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

import xyz.amtstl.soup.Soup;

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 = Soup.getMainLogic().getIndex();
        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 = Soup.getMainLogic().getIndex();
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }
            Soup.getMainLogic().setIndex(groundState);
            Soup.getMainLogic();
            LogicController.v.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 = Soup.getMainLogic().getIndex();
        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 = Soup.getMainLogic().getIndex();
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }
            Soup.getMainLogic().setIndex(groundState);
            LogicController.v.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 = Soup.getMainLogic().getIndex();

        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 = Soup.getMainLogic().getIndex();
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }

            Soup.getMainLogic().setIndex(groundState);
            LogicController.ns = LogicController.p.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 (Soup.getMainLogic().isBreak) {
                Soup.getMainLogic().isBreak = false;
                break;
            }
        }
    }

    /**
     * Executes a new new while not loop
     * @param cache
     * @throws NumberFormatException
     * @throws SoupVariableException
     * @throws SoupSyntaxException
     * @throws SoupFunctionNotDeclaredException
     */
    public static void execNewWhileNotLoop(String cache) throws NumberFormatException, SoupVariableException, SoupSyntaxException, SoupFunctionNotDeclaredException {
        groundState = Soup.getMainLogic().getIndex();

        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 = Soup.getMainLogic().getIndex();
                }
                else {
                    Soup.checkToken(i, cache, cache.charAt(i));
                }
            }

            Soup.getMainLogic().setIndex(groundState);
            LogicController.ns = LogicController.p.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 (Soup.getMainLogic().isBreak) {
                Soup.getMainLogic().isBreak = false;
                break;
            }
        }
    }
}
```