```java
package xyz.amtstl.soup;

import java.util.List;

public class Parser {
    public static int inx;

    /**
     * Parser that gets the numbers
     * @param i
     * @param cache
     * @return
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @deprecated use the new Parser (it's integrated with InterVar)
     */
    public static String[] parseNumbers(int i, String cache) throws NumberFormatException, SoupVariableException {
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == '}'){
                index = e;
                break;
            }
            else if(cache.charAt(e) != '}') {
                whole+=cache.charAt(e);
            }
        }

        inx = index;

        whole = whole.substring(2, whole.length());

        String[] numbers = whole.split(",");
        return numbers;
    }

    /**
     * General parser for soup
     * @param i
     * @param cache
     * @return
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     */
    public static List<String> parse(int i, String cache) throws NumberFormatException, SoupVariableException, SoupSyntaxException {
        String whole = "";

        int index = 0;

        for (int e = i; e < cache.length(); e++) {
            if (cache.charAt(e) == '}'){
```

```
60                  index = e;
61                  break;
62              }
63              else if(cache.charAt(e) != '}') {
64                  whole+=cache.charAt(e);
65              }
66          }
67
68          inx = index;
69
70          whole = whole.substring(2, whole.length());
71
72          InterVar.parseInternalVar(whole.split(","));
73          return InterVar.parsedNumbers;
74      }
75
76      /**
77       * Parser for special functions that don't use the bracket delimiter
78       * @param i
79       * @param cache
80       * @return
81       * @throws NumberFormatException
82       * @throws SoupVariableException
83       * @throws SoupSyntaxException
84       */
85      public static List<String> parseInternalFunctions(int i, String cache) throws
    NumberFormatException, SoupVariableException, SoupSyntaxException {
86          String whole = "";
87
88          int index = 0;
89
90          for (int e = i; e < cache.length(); e++) {
91              if (cache.charAt(e) == ')'){
92                  index = e;
93                  break;
94              }
95              else if(cache.charAt(e) != ')') {
96                  whole+=cache.charAt(e);
97              }
98          }
99
100         inx = index;
101
102         whole = whole.substring(2, whole.length());
103
104         InterVar.parseInternalVar(whole.split("!"));
105         return InterVar.parsedNumbers;
106     }
107
108     /**
109      * Parses a single number
110      * @deprecated
111      * @param i
112      * @param cache
113      * @return
114      */
115     public static String parseSingle(int i, String cache) {
```

```
116         String whole = "";
117 
118         int index = 0;
119 
120         for (int e = i; e < cache.length(); e++) {
121             if (cache.charAt(e) == '}'){
122                 index = e;
123                 break;
124             }
125             else if(cache.charAt(e) != '}') {
126                 whole+=cache.charAt(e);
127             }
128         }
129 
130         inx = index;
131         System.out.println(String.valueOf(whole.charAt(2)));
132         whole = whole.substring(2, whole.length());
133         return whole;
134     }
135 }
```