

Report on Input Validation for a Telephone Directory

CSE 5382 – 001

Abstract

The goal of this assignment is to produce a program that validates its input using regular expressions. The benefits of using whitelisting technique is also discussed in this report.

Adishree Annabathula

Table of Contents

1. Assignment description ……………………………………………….……………………3
2. Tools used …….……………………………………………..……….…………....…….....3
3. Whitelisting vs Blacklisting ……………………………………………………………......3
4. Regular Expressions …………………………………………………………………….....4
5. Outputs …………………………………………………………………………………......6
6. References …………………………………………………………….……………..…....15
7. Assignment description
8. Write a command-driven program that serves as a telephone directory.
9. It should have the following commands:
   1. ADD <Person> <Telephone#>
   2. DEL <Person>
   3. DEL <Telephone#>
   4. LIST
10. The program should reject invalid data.
11. It should display exit code 0 in case of valid input and 1 otherwise.
12. Tools used
13. **Eclipse**

It is an integrated development environment consisting of a workspace and an extensible plug-in system. The Java IDE is used for this assignment.

Version used: Neon (v4.6.2)

1. **MySQL Workbench**

It is an integrated tools environment that can be used for Database design and SQL development. AWS is used for this assignment to connect to the database.

Version used: v6.3.9

1. Whitelisting vs Blacklisting

* Input validation can be defined as correctly testing the user input supplied by a program. The user input can be from various sources like an end user, command-line parameters, data from a database, environment variables or another program.
* Whitelisting is referred to as accepting well-formed data whereas blacklisting is referred to as rejecting bad data.
* Whitelisting is preferably the best approach to input validation.
* It is not possible to test the input against all types of negative conditions. [1] Data can be valid only for a few conditions. Hence it is always better to test it against those conditions to ensure maximum safety from attackers.

1. Regular Expressions

Regular expressions make input validation easy and secure. Whitelists are expressed as regular expressions.

Several patterns for the given list of acceptable conditions are written and tested against the input data sent to the program. If matched, then it is considered as a valid input; otherwise rejected. [2]

The BNF forms of the regular expressions for the given input are listed as follows:

<Person>::=<first-name> <last-name>

| <last-name>, <first-name> <middle-name>

| <last-name>, <first-name> <middle-initial>.

<first-name>::=<name>

<middle-name>::=<middle-initial><name>

<last-name>::=<name> | <name><symbol> | <symbol> | <alpha>

<name>::=<name><alpha> | <alpha>

<symbol>::=’ | - | .

<alpha>::=A | B | C… Z | a | b…z

<Telephone#>::=<Number>

<Number>::=<Symbol><Number> | <Number> <Symbol>

| <Number><Digit> | <Digit>

<Symbol>::=+ | ( | - | ) | .

<Digit>::=0 | 1 | 2….9

(underlined part indicates optional fields)

For the given program, the following are some cases of valid input and the corresponding patterns of regular expressions.

**Valid Names**

* Bruce Schneier

Format: <Person>::=<first-name> <last-name>

Matching pattern:

^[A-Z]{1}[a-z]{1,}+\\s+[A-Z]{1}[a-z]{1,}$

* Schneier, Bruce

Format: <Person>::=<last-name>, <first-name>

Matching pattern:

^[A-Z]{1}[a-z]{1,}+\\,+\\s+[A-Z]{1}[a-z]{1,}$

* Schneier, Bruce Wayne

Format: <Person>::=<last-name>, <first-name> <middle-name>

Matching pattern:

^[A-Z]{1}[a-z]{1,}+\\,+\\s+[A-Z]{1}[a-z]{1,}+\\s++[A-Z]{1}[a-z]{1,}$

* O’Malley, John F.

Format: <Person>::=<last-name>, <first-name> <middle-initial>.

Matching pattern:

^[A-Z]{1}[a-z]{0,}+\\’+[A-Z]{1}[a-z]{0,}+\\,+\\s+[A-Z]{1}[a-z]{1,}+\\s++[A-Z]{1}+\\.$

* John O’Malley-Smith

Format: <Person>::=<first-name> <last-name>

Matching pattern:

^[A-Z]{1}[a-z]{1,}+\\s+[A-Z]{1}[a-z]{0,}+\\’+[A-Z]{1}[a-z]{0,}+\\-+

[A-Z]{1}[a-z]{0,}$

* Cher

Format: <Person>::=<first-name>

Matching pattern:

^[A-Z]{1}[a-z]{1,}$

**Valid Telephone numbers**

* 12345

Matching pattern:

^[0-9]{5}$

* (703)111-2121

Matching pattern:

^\\+{0,1}+[0-9]{0,3}+\\({1}+[1-9]{1}+[0-9]{2}+\\){1}+[0-9]{3}

+\\-{1}+[0-9]{4}$

* 123-1234

Matching pattern:

^[0-9]{3}+\\-{1}+[0-9]{4}$"

* +1(703)111-2121

Matching pattern:

^\\+{1}+[0-9]{1,3}+\\s+\\({1}+[0-9]{2}+\\){1}+\\s+[0-9]{3}+\\-{1}+[0-9]{4}$

* +32 (21) 212-2324

Matching pattern:

