Informatik II

Faust mit Array

Jan Lukas Deichmann

April 19, 2016

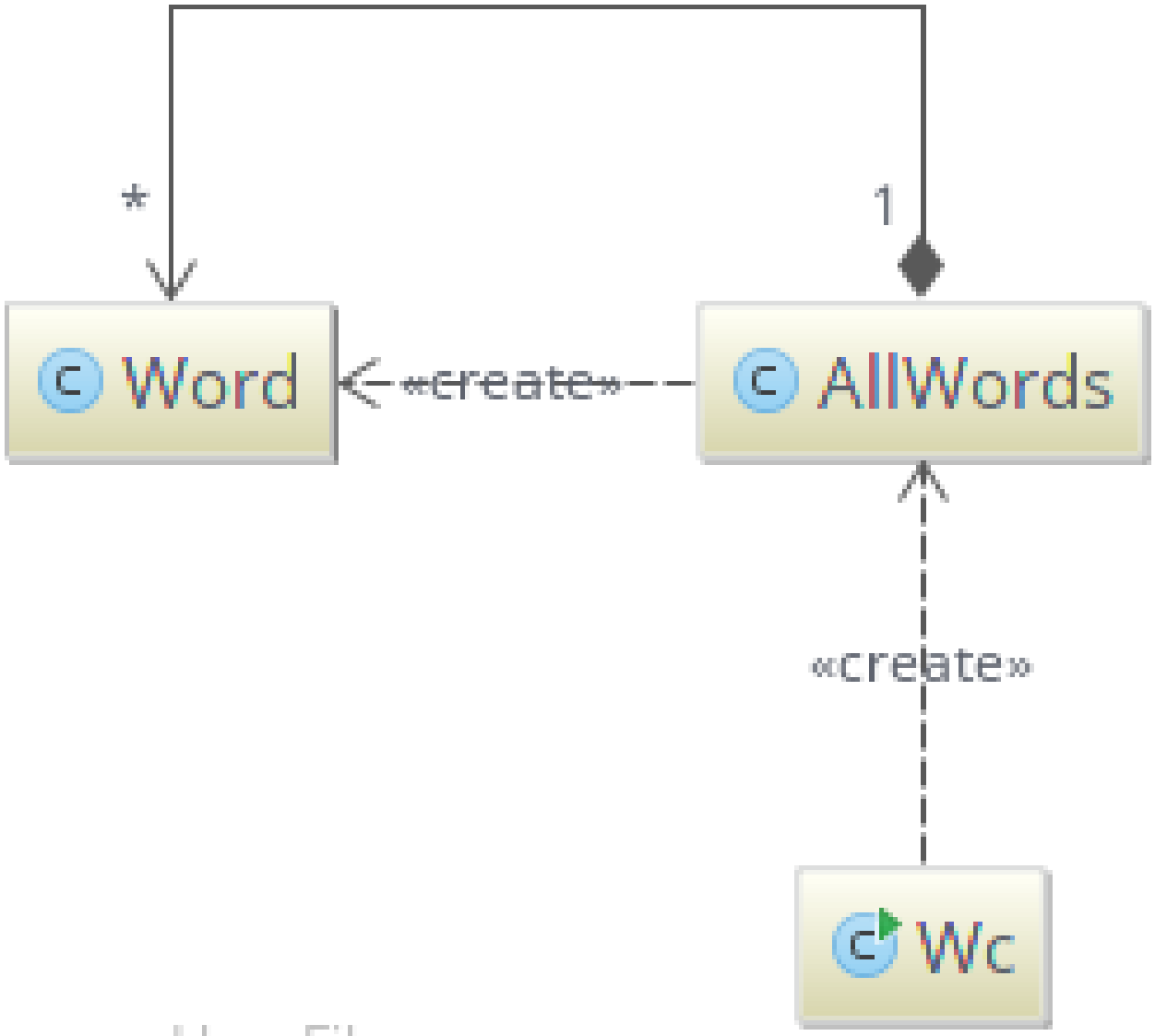
ToC

UML Diagramm

Code

- Word
- AllWords
- Wc

UML Diagramm



Word

```
public class Word implements Comparable<Word> {
 private String content;
 private int n;
 public Word(String s) {
  content = s;// s als content übernehmen,
  n = 1;// zähler auf 1 setzen (erstes Auftreten)
 }
 ..
```

Word

```
...
public int count() { return n; }
public String content() { return content; }
public void inc() { n++; }
public int compareTo(Word w) {
 if (w != null) {
  return w.count() - this.count();
 } else {
  return 1;
 }
}
public String toString(){
 return n + " : " + content;
 // "Häufigkeit : Wort"
}
}
```

AllWords

```
public class AllWords {
 private Word words[];
 private int wordsSize = 0;
 public AllWords (int max) { words = new Word[max];}
```

AllWords - Add

```
public void add(String s) {
 Word sWord = new Word(s);
 boolean found = false;
 if (wordsSize == 0) {
  words[0] = sWord;
  wordsSize++;
 } else {

   ...

 }
}
```

AllWords - Add

```
for (int i = 0; i < wordsSize; i++) { // Laufzeit: O(n/2)
 if (words[i].content().equals(sWord.content())) {
  words[i].inc();
  found = true;
  break;
 }
}
if (found == false) {
 if (words.length == wordsSize) {
  System.exit(-1);
 } else {
  words[wordsSize] = sWord;
  wordsSize++;
 }
}
```

AllWords - Sort

```
public void sort() {

 Word newWords[] = new Word[wordsSize]; // newArray
 System.arraycopy( words, 0, newWords, 0, wordsSize );

 words = newWords;
 Arrays.sort(words);

}
```

AllWords

```
public int distinctWords() {
 return wordsSize;
}
public int totalWords() {
 int count = 0;
 for (int i = 0; i < wordsSize; i++) {
  count = count + words[i].count();
 }
 return count;
}
public String toString() {
 String stringWords = "";
 for (int i = 0; i < wordsSize; i++) {
  stringWords = stringWords + words[i].toString() + "\n";
 }
 return stringWords;
}
```

Wc

```
public static int countWords(InputStream in) {
 AllWords words = new AllWords(15000);
 try {
  Scanner scanner = new Scanner(in);
  while (scanner.hasNext()) {
   String t = scanner.next()
               .replaceAll("[^\\p{L}\\p{Nd}]+", "");
   if (t.length() > 0) { words.add(t); }
  }
  scanner.close();
 } catch (Exception e) {
  System.out.println(e.toString());
 }
 words.sort();
 System.out.println(words.toString());
 System.out.println(words.distinctWords());
 System.out.println(words.totalWords());
 return words.totalWords();
}
```

Wc Calls

```
public static void main(String[] args) {
 countWords(System.in);
}
```

Wc Calls

```
@org.junit.Test
public void testFaust() {
 try {
  InputStream in;
  String content = readFile("/pfad/Faust.txt",
                            StandardCharsets.UTF_8);
  in = new ByteArrayInputStream( content.getBytes() );
  assertTrue(Wc.countWords(in) == 30628);
 } catch (IOException e) {
  e.printStackTrace();
  fail("IO Error");
 }
}
```