CSE214 Data Structures

Map

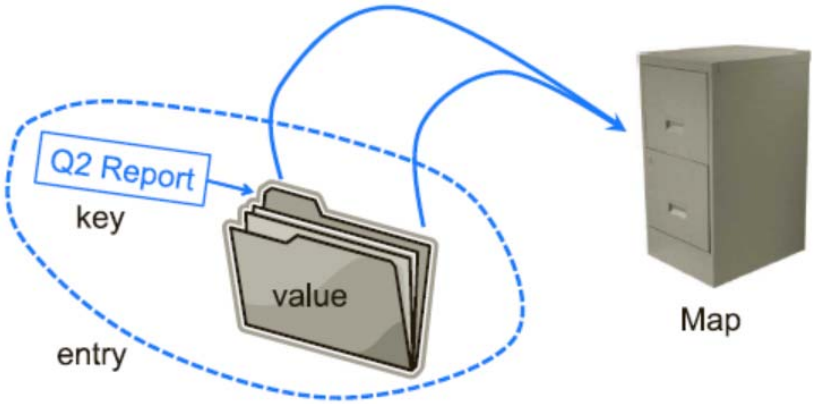
YoungMin Kwon

Map Abstract Data Type

- Map
  - Map is an abstract data type for efficiently storing and retrieving values based on unique search keys
  - Maps store key-value pairs (k, v) called entries
  - Maps are also known as associative arrays
    - Keys serve somewhat like indexes into the map

Map Abstract Data Type

- Map examples



  - Keys are labels
  - Values are folders
  - Map is the file cabinet

Interface Map

- java.util.Map<K,V> has
  - void clear(): to remove all entries in the map
  - boolean containsKey(Object key): to check if the map has an entry with the key
  - boolean containsValue(Object value): to check if the map has an entry with the value
  - Set<Entry<K,V>> entrySet(): all entries in the map as a set

Interface Map

- V get(Object key): return the value associated with the key
- V put(K key, V value): add an entry with key and value to the map
- void remove(Object key): remove the entry with key

```java
public class WordCounter {
    //Use Map to count word occurrences
    Map<String, Integer> map;
    String fileName;

    public WordCounter (String fileName) {
        this.fileName = fileName;
        //TODO: create a HashMap for map
    }

    public void readFile() throws IOException {
        //read text from file
        FileReader r = new FileReader(fileName);

        //divide the text into words
        StreamTokenizer tok = new StreamTokenizer(r);

        //read through the file
        while(tok.nextToken() != StreamTokenizer.TT_EOF) {
            if(tok.ttype != StreamTokenizer.TT_WORD)
                continue;
            String word = tok.sval;
            //TODO: add word as a new entry or increase the counter
        }
    }
```

```
public void printWords() {
        System.out.println("All words");
        //TODO: print all entries in the form of <word>: <frequency>

        System.out.println("");

        System.out.println("All words sorted by key");
        ArrayList<Entry<String, Integer>> byKey = new ArrayList<>();
        //TODO: sort sort by key
        //      add all entries to byKey
        //      sort byKey using Collections.sort

        System.out.println("");


        System.out.println("All words sorted by value");
        ArrayList<Entry<String, Integer>> byVal = new ArrayList<>();
        //TODO: sort sort by value
        //      add all entries to byVal
        //      sort byVal using Collections.sort

        System.out.println("");
    }
```

```
 public static void main(String[] args) {
        String fileName = "C:\\Users\\... \\WordCounter.java";
        WordCounter wc = new WordCounter(fileName);
        try {
            wc.readFile();
            wc.printWords();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
…
All words sorted by value
String: 15
e: 15
sort: 10
…
```