1.import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class PositiveNumberChecker {
    public static void main(String[] args) {
        String filePath = "C:\Users\bhala\OneDrive\Documents\Java-Day-5-AF\number.txt";

        try {
            checkForPositiveNumbers(filePath);
        } catch (PositiveNumberException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("Error reading the file: " + e.getMessage());
        }
    }

    private static void checkForPositiveNumbers(String filePath) throws IOException,
PositiveNumberException {
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            System.out.print("Content of " + filePath + ": ");
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.print(line + " ");
                String[] numbers = line.split("\\s+");
                for (String number : numbers) {
                    int num = Integer.parseInt(number);
                    if (num > 0) {
                        throw new PositiveNumberException("Positive number found: " + num);
                    }
                }
            }
        }
    }

    static class PositiveNumberException extends Exception {
        public PositiveNumberException(String message) {
            super(message);
        }
    }
}

2.
3.import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.nio.file.DirectoryStream;
import java.nio.file.FileVisitOption;

```
import java.nio.file.FileVisitResult;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.HashMap;
import java.util.Map;
import java.util.stream.Stream;

public class MostCommonWordFinder {

    public static void main(String[] args) {
        System.out.print("Enter directory name: ");
        String directoryPath = System.console().readLine();

        try {
            findMostCommonWords(directoryPath);
        } catch (IOException e) {
            System.out.println("Error reading files: " + e.getMessage());
        }
    }

    private static void findMostCommonWords(String directoryPath) throws IOException {
        Map<String, Integer> wordFrequencyMap = new HashMap<>();

        try (Stream<Path> paths = Files.walk(Paths.get(directoryPath),
FileVisitOption.FOLLOW_LINKS)) {
            paths.filter(Files::isRegularFile)
                 .forEach(file -> processFile(file, wordFrequencyMap));
        }

        int maxFrequency =
wordFrequencyMap.values().stream().max(Integer::compareTo).orElse(0);

        System.out.println("Most common words:");

        wordFrequencyMap.entrySet().stream()
                .filter(entry -> entry.getValue() == maxFrequency)
                .forEach(entry -> System.out.println("Word: " + entry.getKey() + ", Frequency: " +
entry.getValue()));
    }

    private static void processFile(Path filePath, Map<String, Integer> wordFrequencyMap) {
        try (BufferedReader reader = new BufferedReader(new FileReader(filePath.toString())))
{
            String line;
            while ((line = reader.readLine()) != null) {
                String[] words = line.toLowerCase().split("\\s+");
                for (String word : words) {
```

```
                wordFrequencyMap.put(word, wordFrequencyMap.getOrDefault(word, 0) + 1);
            }
        }
    } catch (IOException e) {
        System.out.println("Error reading file " + filePath + ": " + e.getMessage());
    }
  }
}
```