Software Requirements Specification

For

Logical Expression Calculator

Version 1.1.0 Approved By Amir Yousef

Prepared By Amir Yousef

September 5th 2015

Home

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Amir Yousef	9/2/2015	Beta Version Needs Modifications	1.0.0
Amir Yousef	9/5/2015	Final Version After All Modifications	1.1.0

1. Introduction

1.1 Purpose

This is java test I have to complete it to be considered for Jr. Java opportunity position.

1.2 Document Conventions

My standard or reference is the test page that I received as an email attachment. Build a logical expression calculator. The case sensitive keywords are:

NOT, TRUE, FALSE, AND, OR, parentheses and the period

Whitespace serves to separate alphabetic keywords. The AND operator should bind more tightly than OR. NOT is the tightest binding. A statement is terminated with a period. Parentheses allow one to overrule that precedence, such that:

FALSE OR TRUE AND FALSE OR NOT TRUE.
Equivalent to:
FALSE OR (TRUE AND FALSE) OR NOT TRUE.

Take one expression from stdin and print the result TRUE or FALSE to stdout.

Feel free but not required to use any lexical analyzer or parser generator tools appropriate to your chosen language. Bad input syntax should terminate the parsing with appropriate stderr. What happens after the attempted evaluation of the first read expression is unimportant.

1.3 Intended Audience and Reading Suggestions

The document is intended for human resources, lead developers, project managers and account managers of the Jr. Java opportunity position.

1.4 Product Scope

Logical Expression Calculator to evaluate any expression input by user as strings in the form of {NOT, TRUE, FALSE, AND, OR, parentheses and the period} and give the result as TRUE or False based in the Boolean logical evaluation.

1.5 References

Java Test.docx document

1.6 Product Perspective

This is a new, self-contained product. A simple diagram shows the major components attached in Appendix B: Analysis Models.

1.7 Product Functions

The major function is to receive expression from the user and translate it to Boolean logical expression then evaluate and generate the result.

1.8 User Classes and Characteristics

Used only one public class “LogicalExpressionCalculator”.

1.9 Operating Environment

The environment in which the software will operate is any windows operating system such as Windows7 or Windows8, and required to have Java JDK to be able to compile and run.

1.10 Design and Implementation Constraints

The only limitation is to have Java installed.

1.11 User Documentation

No documentation required.

1.12 Assumptions and Dependencies

Not applicable, the software can be shared without a problem.

2. External Interface Requirements

2.1 User Interfaces

The user will run the application and type the expression using the keyboard.

2.2 Hardware Interfaces

The computer, monitor and keyboard.

2.3 Software Interfaces

Windows, Java Runtime and Java JDK. Attached Appendix D “Compile and Run Instructions”

2.4 Communications Interfaces

Not Required.

3. System Features

The functional requirements for the product:

3.1 System Feature 1

The case sensitive keywords are: NOT, TRUE, FALSE, AND, OR, parentheses and the period.

3.2 System Feature 2

Whitespace serves to separate alphabetic keywords. The AND operator should bind more tightly than OR. NOT is the tightest binding. A statement is terminated with a period. Parentheses allow one to overrule that precedence.

4. Other Nonfunctional Requirements

Use any lexical analyzer or parser generator tools appropriate to your chosen language. Bad input syntax should terminate the parsing with appropriate stderr. What happens after the attempted evaluation of the first read expression is unimportant.

4.1 Performance Requirements

Take one expression from stdin and print the result TRUE or FALSE to stdout.

4.2 Safety Requirements

Not Required.

4.3 Security Requirements

Not Required.

4.4 Software Quality Attributes

Tested and passed QA. Result Screenshots Attached in Appendix E.

4.5 Business Rules

Not Required.