^\\+{0,1}+[0-9]{0,3}+\\({1}+[1-9]{1}+[0-9]{2}+\\){1}+[0-9]{3}

+\\-{1}+[0-9]{4}$

* 1(703)123-1234

Matching pattern:

^\\+{0,1}+[0-9]{0,3}+\\({1}+[1-9]{1}+[0-9]{2}+\\){1}+[0-9]{3}

+\\-{1}+[0-9]{4}$

* 011 701 111 1234

Matching pattern:

^[0-9]{3}+\\s+[0-9]{3}+\\s+[0-9]{3}+\\s+[0-9]{4}$

* 12345.12345

Matching pattern:

^[0-9]{5}+\\.+[0-9]{5}$

* 011 1 703 111 1234

Matching pattern:

^[0-9]{3}+\\s+[0-9]{1}+\\s+[0-9]{3}+\\s+[0-9]{3}+\\s+[0-9]{4}$

1. Outputs

The program is executed on the command-line as follows:

<executable-name> <mode> <arguments>

<mode> can be ADD or DEL or LIST.

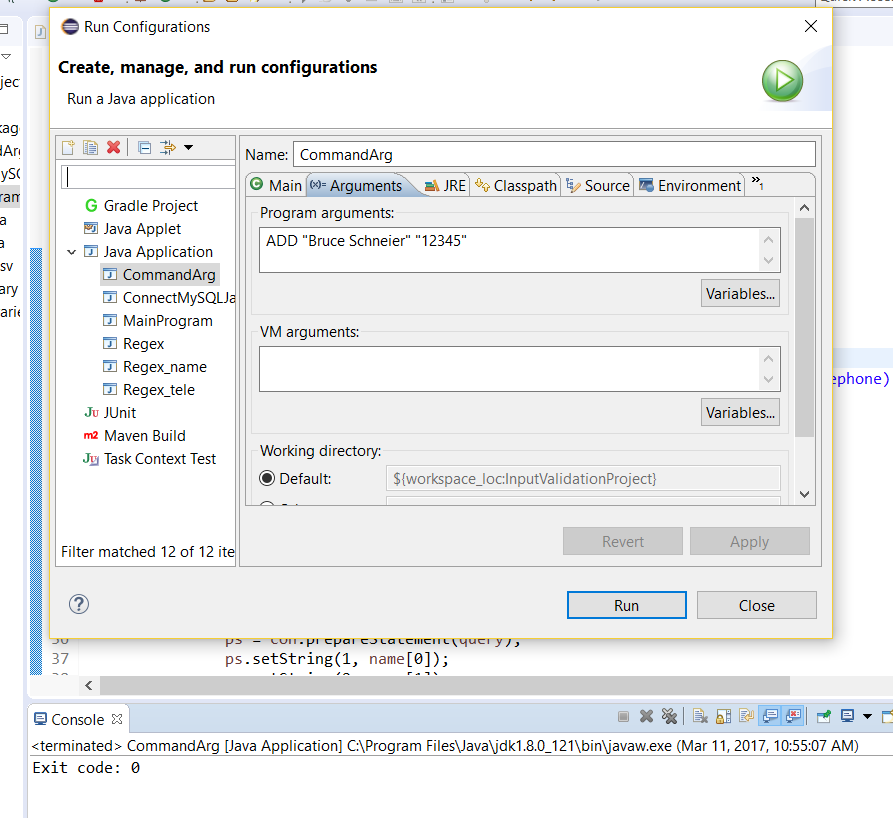
ADD is followed by both person name and telephone number. It is used to insert the person’s name and telephone number into the database.

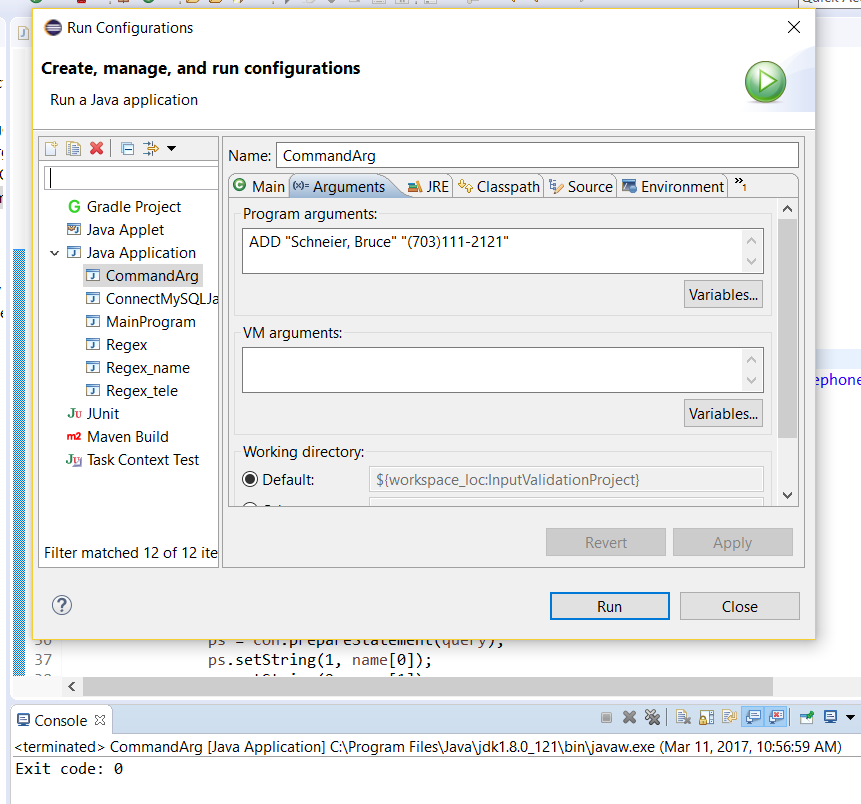
DEL is followed by either the person name or the telephone number. It is used to delete the person’s record in the database.

