```java
package xyz.amtstl.soup;

import java.util.List;

public class Parser {
    private 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 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) == '}'){
```

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

```
117         String whole = "";
118 
119         int index = 0;
120 
121         for (int e = i; e < cache.length(); e++) {
122             if (cache.charAt(e) == '}'){
123                 index = e;
124                 break;
125             }
126             else if(cache.charAt(e) != '}') {
127                 whole+=cache.charAt(e);
128             }
129         }
130 
131         inx = index;
132         System.out.println(String.valueOf(whole.charAt(2)));
133         whole = whole.substring(2, whole.length());
134         return whole;
135     }
136 
137     public int getIndex() {
138         return inx;
139     }
140 
141     public void setIndex(int newIndex) {
142         inx = newIndex;
143     }
144 }
```