GTU Department of Computer Engineering

CSE 222/505 - Spring 2020

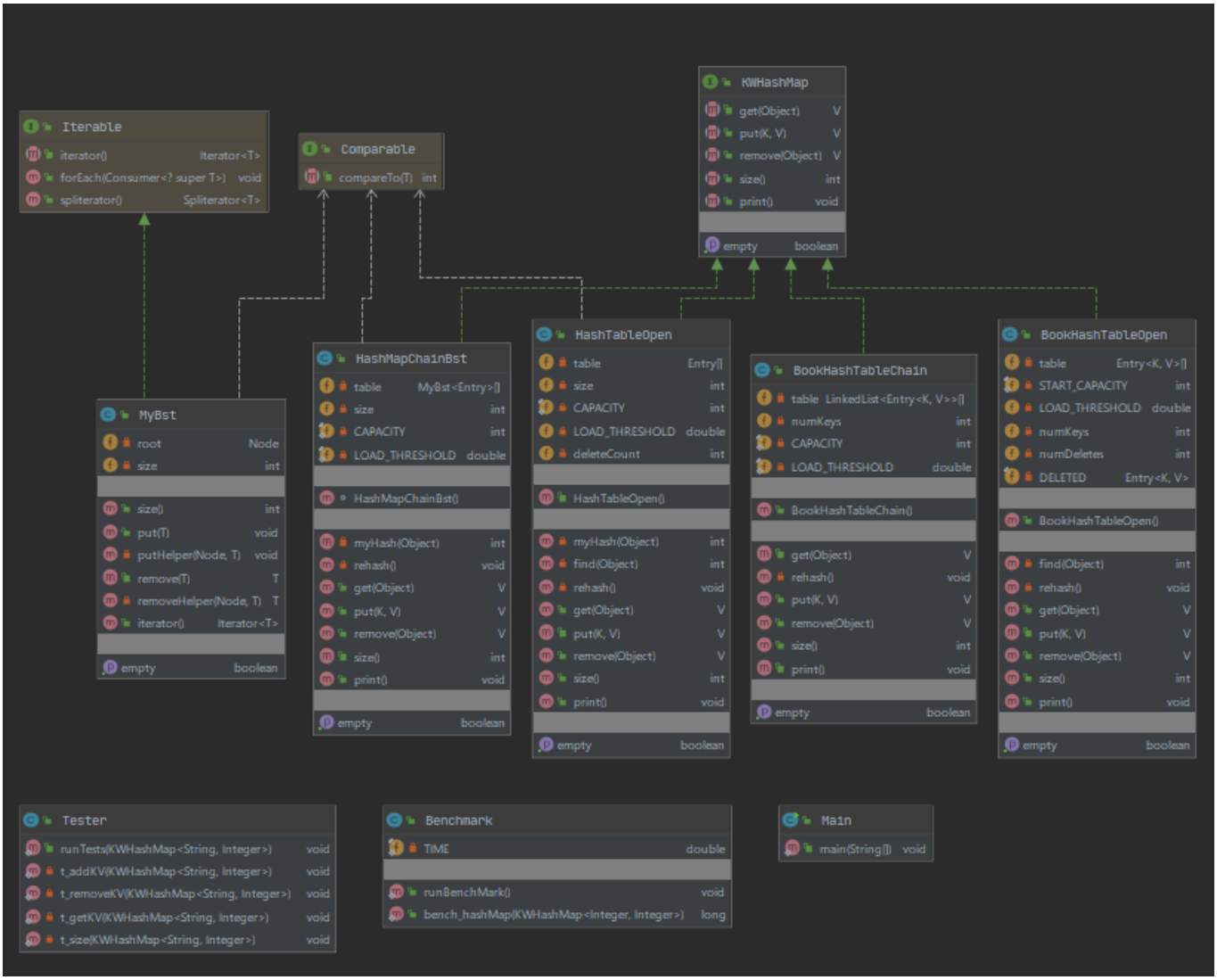
Homework 6 Report

Q4

Buğra Eren Yılmaz

1801042669

1. Class Diagrams



2. Problem Solution Approach

For this problem I needed to implement various hash maps, given as in the KWHashMap interface.

The book already had some implementations like, chaining linkedlists for implementing hash map and open table with linear probing.

It heavily used iterable structures as you can see like linkedlists. But the problem at hand needed binary tree structure to implement the KWHashMap, which is not iterable out-of-box. So I started from there.

I created BST class which is a fully iterable binary search tree structure, which also has custom iterator implementations too.

Used that class for HashTableChainBinTree implementation. It was easy to implement because I already had the linkedlist version of it from the book. Just translated the code to use BST instead of linkedlist. After all they were both iterable collection classes.

After that I wrote the tests for testing KWHashMap structures, which is Tester class. The tests contains every available method of a KWHashMap.

For double hashing open table, I used the HashTableOpen from book but instead of single hashing when finding index, I double hashed.

For benchmarks I created a class called Benchmark. It benchmarks KWHashMap structures.

The benchmark test consists of adding 150k elements to map and reading 150k elements from the map.

3. Test Cases

```
private static void t_addKV(KWHashMap<String, Integer> map) {
    System.out.println("Adding key/value pairs to map.");
    map.put("Apple", 5);
    map.put("Banana", 12);
    map.put("Orange", 0);
    map.put("Cherry", 43);
    map.put("Citrus", 64);
    map.put("Grapes", 21);
    System.out.println("Add done.");
    map.print();
}

private static void t_removeKV(KWHashMap<String, Integer> map) {
    System.out.println("Removing key/value pairs (Apple, Citrus) from map.");
    map.remove("Apple");
    map.remove("Citrus");
    System.out.println("Remove done.");
    map.print();
}

private static void t_getKV(KWHashMap<String, Integer> map) {
    System.out.println("Get key Orange from map.");
    int value = map.get("Banana");
    System.out.println("Get done: " + value);
}

private static void t_size(KWHashMap<String, Integer> map) {
    System.out.println("Get size of map.");
    int size = map.size();
    System.out.println("Size: " + size);
}
```

4. Results

```
Benchmark my HashMap with chaining Binary Trees:
Time: 0.1292858s
Benchmark my HashMap with open table double hashing:
Time: 0.0376061s
Benchmark book HashMap with chaining linkedlists:
Time: 0.0973724s
Benchmark book HashMap with open table linear probing:
Time: 0.0423506s
==END OF BENCHMARK==
```