```java
package xyz.amtstl.soup.logic;

import xyz.amtstl.soup.Parser;

public class Looper {
    private static int groundState = 0;

    /**
     * Executes a new loop sequence
     * @param minBound starting point
     * @param maxBound loop will continue until
     * @param cache line of code from program
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     * @throws SoupFunctionNotDeclaredException
     */
    public static void execNewForLoop(int minBound, int maxBound, String cache, String direction) throws NumberFormatException, SoupVariableException, SoupSyntaxException, SoupFunctionNotDeclaredException {
        groundState = LogicController.index;
        for (int e = minBound; e < maxBound; e++) {

            for (int i = groundState; i < cache.length(); i++) {
                if (cache.charAt(i) == ';') {
                    Soup.checkToken(i, cache, cache.charAt(i));
                    i = LogicController.index;
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }
            LogicController.index = groundState;
            VariableHandler.insertVar((float) Float.valueOf(e), 1000);

            if (LogicController.isBreak) {
                LogicController.setBreak(false);
                break;
            }
        }
    }

    /**
     * Function for doing a decrementing for loop
     * @param maxBound max bound
     * @param minBound min bound
     * @param cache
     * @param direction depreciated
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     * @throws SoupFunctionNotDeclaredException
     */
    public static void execNewForLoopDecre(int maxBound, int minBound, String cache, String direction) throws NumberFormatException, SoupVariableException, SoupSyntaxException, SoupFunctionNotDeclaredException {
        groundState = LogicController.index;
```

```
        for (int e = maxBound; e > minBound; e--) {

            for (int i = groundState; i < cache.length(); i++) {
                if (cache.charAt(i) == ';') {
                    Soup.checkToken(i, cache, cache.charAt(i));
                    i = LogicController.index;
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }
            LogicController.index = groundState;
            VariableHandler.insertVar((float) Float.valueOf(e), 1000);
            if (LogicController.isBreak) {
                LogicController.isBreak = false;
                break;
            }
        }
    }

    /**
     * Executes a new while loop
     * @param cache
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     * @throws SoupFunctionNotDeclaredException
     */
    public static void execNewWhileLoop(String cache) throws NumberFormatException,
  SoupVariableException, SoupSyntaxException, SoupFunctionNotDeclaredException {
        groundState = LogicController.index;

        int firstCondition = (int)Integer.valueOf((int)
  Float.parseFloat(LogicController.ns.get(0)));
        int secondCondition = (int)Integer.valueOf((int)
  Float.parseFloat(LogicController.ns.get(1)));

        while ((float)firstCondition == (float)secondCondition) {
            // parse the line
            for (int i = groundState; i < cache.length(); i++) {
                if (cache.charAt(i) == ';') {
                    Soup.checkToken(i, cache, cache.charAt(i));
                    i = LogicController.index;
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }

            LogicController.index = groundState;
            LogicController.ns = Parser.parse(0, cache);
            firstCondition = (int)Integer.valueOf((int)
  Float.parseFloat(LogicController.ns.get(0)));
            secondCondition = (int)Integer.valueOf((int)
  Float.parseFloat(LogicController.ns.get(1)));
            if (LogicController.isBreak) {
                LogicController.isBreak = false;
```

```
110                 break;
111             }
112         }
113     }
114 
115     /**
116      * Executes a new new while not loop
117      * @param cache
118      * @throws NumberFormatException
119      * @throws SoupVariableException
120      * @throws SoupSyntaxException
121      * @throws SoupFunctionNotDeclaredException
122      */
123     public static void execNewWhileNotLoop(String cache) throws NumberFormatException,
    SoupVariableException, SoupSyntaxException, SoupFunctionNotDeclaredException {
124         groundState = LogicController.index;
125 
126         int firstCondition = (int)Integer.valueOf((int)
    Float.parseFloat(LogicController.ns.get(0)));
127         int secondCondition = (int)Integer.valueOf((int)
    Float.parseFloat(LogicController.ns.get(1)));
128 
129         while ((float)firstCondition != (float)secondCondition) {
130             // parse the line
131             for (int i = groundState; i < cache.length(); i++) {
132                 if (cache.charAt(i) == ';') {
133                     Soup.checkToken(i, cache, cache.charAt(i));
134                     i = LogicController.index;
135                 }
136                 else {
137                     Soup.checkToken(i, cache, cache.charAt(i));
138                 }
139             }
140 
141             LogicController.index = groundState;
142             LogicController.ns = Parser.parse(0, cache);
143             firstCondition = (int)Integer.valueOf((int)
    Float.parseFloat(LogicController.ns.get(0)));
144             secondCondition = (int)Integer.valueOf((int)
    Float.parseFloat(LogicController.ns.get(1)));
145             if (LogicController.isBreak) {
146                 LogicController.isBreak = false;
147                 break;
148             }
149         }
150     }
151 }
```