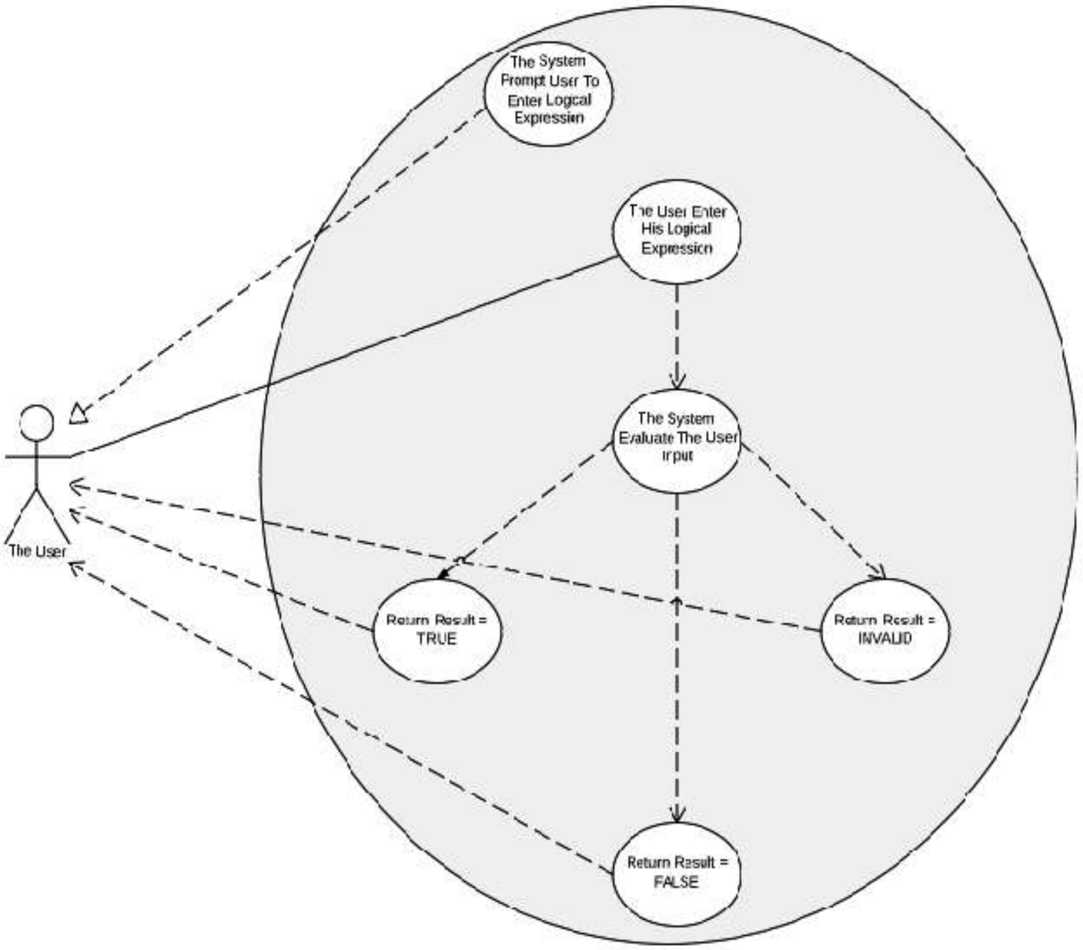
5. Other Requirements

Not Required.

Appendix A: Glossary

NOT == !
TRUE == true
FALSE == false
AND == &&
OR == ||
Parentheses == ()
Period == .

Appendix B: Analysis Models

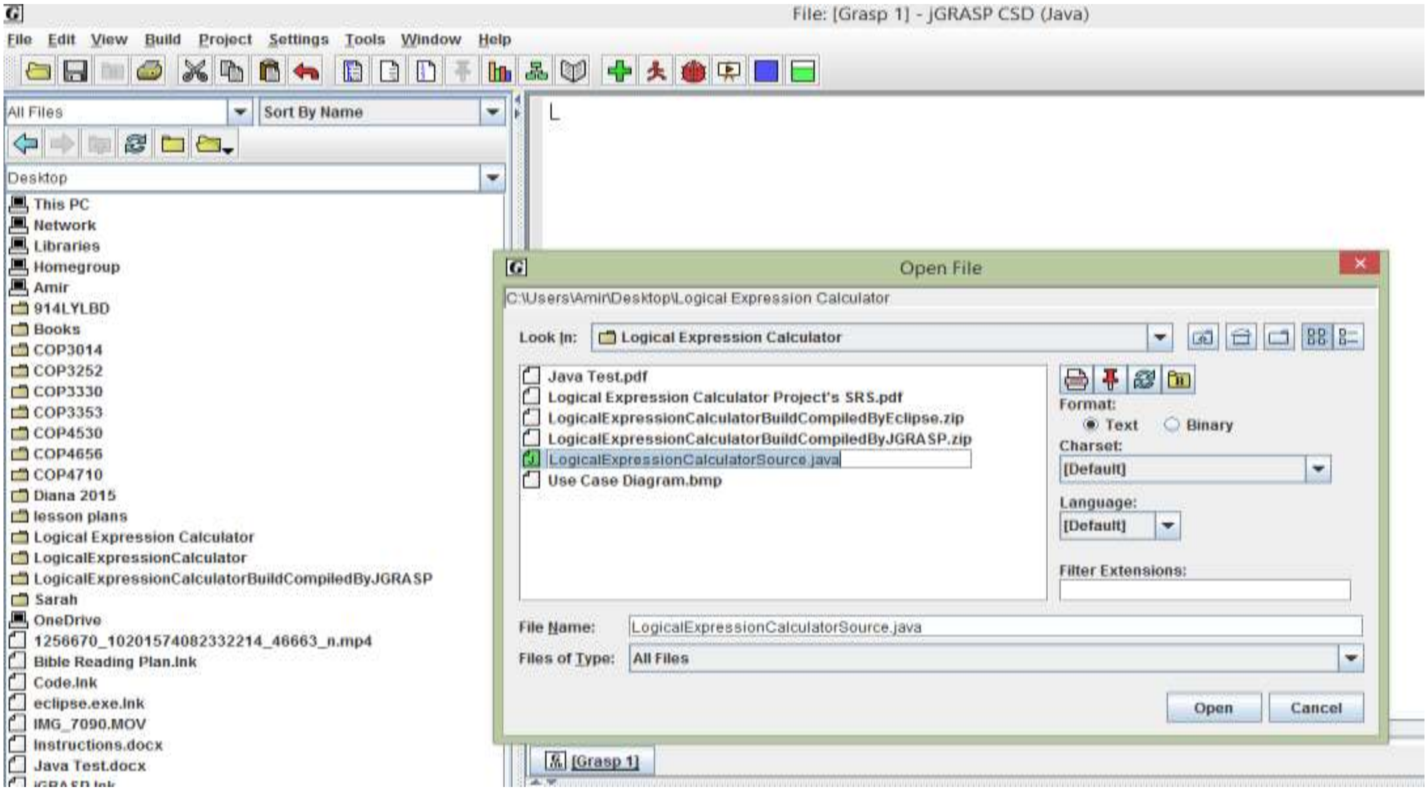

The System Prompt User To Enter Logical Expression
The User Enter His Logical Expression
The System Evaluate The User Input
Return Result = TRUE
Return Result = INVALID
Return Result = FALSE
The User

Appendix C: To Be Determined List

If require to evaluate lower case true, false, not, and, or that will be version 1.1.1
If require to evaluate other math functions such +, -, /, * that will be version 1.1.2
If require to evaluate discrete mathematics such legal expressions that will be version 1.1.3

Appendix D: Build, Compile and Run Instructions

If you are using GRASP: File -> New -> Java -> Choose LogicalExpressionCalculatorSource.java -> Ok