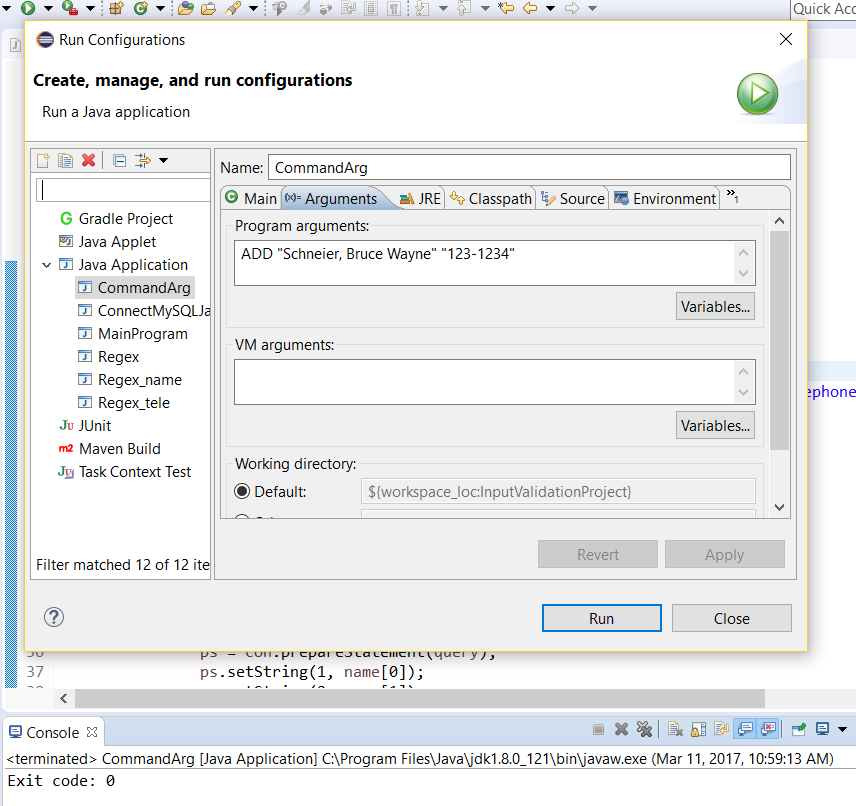
LIST retrieves the information in the database.

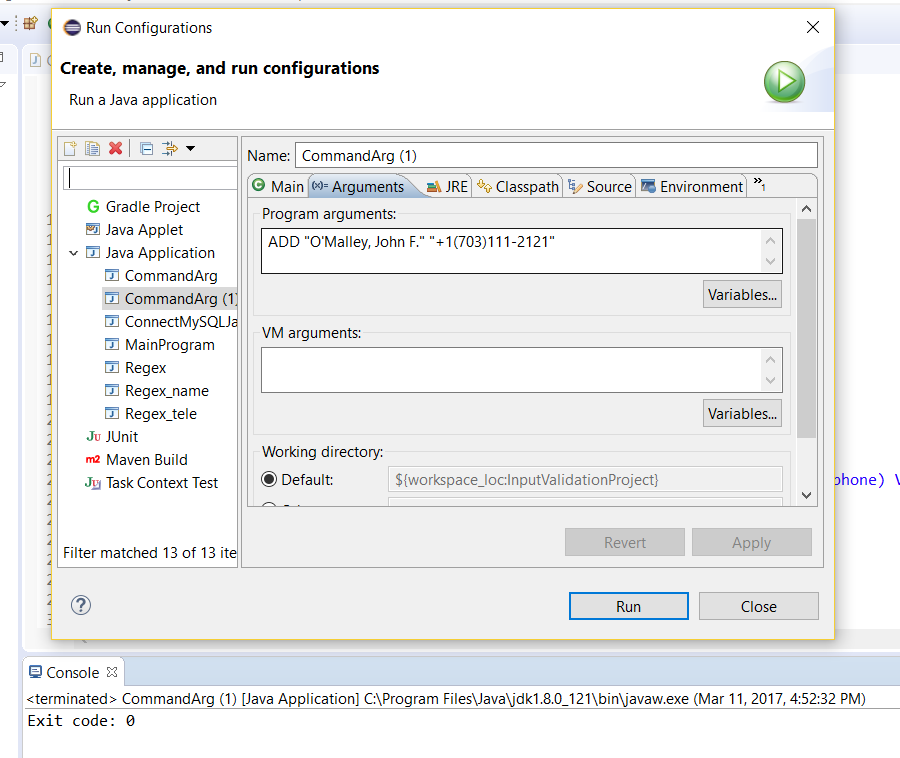
<arguments> can be either the person’s name or telephone number.

