

Brendan Whitaker

CSE 2221 Homework 20

Professor Bucci

11 April 2017

1. We implement the generator method:

```
/**
 * Generates the set of characters in the given
 * {@code String} into the
 * given {@code Set}.
 *
 * @param str
 *         the given {@code String}
 * @param strSet
 *         the {@code Set} to be replaced
 * @replaces strSet
 * @ensures strSet = entries(str)
 */
private static void generateElements(String str,
    Set<Character> strSet) {
    strSet.clear();
    int length = str.length();
    for (int i = 0; i < length; i++) {
        if (!strSet.contains(str.charAt(i))) {
            strSet.add(str.charAt(i));
        }
    }
}
```

2. We implement the word return method:

```
/**
 * Returns the first "word" (maximal length string
 * of characters not in
 * {@code separators}) or "separator string"
 * (maximal length string of
```

```

* characters in {@code separators}) in the
* given {@code text} starting at
* the given {@code position}.
*
* @param text
*           the {@code String} from which
* to get the word or separator
*           string
* @param position
*           the starting index
* @param separators
*           the {@code Set} of separator
* characters
* @return the first word or separator string
* found in {@code text} starting
*       at index {@code position}
* @requires 0 <= position < |text|
* @ensures <pre>
* nextWordOrSeparator =
*   text[position, position +
* |nextWordOrSeparator|) and
* if entries(text[position, position + 1))
* intersection separators = {}
* then
*   entries(nextWordOrSeparator) intersection
* separators = {} and
*   (position + |nextWordOrSeparator| = |text|
* or
*   entries(text[position, position +
* |nextWordOrSeparator| + 1))
*   intersection separators /= {})
* else
*   entries(nextWordOrSeparator) is subset
* of separators and
*   (position + |nextWordOrSeparator| = |text|
* or
*   entries(text[position, position +
* |nextWordOrSeparator| + 1))
*   is not subset of separators)
* </pre>
*/
private static String nextWordOrSeparator(
String text, int position,
    Set<Character> separators) {
    Queue<Character> nextWord = new Queue1L<Character>();
    int i = position;

```

```

char nextChar = text.charAt(i);
if (separators.contains(nextChar)) {
    while (separators.contains(nextChar)) {
        nextWord.enqueue(nextChar);
        i++;
        nextChar = text.charAt(i);
    }
} else {
    i = position;
    while (!separators.contains(nextChar)) {
        nextWord.enqueue(nextChar);
        i++;
        nextChar = text.charAt(i);
    }
}
int length = nextWord.length();
char[] nextWordArray;
nextWordArray = new char[length];
for (int j = 0; j < length; j++) {
    nextWordArray[j] = nextWord.dequeue();
}
String word = "";
word = word.valueOf(nextWordArray);
return word;
}

```