Open File

C:\Users\Amir\Desktop\Logical Expression Calculator

Look In: Logical Expression Calculator

- Java Test.pdf
- Logical Expression Calculator Project's SRS.pdf
- LogicalExpressionCalculatorBuildCompiledByEclipse.zip
- LogicalExpressionCalculatorBuildCompiledByJGRASP.zip
- LogicalExpressionCalculatorSource.java
- Use Case Diagram.bmp

Format: ● Text ○ Binary

Charset: [Default]

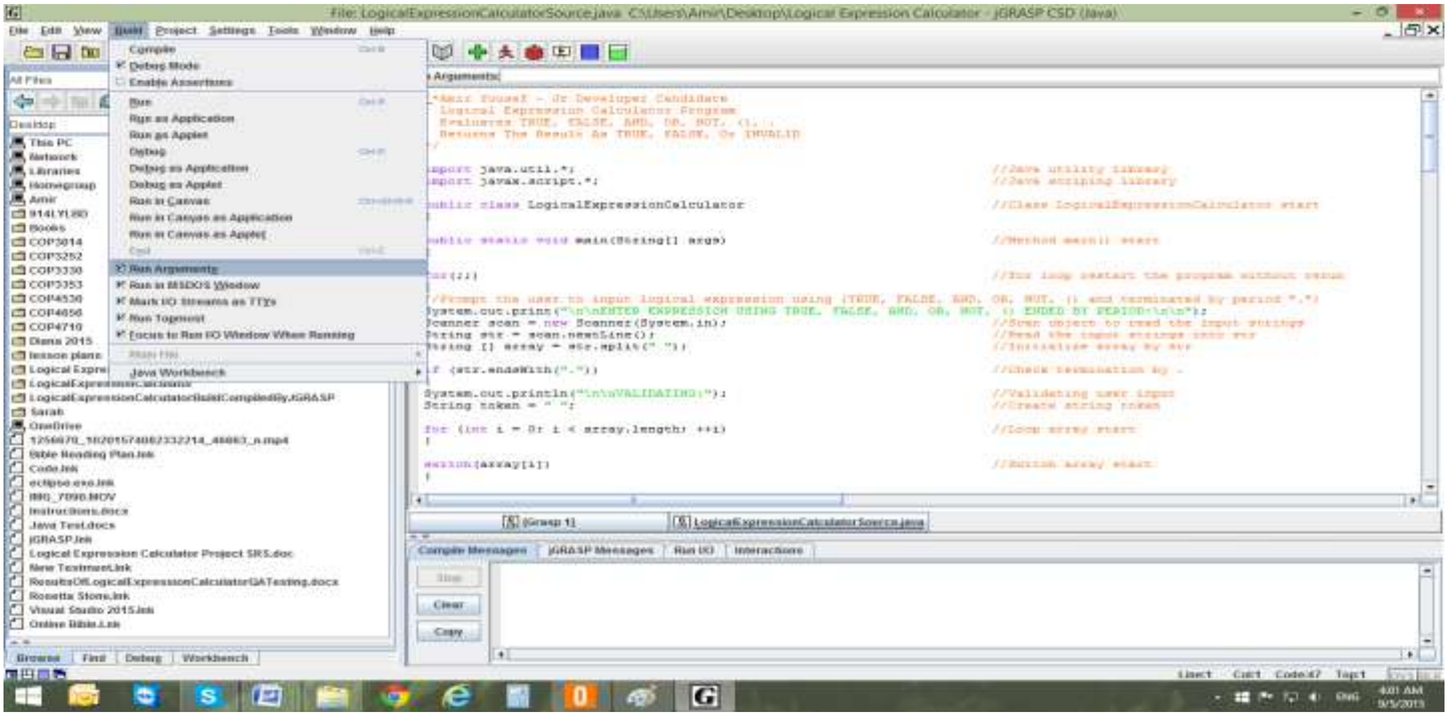
Language: [Default]

Filter Extensions:

File Name: LogicalExpressionCalculatorSource.java

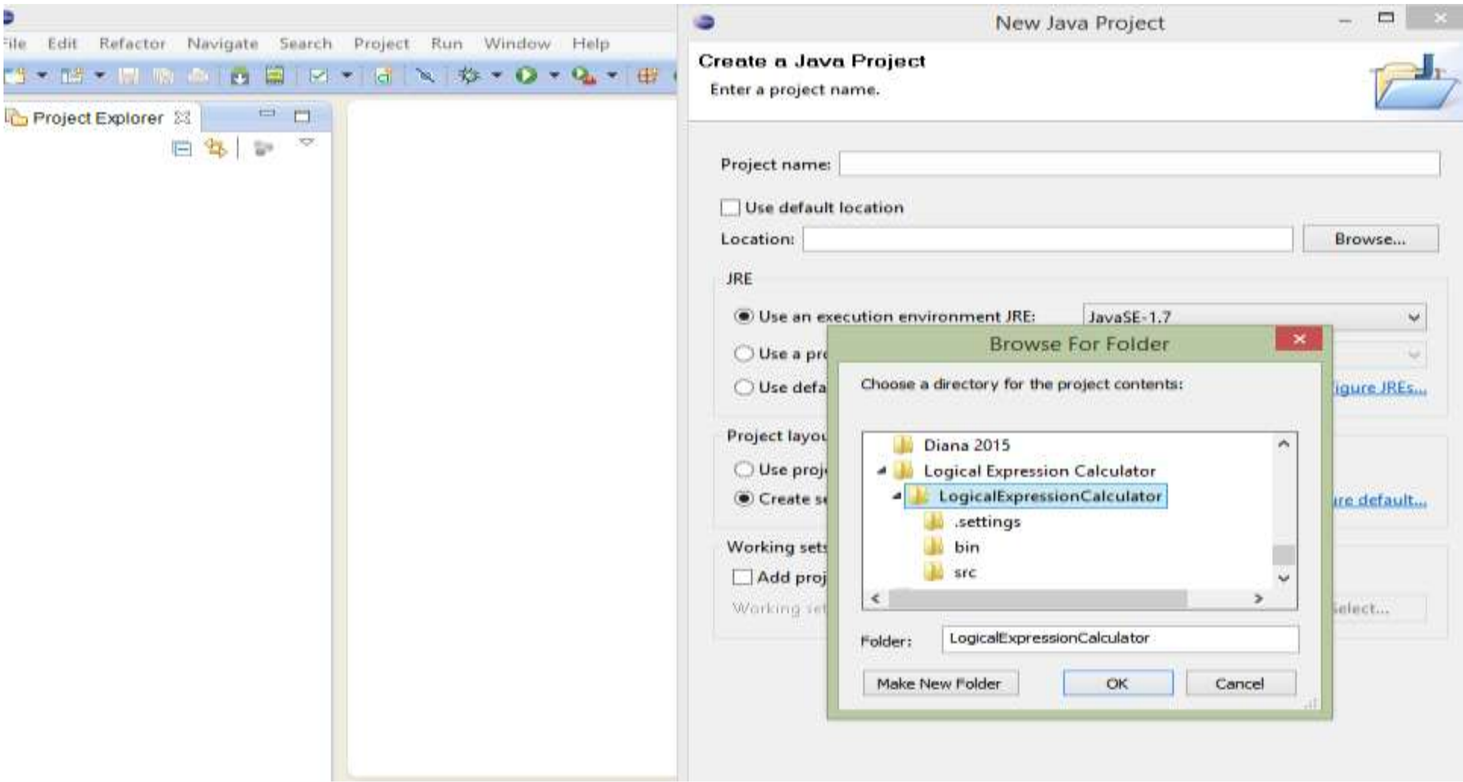
Files of Type: All Files

Click Build and select Run Arguments and Run in MSDOS to be able to see a new window to run:



```
/*Amir Yousef - Jr Developer Candidate
  Logical Expression Calculator Program
  Evaluates TRUE, FALSE, AND, OR, NOT, (), )
  Returns The Result As TRUE, FALSE, Or INVALID
*/

import java.util.*;                                      //Java utility library
import javax.script.*;                                   //Java scripting library

public class LogicalExpressionCalculator                 //Class LogicalExpressionCalculator start
{

public static void main(String[] args)                   //Method main() start
{

for(;;)                                                  //For loop restart the program without rerun
{
//Prompt the user to input logical expression using (TRUE, FALSE, AND, OR, NOT, () and terminated by period ".")
System.out.print("\n\nENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:\n\n");
Scanner scan = new Scanner(System.in);                   //Scan object to read the input strings
String str = scan.nextLine();                            //Read the input strings into str
String [] array = str.split(" ");                        //Initialize array by str

if (str.endsWith("."))                                   //Check termination by .
{
System.out.println("\n\nVALIDATING:");                   //Validating user input
String token = " ";                                      //Create string token

for (int i = 0; i < array.length; ++i)                   //Loop array start
{

switch(array[i])                                         //Switch array start
{
```

If you are using Eclipse: File -> New Java Project -> Browse -> Choose Folder Location -> Ok -> Finish



Choose a directory for the project contents:

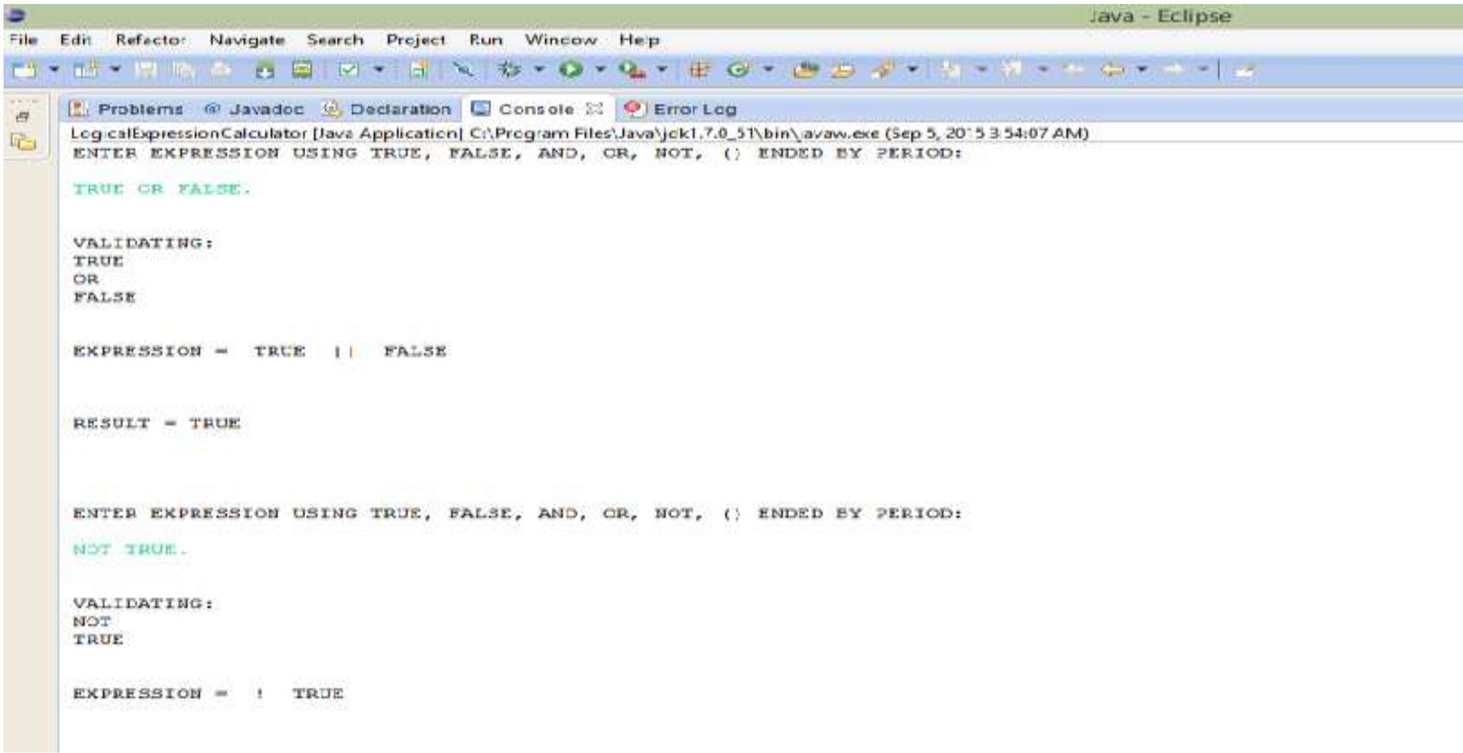
- Diana 2015
- Logical Expression Calculator
  - LogicalExpressionCalculator
    - .settings
    - bin
    - src

