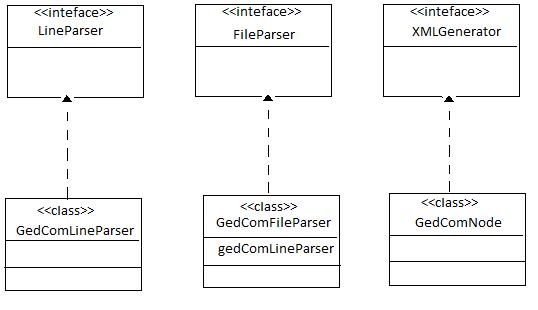
Approach:

* Need a parser to parse the file, represent the parsed data as a bean, generate xml from this, and write the data to the file.
* So, there is a need for reading the file(may be line by line), parse each line, create a hierarchical bean having parent, children, then convert this bean to xml and write to a file.
* Started thinking about all possible, test cases and scenarios of data, taken from the sample file given. Started TDD – it helped in every step of refactoring and the confidence to build the system, without worrying of bugs.

Design:



/gedcom/src/main/java

**com.gedcom.elements**

* GedComAttribute.java --> Enum - whether a GedComNode is a TAG or ID
* GedComNode.java --> reperesents parsed gedcom data, parent and child relations also maintained, implements XMLGenerator
* LineData.java --> represents a parsed gedcom single line data

**com.gedcom.parser**

* GedComFileParser.java --> Parser for parsing gedcom file, implements FileParser
* GedComLineParser.java --> parses a gedcom line, implements LineParser

**com.gedcom.util**

* FileParser.java --> interface
* FileUtility.java --> utility class for reading a file and writing data to a file
* GedComToXMLConverter.java --> Main class, which convets gedcom file to xml file, accepts input file name and output file name as arguments
* LineParser.java --> interface
* XMLGenerator.java -> interface