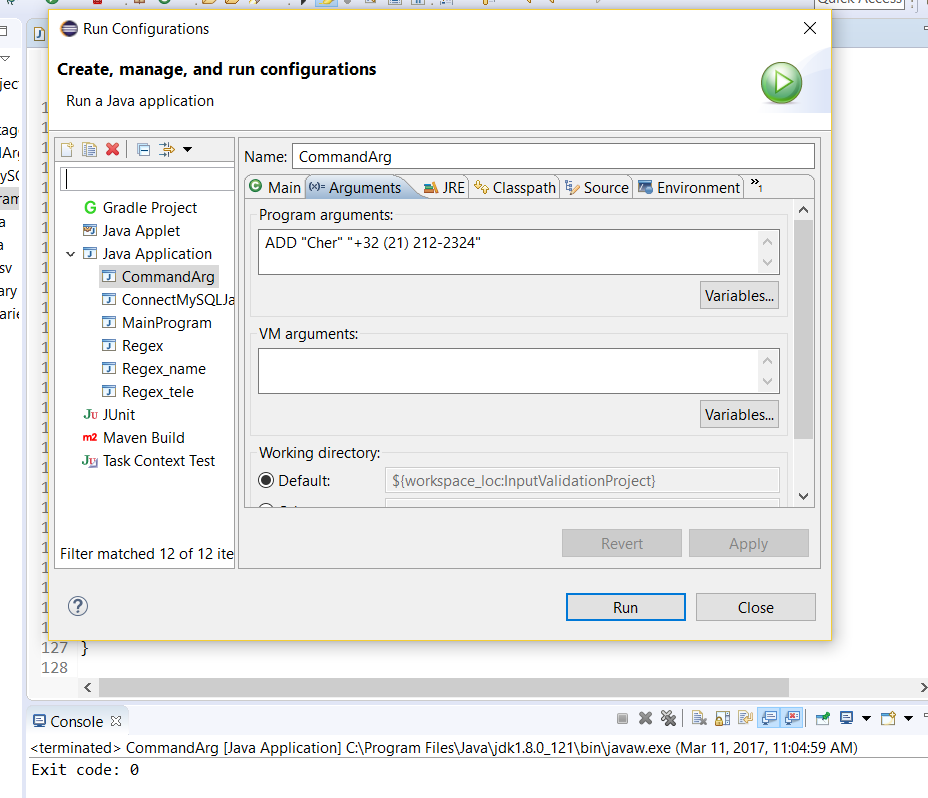
The following are screenshots of the ADD mode for various inputs.

