Shohin Abdulkhamidov

Professor K. Potika

CS-146

December 3, 2021

**Programming Project 4 and its goal.**

In this project we created a Dictionary, aka a Red Black Tree and inserted the provided link to a dictionary with the help of Red Black Tree

**It consists of such classes below:**

1. RedBlackTree.java consists of isLeaf, visit, printTree, addNode, insert, lookup, getSiblings, getAunt, getGrandparent, rotateLeft, rotateRight, fixTree, isEmpty, isLeftChild, preOrderVisit methods.
2. RBTTester.java is JUnit for RedBlackTree class
3. Dictionary.java consists of generateDictionary(), wordsOfPoem(), and spellChecker() methods
4. DictionaryJUnit.java is JUnit for Dictionary class
5. Node.java consists of getRoot, setRoot, compareTo parameters and isLeaf methods
6. Visitor.java consists of visit parameter

**Additional Requirements (Optional)**

1. I created JUnit class and separated all the test cases in that class to avoid having long lines of code inside the Main class
2. Inside the JUnit class I created a method to better understand how efficient my algorithms. It basically shows the nanoseconds spent while traversing the method.

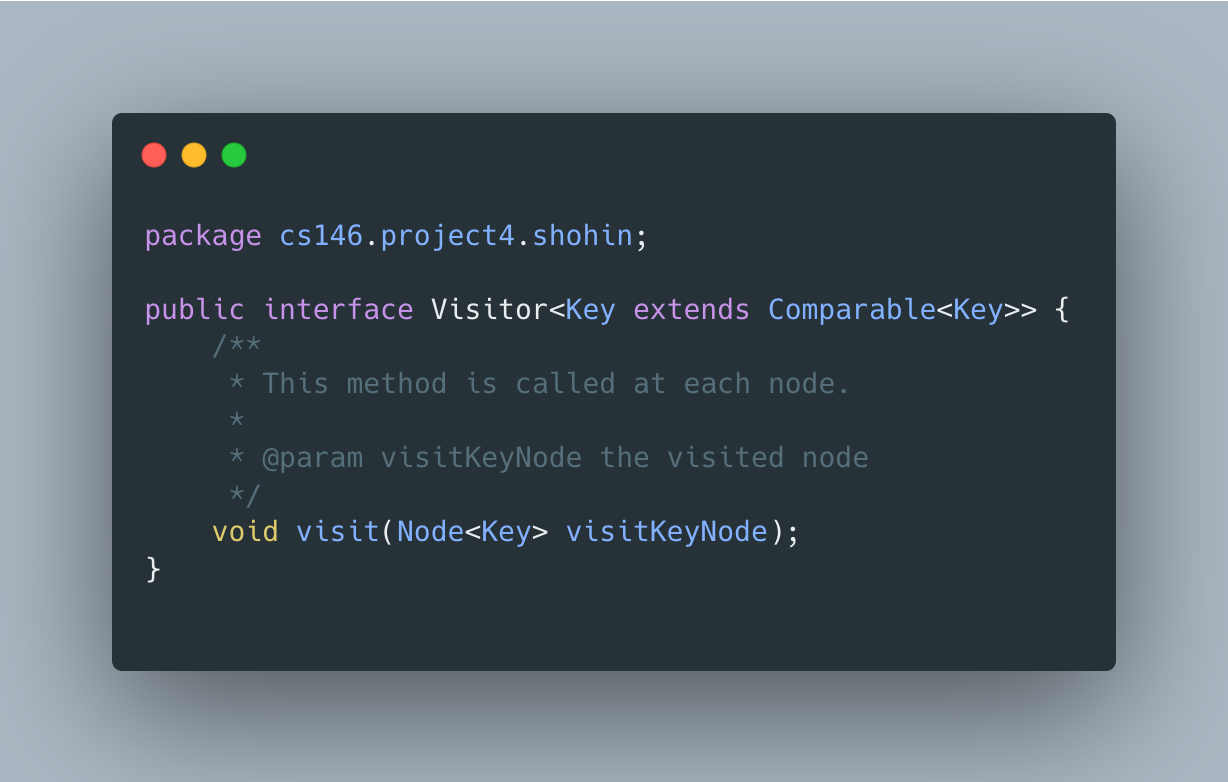
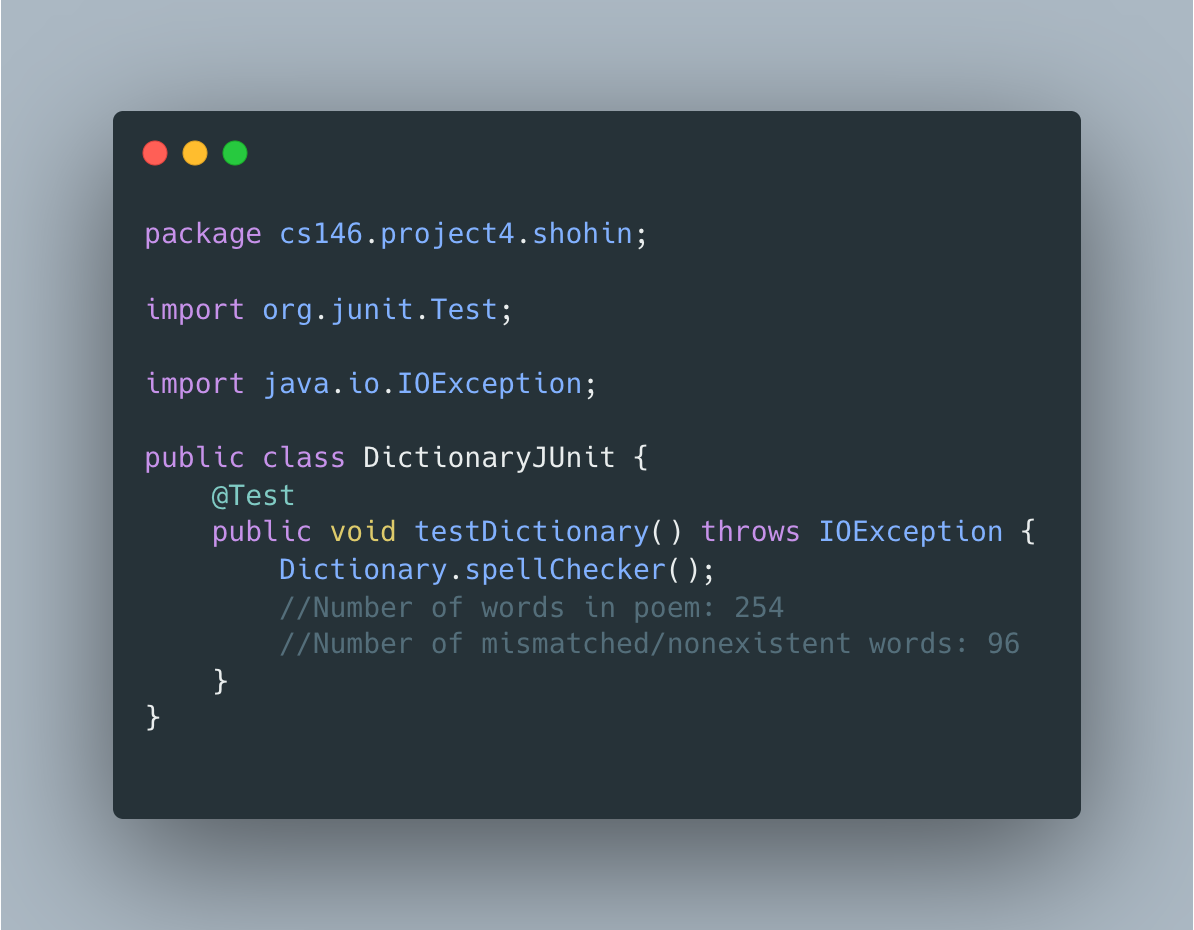
**Cases Consider**

