Übung 5: 234- und Rot-Schwarz-Bäume

Abgabetermin: 24.04.2018

Name: **Matrikelnummer:**

Gruppe: ☐ **G1** Di 10:15-11:00 ☐ ~~**G2** Di 11:00-11:45~~ ☐ **G3** Di 10:15-11:00

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	14	☐	Zeichnung nach jedem Schritt	Zeichnung nach jedem Schritt	☐	
Aufgabe 2	10	☐	Java-Programm	Projekt Archiv	☐	
Aufgabe 3 (freiwillig)	6	☐	Java-Programm	Projekt Archiv	☐	

Aufgabe 1: 234-Baum & Rot-Schwarz-Baum (14 Punkte)

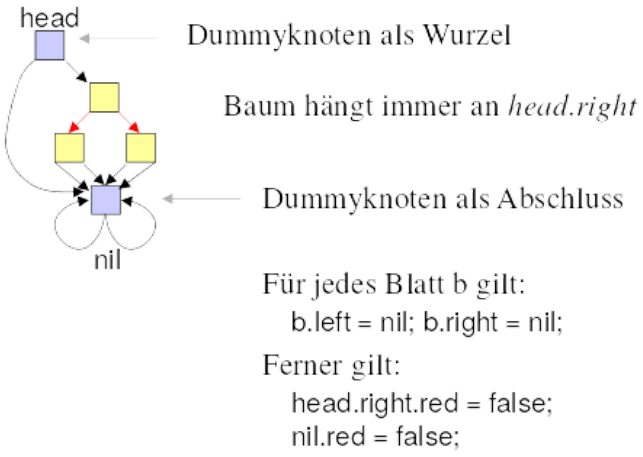
Simulieren Sie das Einfügen der Buchstaben T R E E S R O C K B I G T I M E (in dieser Reihenfolge) in einen **234-Baum** und einen **Rot-Schwarz-Baum**. Zeichnen Sie beide Bäume schrittweise nach jedem eingefügten Buchstaben.
Hinweis: Sortieren Sie Buchstaben, die kleiner als die Wurzel sind, links ein und sortieren Sie Buchstaben, die größer oder gleich der Wurzel sind, rechts ein.

Abzugeben ist: Zeichnungen

Aufgabe 2: Insert Rot/Schwarz-Baum (10 Punkte)

Implementieren Sie die ***insert(char key)*** Methode eines Rot-Schwarz-Baums, der Zeichen speichert. Verwenden Sie dazu die vorgegebene Klasse **at.jku.students.redblacktree.*RedBlackTree***.

Hinweis: Beachten Sie, dass jeweils ein Dummyknoten als Wurzel und als Abschluss verwendet wird (siehe Vorlesungsunterlagen).



Abzugeben ist: Java-Code

Bonusaufgabe 3: Traversierungs Iteratoren (2+2+2=6 Punkte)

Implementieren Sie die Baum Traversierungen aus Übung 3 (InOrder,PreOrder,PostOrder) als *Iteratoren.* Dafür sind die die Methoden ***next()*** und ***hasNext()*** in den Klassen ***InOrderTreeIterator*, *PreOrderTreeIterator*** und ***PostOrderTreeIterator*** im Package **at.jku.students.redblacktree** zu implementieren**.** Die Iteratoren liefern die Werte des Baumes in der jeweiligen Traversierungsreihenfolge.

Die Traversierung des Binärbaums soll "lazy" passieren, das heißt erst beim Aufruf von ***next()*** und dem jeweilige Traversierungsschritt wird der nächste Wert der Traversierungsreihenfolge berechnet. Es ist also keine gültige Lösung, die Traversierung des Baumes im Konstruktor abzuarbeiten und diese zu speichern und bei ***next()*** zurückzugeben. Ebenso ist das mehrfache traversieren des Baums nicht erlaubt. Beachten Sie, dass für die Lösung dieser Aufgabe der *LinkedListStack* (aus dem Package at.jku.ssw.stack.impl) als Hilfsdatenstruktur herangezogen werden kann.

Tipp: Simulieren Sie beim Aufruf von ***next()*** die Traversierung mit einer Hilfsdatenstruktur die den nächsten Rückgabewert von ***next()*** berechnet.

Abzugeben ist: Java-Code

Implementierungshinweise:

- Verwenden Sie das Vorgabeprojekt **PI2_UE05.zip**.
- Fügen Sie Ihre Implementierung in den mit **TODO** markierten Teilen der Klasse *RedBlackTree* ein.
- Ändern sie **keine public Interfaces** vorgegebener Skeleton Klassen (mit Ausnahme der Sichtbarkeiten).
- Halten Sie sich an die **Codierungsrichtlinien** auf der Kurs Website.