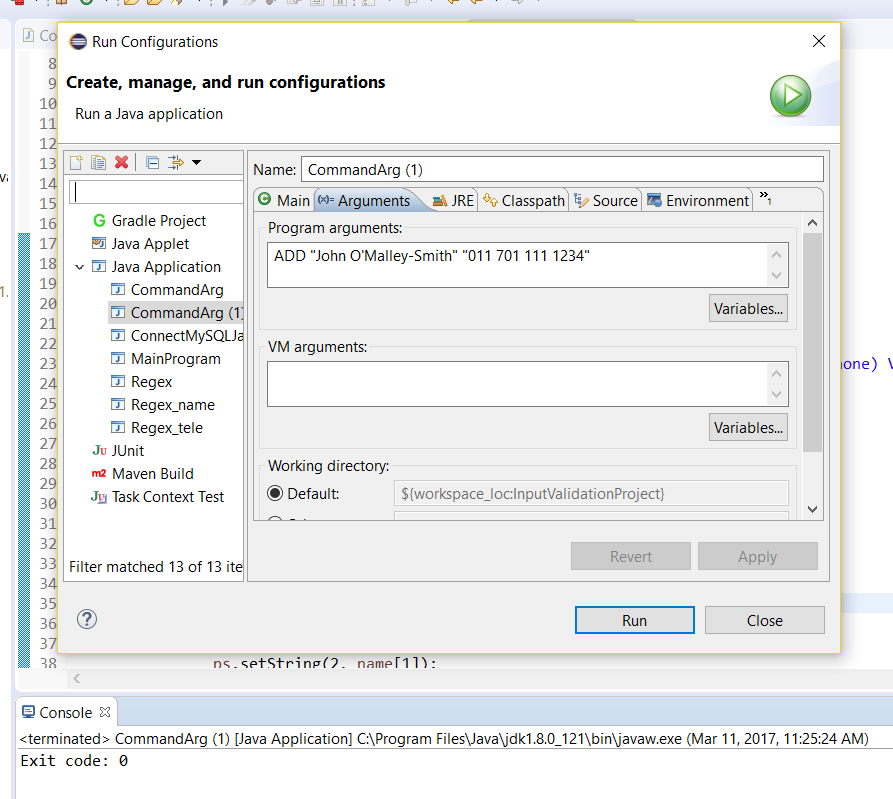
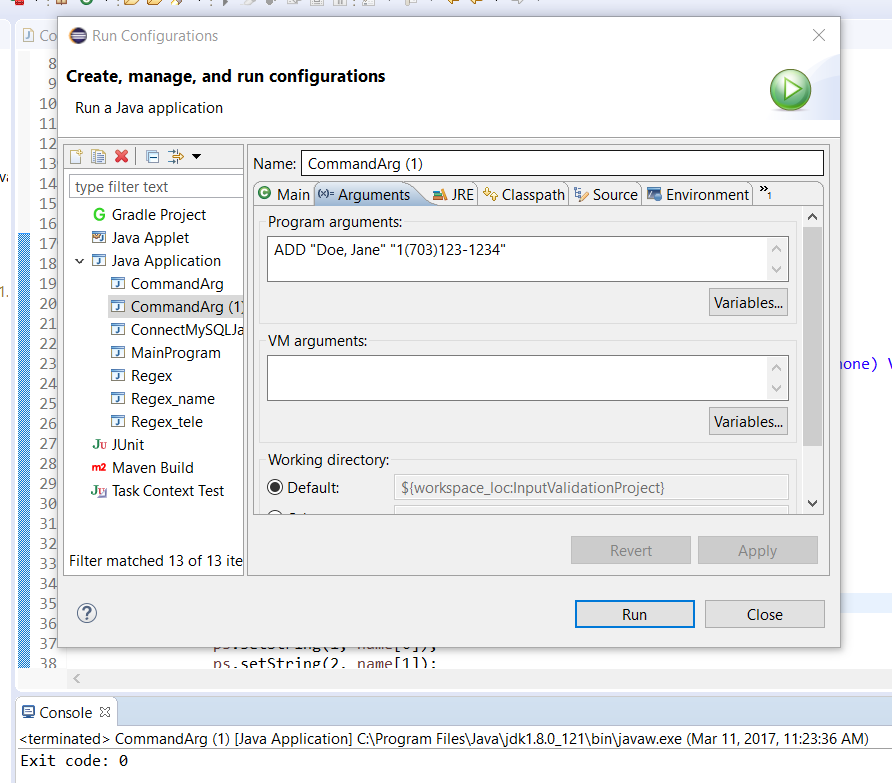