Tested cased according to requirements

**Conclusion**

In conclusion I learned how Red Black Tree works.

**ZOOM IN INSIDE PDF TO GO THROUGH THIS CODE**

****

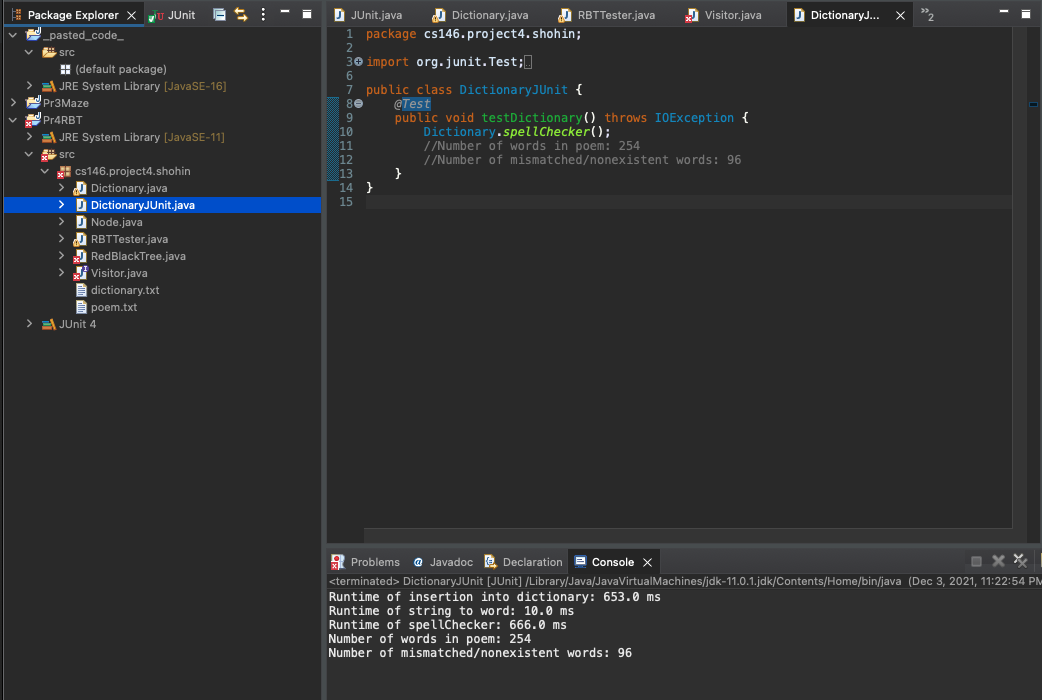
****

****

**Instructions:**

1. Downloaded Eclipsefor Mac
2. Created Java Project
3. Created Package as written in the requirements.
4. Created RedBlackTree, Dictionary, DictionaryJUnit, RBTTester, Node, and Visitor classes
5. Before running JUnit, we exported JUnit 4 of what’s required like Assert Equals, Assert True and etc.
6. Please Zoom In

This is separate Screenshot for Dictionary class JUnit that outputs number of words and mismatched words



And this is separate Screenshot for RBTester that outputs DBACFEHGIJ