Folder: LogicalExpressionCalculator

☐ Use default location

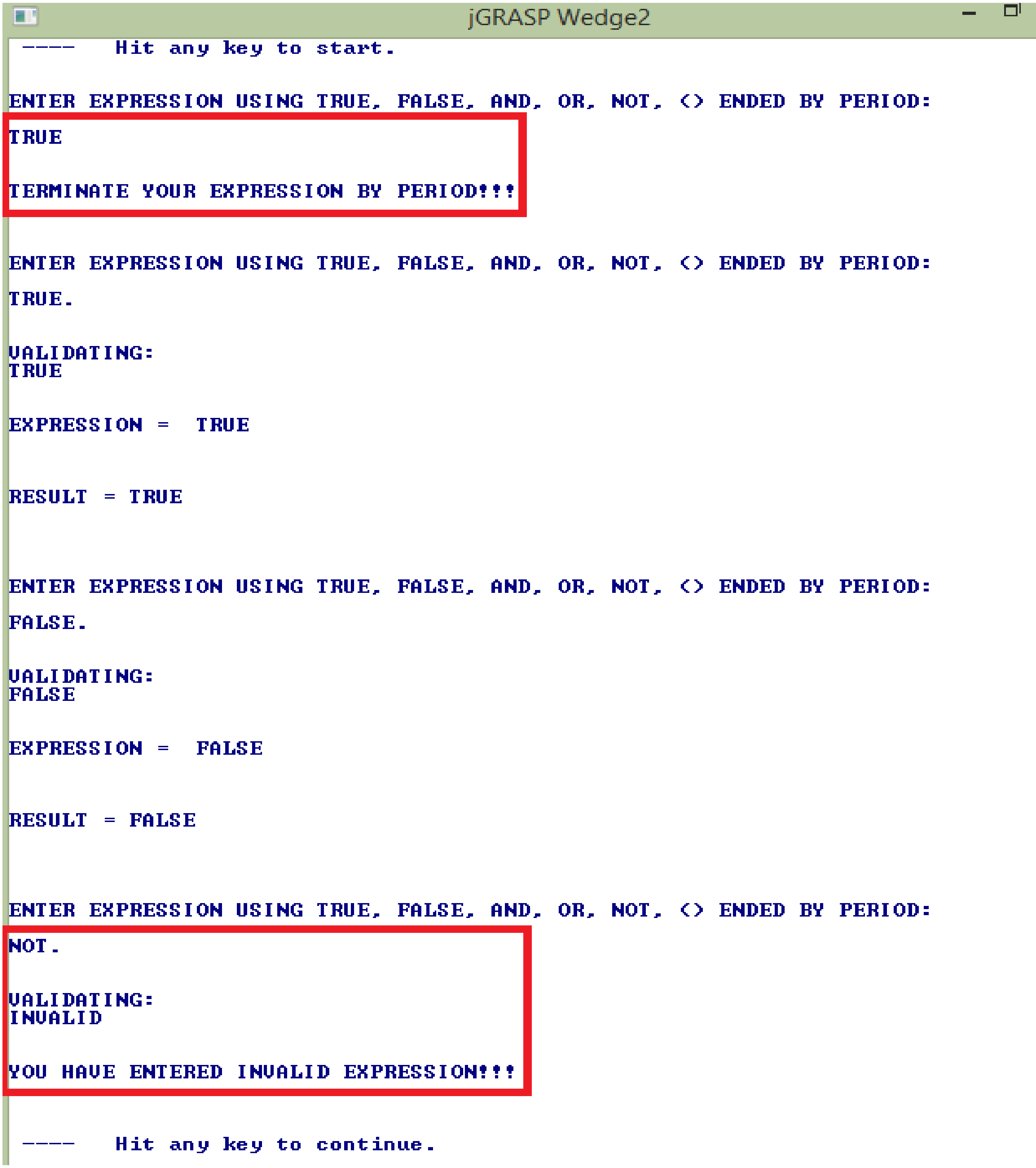
● Use an execution environment JRE: JavaSE-1.7