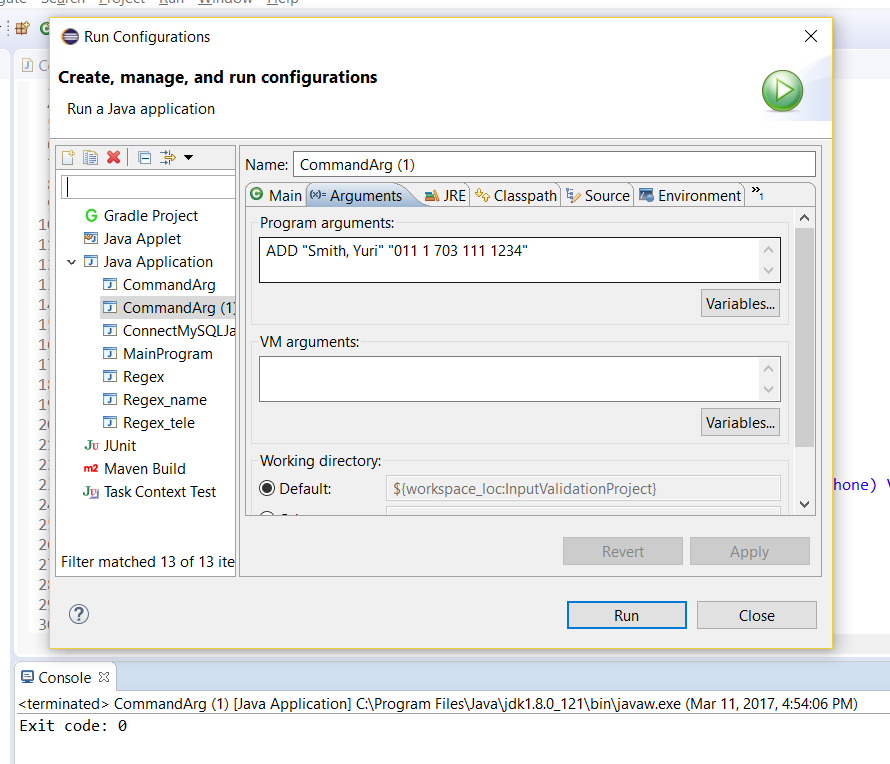
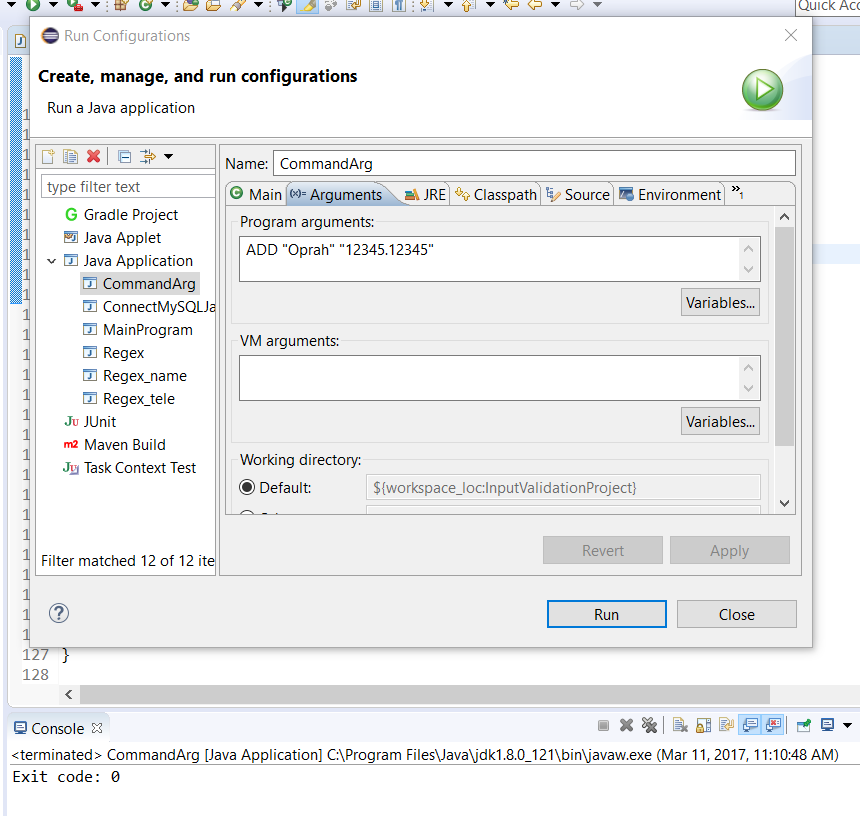


