχοντας έτσι το καθαρό κέρδος κάθε ξεχωριστού γεγονότος πώλησης.

Στο τέλος κάθε πώλησης εμφανίζουμε μήνυμα.

```
Total net benefit after sell 110 price 30: 510
Total net benefit after sell 30 price 40: 170
```

δ. Τρόπος εκτέλεσης

Για το μέρος Α απλώς τρέχουμε το merosA.java, για το μέρος Β τρέχουμε το TagMatching.java έχοντας προηγουμένως τροποποιήσει τα arguments στο Eclipse όπως ζητείται. Και για το μέρος Γ αντίστοιχα τρέχουμε το NetBenefit.java με τις κατάλληλες αλλαγές στο Run Configurations παράθυρο του Eclipse.

ΠΡΟΣΟΧΗ, όλα τα αρχεία του source code είναι σε ένα package MerosA.