Click Run to execute: try ex: TRUE OR FALSE. Press enter

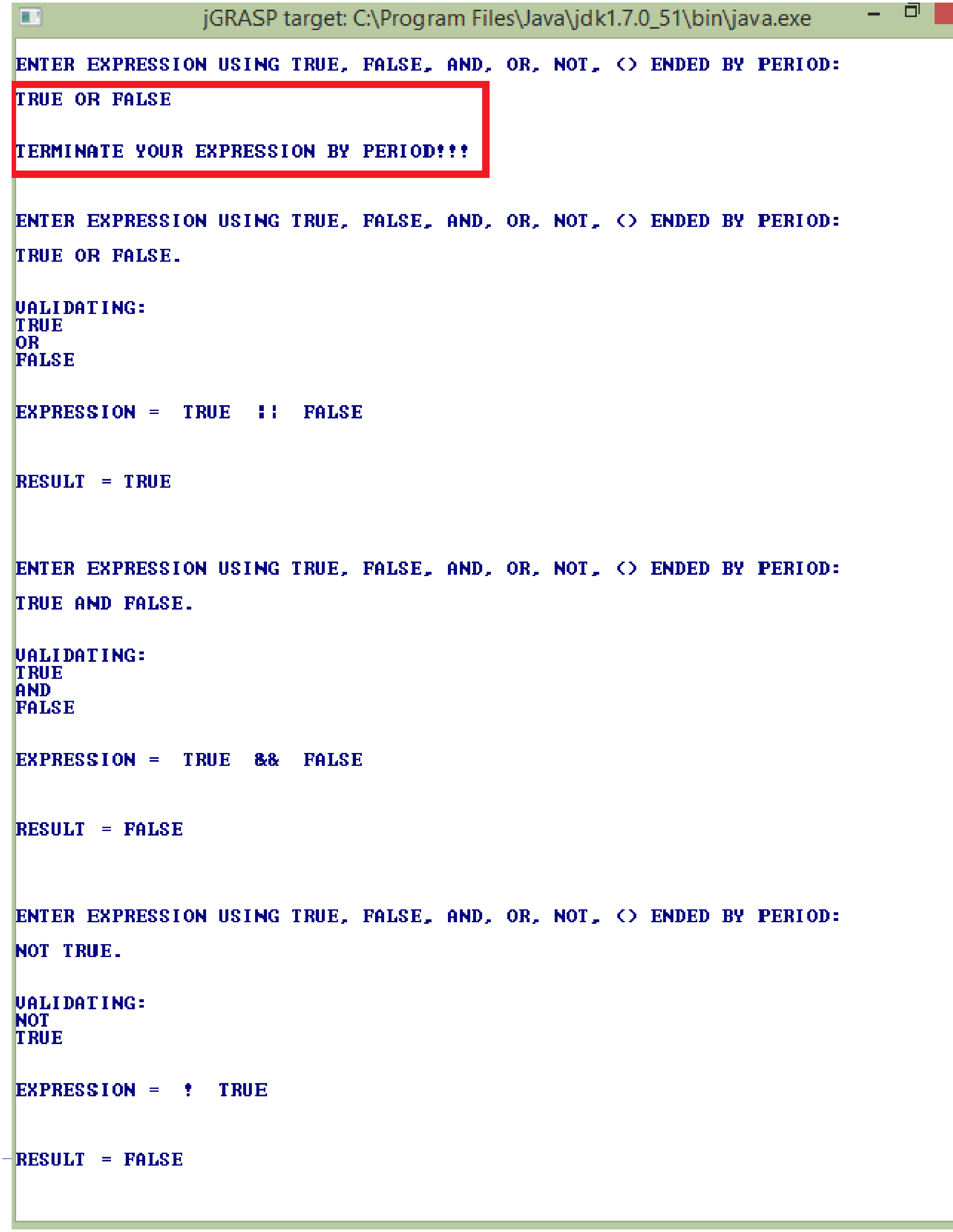


```
LogicalExpressionCalculator [Java Application] C:\Program Files\Java\jdk1.7.0_51\bin\javaw.exe (Sep 5, 2015 3:54:07 AM)
ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:

TRUE OR FALSE.


VALIDATING:
TRUE
OR
FALSE


EXPRESSION =  TRUE  ||  FALSE


RESULT = TRUE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:

NOT TRUE.


VALIDATING:
NOT
TRUE


EXPRESSION =  !  TRUE
```

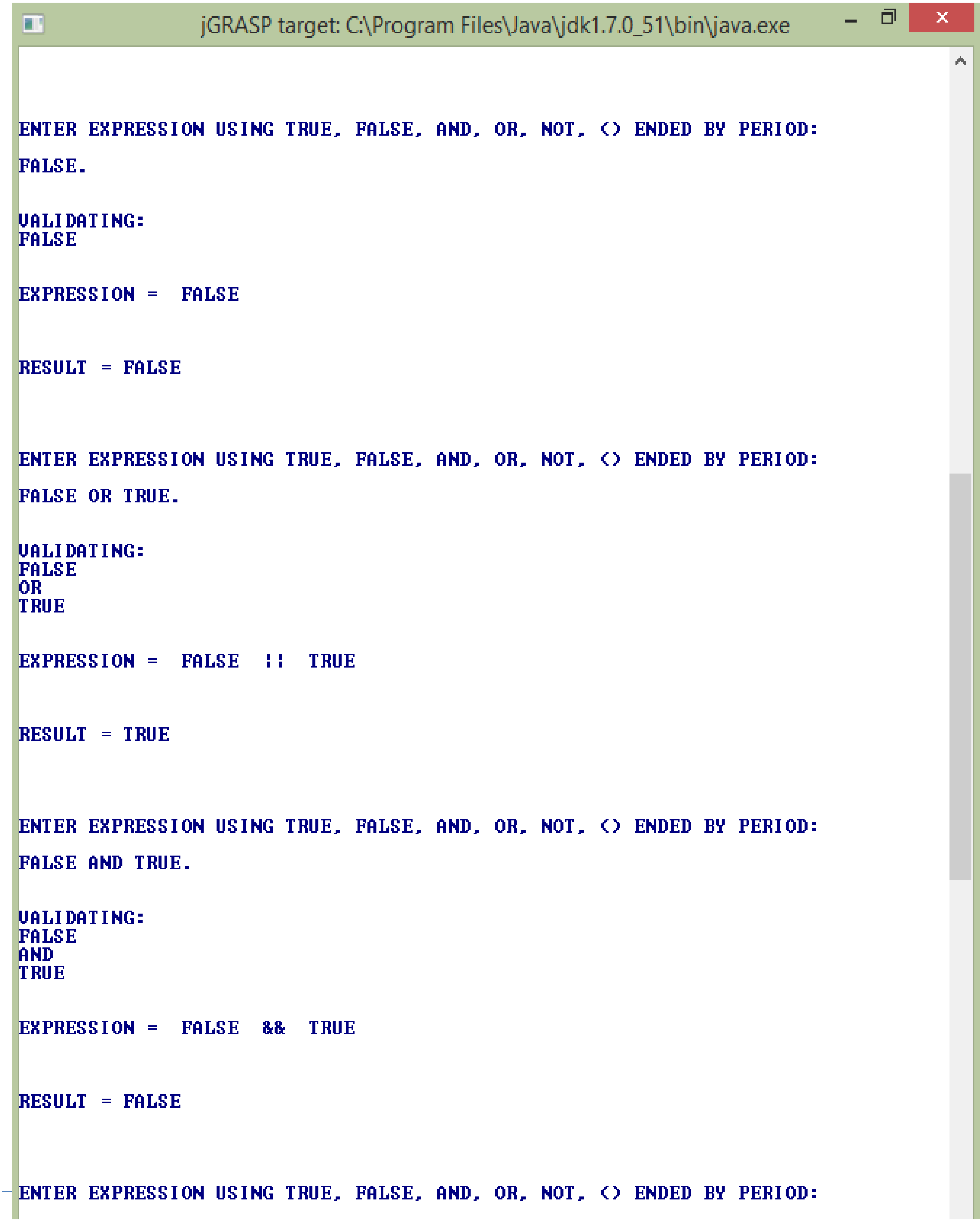
Appendix E: JGRASP QA Testing Result Screen Shots