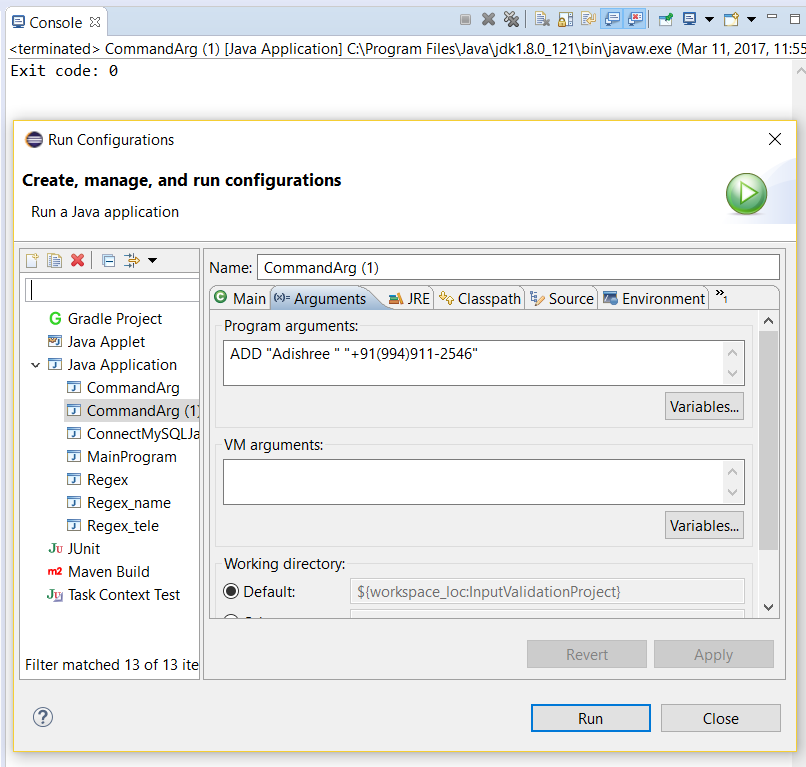




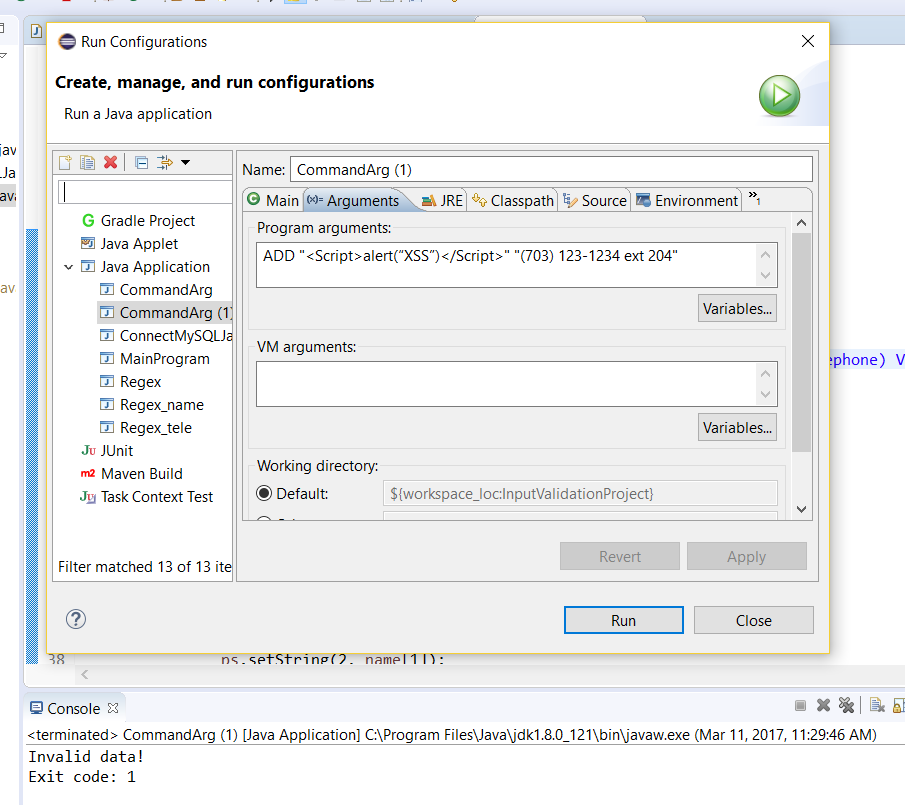






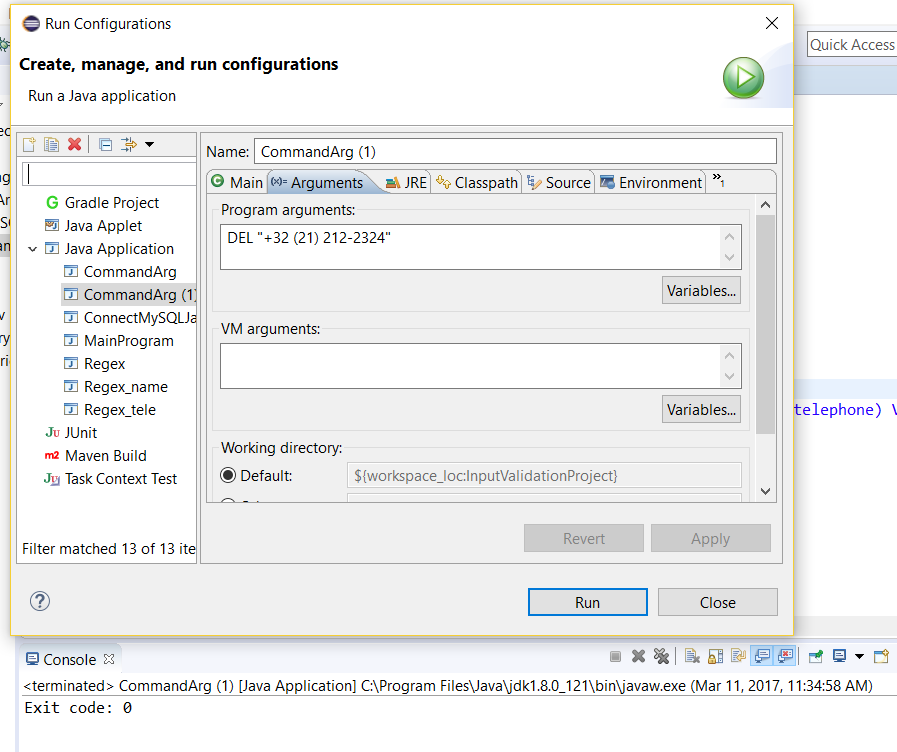
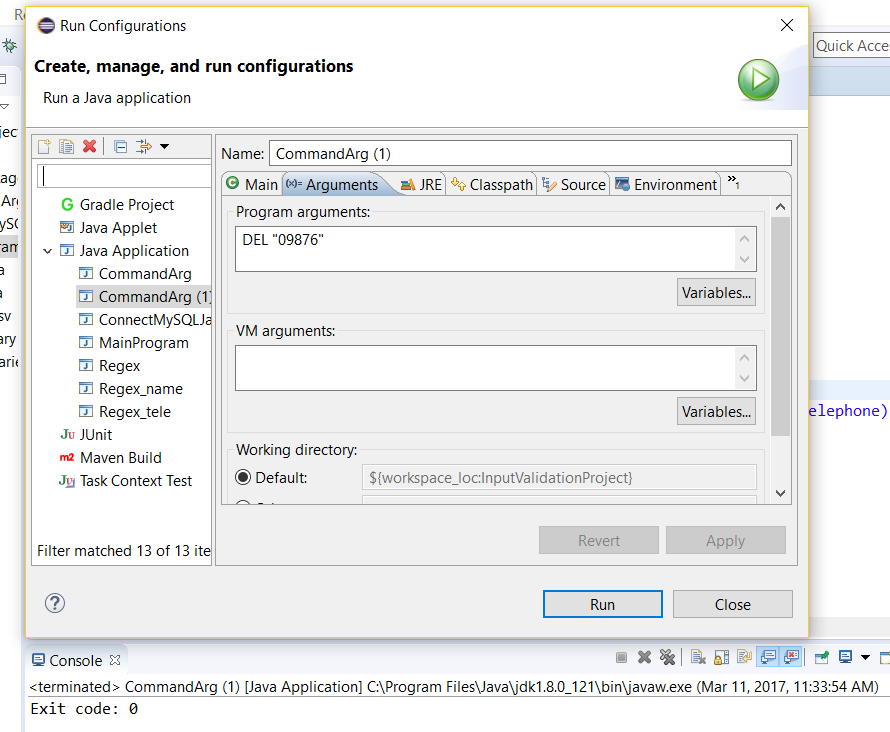


