```java
import static org.junit.Assert.assertEquals;

public class GlossaryTest {
    /*
     * tests of generateElements
     */
    @Test
    public void testGenerateElements1() {
        String str = " . !   ";
        Set<Character> charSet = new Set1L<>();
        Glossary.generateElements(str, charSet);
        Set<Character> charSetExpected = charSet.newInstance();
        charSetExpected.add(' ');
        charSetExpected.add('.');
        charSetExpected.add('!');
        assertEquals(" . !   ", str);
        assertEquals(charSetExpected, charSet);
    }

    @Test
    public void testGenerateElements2() {
        String str = "heyo";
        Set<Character> charSet = new Set1L<>();
        Glossary.generateElements(str, charSet);
        Set<Character> charSetExpected = charSet.newInstance();
        charSetExpected.add('h');
        charSetExpected.add('e');
        charSetExpected.add('y');
        charSetExpected.add('o');
        assertEquals("heyo", str);
        assertEquals(charSetExpected, charSet);
    }

    //test about border
    @Test
    public void testGenerateElements3() {
        String str = "c";
        Set<Character> charSet = new Set1L<>();
        Glossary.generateElements(str, charSet);
        Set<Character> charSetExpected = charSet.newInstance();
        charSetExpected.add('c');
        assertEquals("c", str);
        assertEquals(charSetExpected, charSet);
    }

    /*
     * tests of nextWordOrSeparator
     */
    @Test
    public void testNextWordOrSeparator1() {
        int position = 0;
        String text = "u";
        final String separStr = " \t ";
        Set<Character> separators = new Set1L<>();
        Glossary.generateElements(separStr, separators);
        String res = Glossary.nextWordOrSeparator(text, position, separators);
        assertEquals("u", res);
    }

    @Test
    public void testNextWordOrSeparator2() {
        int position = 0;
        String text = "halo.hi";
        final String separStr = " . ";
        Set<Character> separators = new Set1L<>();
        Glossary.generateElements(separStr, separators);
        String res = Glossary.nextWordOrSeparator(text, position, separators);
        assertEquals("halo", res);
    }

    @Test
    public void testNextWordOrSeparator3() {
        String text = "...";
        int position = 0;
        final String separStr = " . ";
        Set<Character> separators = new Set1L<>();
        Glossary.generateElements(separStr, separators);
        String res = Glossary.nextWordOrSeparator(text, position, separators);
        assertEquals(".", res);
    }

    @Test
    public void testNextWordOrSeparator4() {
        String text = "hello..";
        int position = 5;
        final String separStr = " . ";
        Set<Character> separators = new Set1L<>();
        Glossary.generateElements(separStr, separators);
        String res = Glossary.nextWordOrSeparator(text, position, separators);
        assertEquals(".", res);
    }

    /*
     * tests of generateIn
     */

    @Test
    public void testGenereateIn1() {
        SimpleReader in = new SimpleReader1L("test1.txt");
        Queue<String> title = new Queue1L<>();
        Map<String, String> word = new Map1L<>();
        title = Glossary.generateIn(word, in);
        Queue<String> titleExpected = title.newInstance();
        titleExpected.enqueue("meaning");
        Map<String, String> wordExpected = new Map1L<>();
        wordExpected.add("meaning",
                "something that one wishes to convey, especially by language");

        assertEquals(titleExpected, title);
        assertEquals(wordExpected, word);
    }

    @Test
    public void testGenereateIn2() {
        SimpleReader in = new SimpleReader1L("test2.txt");
        Queue<String> title = new Queue1L<>();
        Map<String, String> word = new Map1L<>();
        title = Glossary.generateIn(word, in);
        Queue<String> titleExpected = title.newInstance();
        titleExpected.enqueue("meaning");
        titleExpected.enqueue("term");
        Map<String, String> wordExpected = new Map1L<>();
        wordExpected.add("meaning",
                "something that one wishes to convey, especially by language");
```

```java
        wordExpected.add("term", "a word whose definition is in a glossary");

        assertEquals(titleExpected, title);
        assertEquals(wordExpected, word);
    }

    @Test
    public void testGenereateIn3() {
        SimpleReader in = new SimpleReader1L("test3.txt");
        Queue<String> title = new Queue1L<>();
        Map<String, String> word = new Map1L<>();
        title = Glossary.generateIn(word, in);
        Queue<String> titleExpected = title.newInstance();
        titleExpected.enqueue("meaning");
        titleExpected.enqueue("glossary");
        Map<String, String> wordExpected = new Map1L<>();
        wordExpected.add("meaning",
                "something that one wishes to convey, especially by language");
        wordExpected.add("glossary",
                "a list of difficult or specialized terms, with their definitions,usually near the end of a book");

        assertEquals(titleExpected, title);
        assertEquals(wordExpected, word);
    }

    @Test
    public void changeTheTerms1() {
        Map<String, String> word = new Map1L<>();
        Queue<String> title = new Queue1L<>();
        word.add("book", "a printed or written literary work");
        title.enqueue("book");
        Set<Character> strSet = new Set1L<>();
        strSet.add(' ');
        strSet.add(',');
        strSet.add('.');
        Glossary.changeTheTerms(word, title, strSet);
        Map<String, String> wordExpected = new Map1L<>();
        wordExpected.add("book", "a printed or written literary work");
        assertEquals(wordExpected, word);
    }

}

```