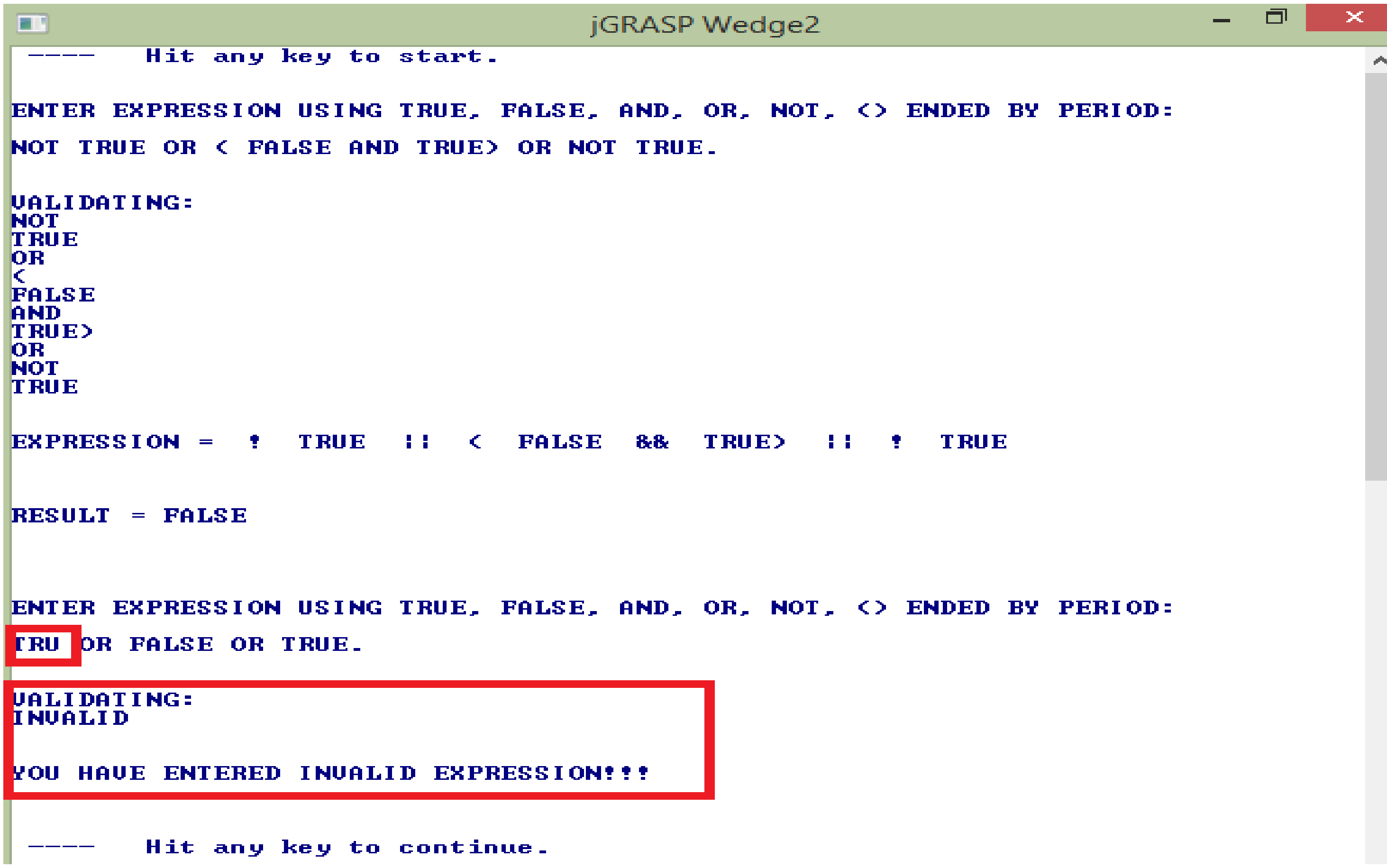
```
 ----   Hit any key to start.

ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, <> ENDED BY PERIOD:
TRUE

TERMINATE YOUR EXPRESSION BY PERIOD!!!


ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, <> ENDED BY PERIOD:
TRUE.

VALIDATING:
TRUE

EXPRESSION =  TRUE


RESULT = TRUE


ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, <> ENDED BY PERIOD:
FALSE.

VALIDATING:
FALSE

EXPRESSION =  FALSE


RESULT = FALSE


ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, <> ENDED BY PERIOD:
NOT.

VALIDATING:
INVALID

YOU HAVE ENTERED INVALID EXPRESSION!!!


 ----   Hit any key to continue.
```


