

Judith Koch - Übung 4

4.1 Das Interface Schlange

Um das vorgegebene Interface zu vervollständigen und zu erproben habe ich zwei neue Klassen geschrieben: MyQueue, welche das Interface implementiert, und QueueNode, welche im Moment zu grossen Teilen mit der bereits geschriebenen Klasse Node übereinstimmt.

```
public class MyQueue<T> implements Queue<T>{  
    private QueueNode<T> first = null;  
    private QueueNode<T> last = null;  
    //Konstruktor mit nur einem Element  
    public MyQueue(QueueNode<T> queueNode){  
        first = queueNode;  
        last = queueNode;  
    }  
    //Konstruktor mit nur einem Element  
    public MyQueue(QueueNode<T> firstQueueNode, QueueNode<T> lastQueueNode){  
        first = firstQueueNode;  
        last = lastQueueNode;  
    }  
}
```

Die Methoden habe ich entsprechend umgesetzt.

Die „falsche Schlange“, die das Stapel-Interface verwendet, ist MyQueue prinzipiell sehr ähnlich. Hier wird die Schlange nur von der anderen Seite her aufgebaut.

4.2 Klammern prüfen 2.0

Um die Klammerprüfung rekursiv umzusetzen wollte ich unbedingt die oben geschriebene Klasse MyQueue verwenden. Für die Implementierung musste ich im Nachhinein einige Anpassungen vornehmen: z.B. habe ich die Methode cancelLast() umgesetzt, in der das zuletzt hinzugefügte Element wieder gelöscht wird.

Um die Klammern zu überprüfen, lese ich sie beim Erstellen der Brackets-Instanz mit dem StringTokenizer aus und füge sie einer Schlange hinzu.

Intensive Überlegungen haben mich letztendlich zu folgender Methode gebracht:

```
public boolean checkBrackets(){  
    if(queue.getFirst() != null && queue.getFirst().contentEquals("(")){  
        queue.dequeue();  
        if(queue.getLast() != null && queue.getLast().contentEquals(")")){  
            queue.cancelLast();  
            return checkBrackets();  
        } else {  
            return false;  
        }  
    } else if (queue.isEmpty()){  
        return true;  
    } else {  
        return false;  
    }  
}
```