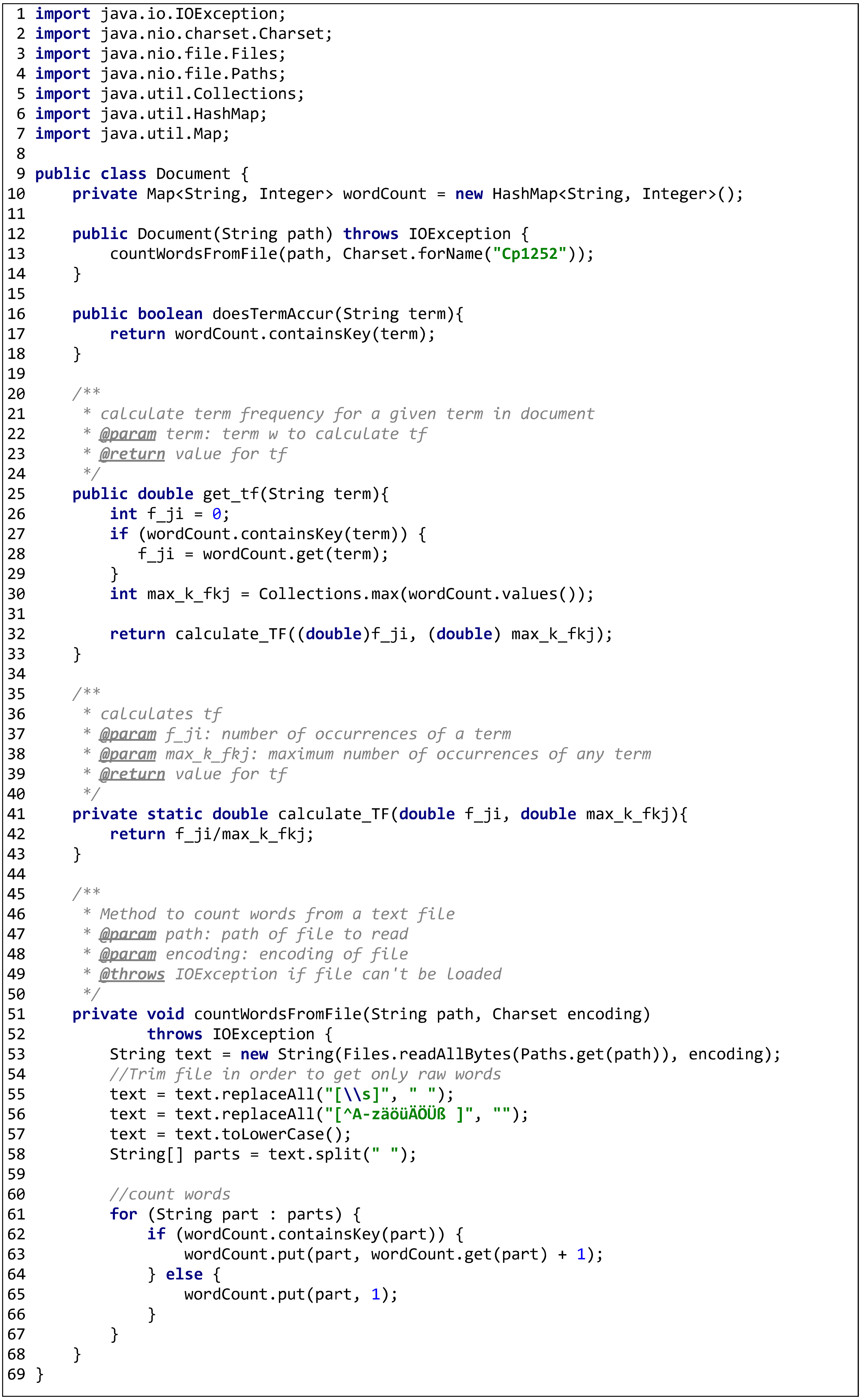

```java
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

public class Document {
    private Map<String, Integer> wordCount = new HashMap<String, Integer>();

    public Document(String path) throws IOException {
        countWordsFromFile(path, Charset.forName("Cp1252"));
    }

    public boolean doesTermAccur(String term){
        return wordCount.containsKey(term);
    }

    /**
     * calculate term frequency for a given term in document
     * @param term: term w to calculate tf
     * @return value for tf
     */
    public double get_tf(String term){
        int f_ji = 0;
        if (wordCount.containsKey(term)) {
            f_ji = wordCount.get(term);
        }
        int max_k_fkj = Collections.max(wordCount.values());

        return calculate_TF((double)f_ji, (double) max_k_fkj);
    }

    /**
     * calculates tf
     * @param f_ji: number of occurrences of a term
     * @param max_k_fkj: maximum number of occurrences of any term
     * @return value for tf
     */
    private static double calculate_TF(double f_ji, double max_k_fkj){
        return f_ji/max_k_fkj;
    }

    /**
     * Method to count words from a text file
     * @param path: path of file to read
     * @param encoding: encoding of file
     * @throws IOException if file can't be loaded
     */
    private void countWordsFromFile(String path, Charset encoding)
            throws IOException {
        String text = new String(Files.readAllBytes(Paths.get(path)), encoding);
        //Trim file in order to get only raw words
        text = text.replaceAll("[\\s]", " ");
        text = text.replaceAll("[^A-zäöüÄÖÜß ]", "");
        text = text.toLowerCase();
        String[] parts = text.split(" ");

        //count words
        for (String part : parts) {
            if (wordCount.containsKey(part)) {
                wordCount.put(part, wordCount.get(part) + 1);
            } else {
                wordCount.put(part, 1);
            }
        }
    }
}
```