jGRASP target: C:\Program Files\Java\jdk1.7.0_51\bin\java.exe

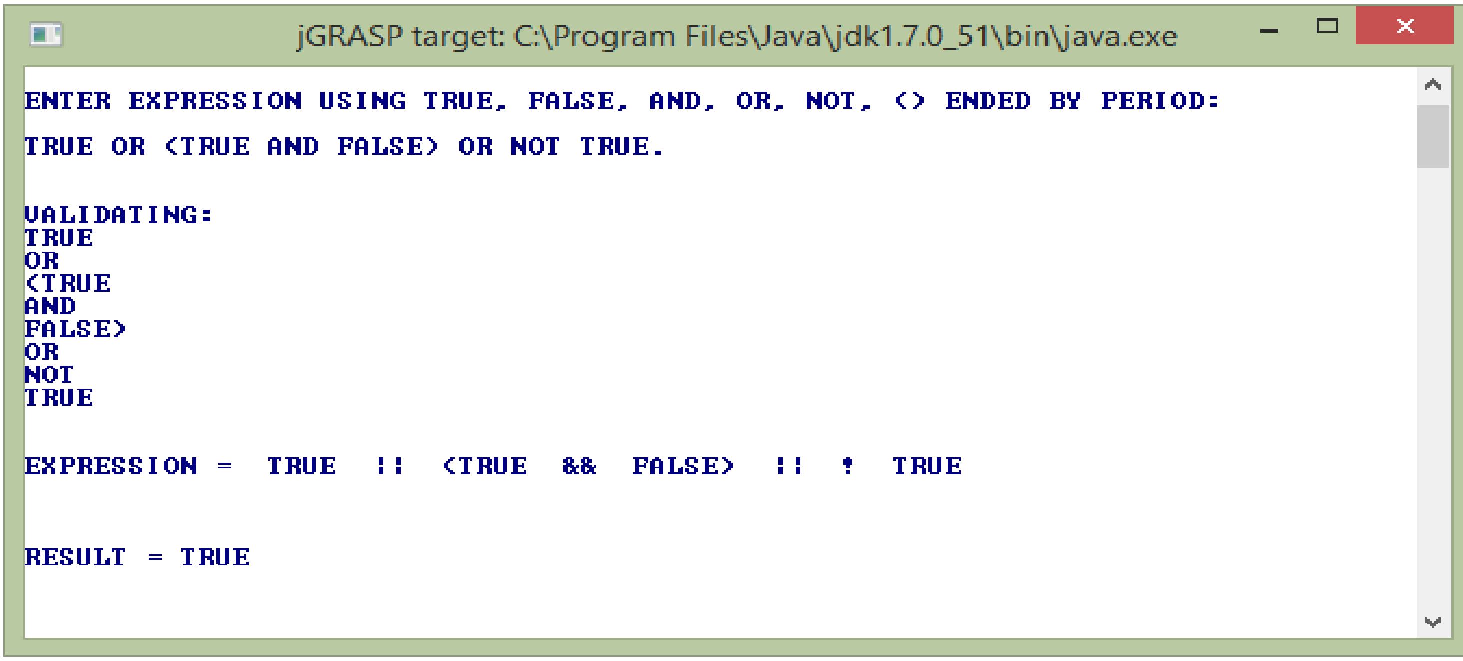
```
ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
TRUE OR FALSE

TERMINATE YOUR EXPRESSION BY PERIOD!!!


ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
TRUE OR FALSE.

VALIDATING:
TRUE
OR
FALSE

EXPRESSION =  TRUE  ||  FALSE


RESULT  = TRUE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
TRUE AND FALSE.

VALIDATING:
TRUE
AND
FALSE

EXPRESSION =  TRUE  &&  FALSE


RESULT  = FALSE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
NOT TRUE.

VALIDATING:
NOT
TRUE

EXPRESSION =  !  TRUE


RESULT  = FALSE
```


jGRASP target: C:\Program Files\Java\jdk1.7.0_51\bin\java.exe

```
ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:

FALSE.

VALIDATING:
FALSE

EXPRESSION =  FALSE


RESULT  = FALSE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:

FALSE OR TRUE.

VALIDATING:
FALSE
OR
TRUE

EXPRESSION =  FALSE  ||  TRUE


RESULT  = TRUE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:

FALSE AND TRUE.

VALIDATING:
FALSE
AND
TRUE

EXPRESSION =  FALSE  &&  TRUE


RESULT  = FALSE



ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
```



```
 ----   Hit any key to start.

ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
NOT TRUE OR ( FALSE AND TRUE) OR NOT TRUE.

VALIDATING:
NOT
TRUE
OR
(
FALSE
AND
TRUE)
OR
NOT
TRUE

EXPRESSION =  !  TRUE  ||  (  FALSE  &&  TRUE)  ||  !  TRUE


RESULT  = FALSE


ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
TRU OR FALSE OR TRUE.

VALIDATING:
INVALID

YOU HAVE ENTERED INVALID EXPRESSION!!!

 ----   Hit any key to continue.
```


jGRASP target: C:\Program Files\Java\jdk1.7.0_51\bin\java.exe

```
ENTER EXPRESSION USING TRUE, FALSE, AND, OR, NOT, () ENDED BY PERIOD:
TRUE OR (TRUE AND FALSE) OR NOT TRUE.

VALIDATING:
TRUE
OR
(TRUE
AND
FALSE)
OR
NOT
TRUE

EXPRESSION =  TRUE  ||  (TRUE  &&  FALSE)  ||  !  TRUE


RESULT  = TRUE
```