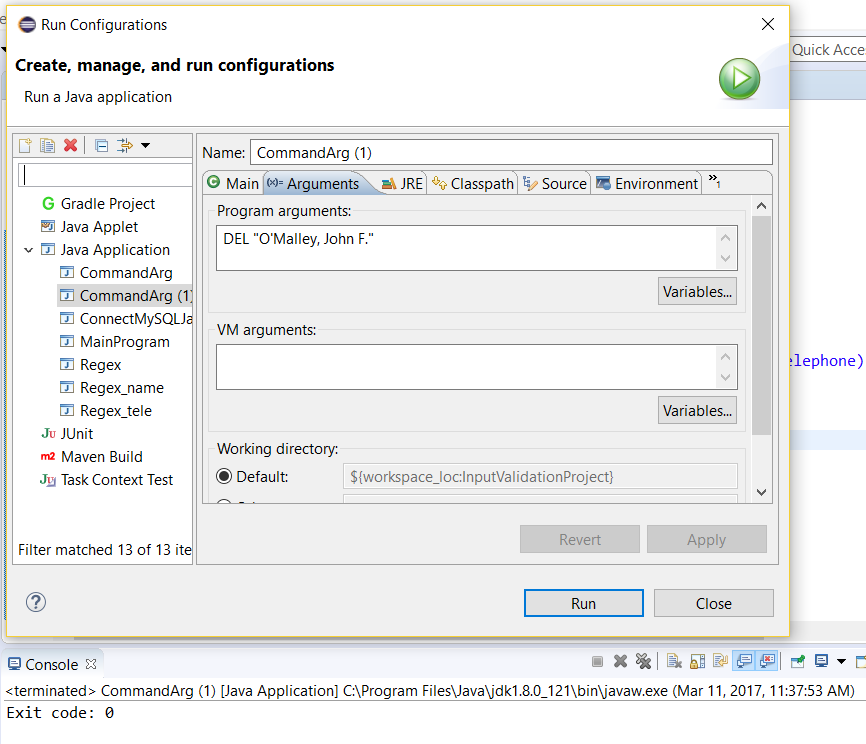
The telephone number has been tested against an international number.

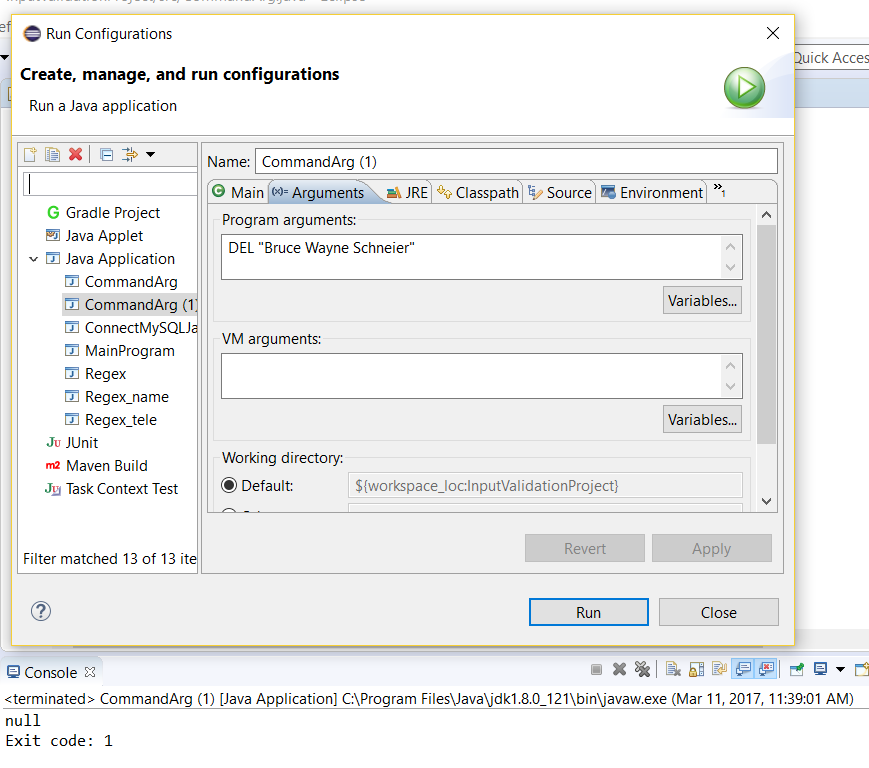


A case for invalid data

The following are screenshots of the DEL mode for various inputs.

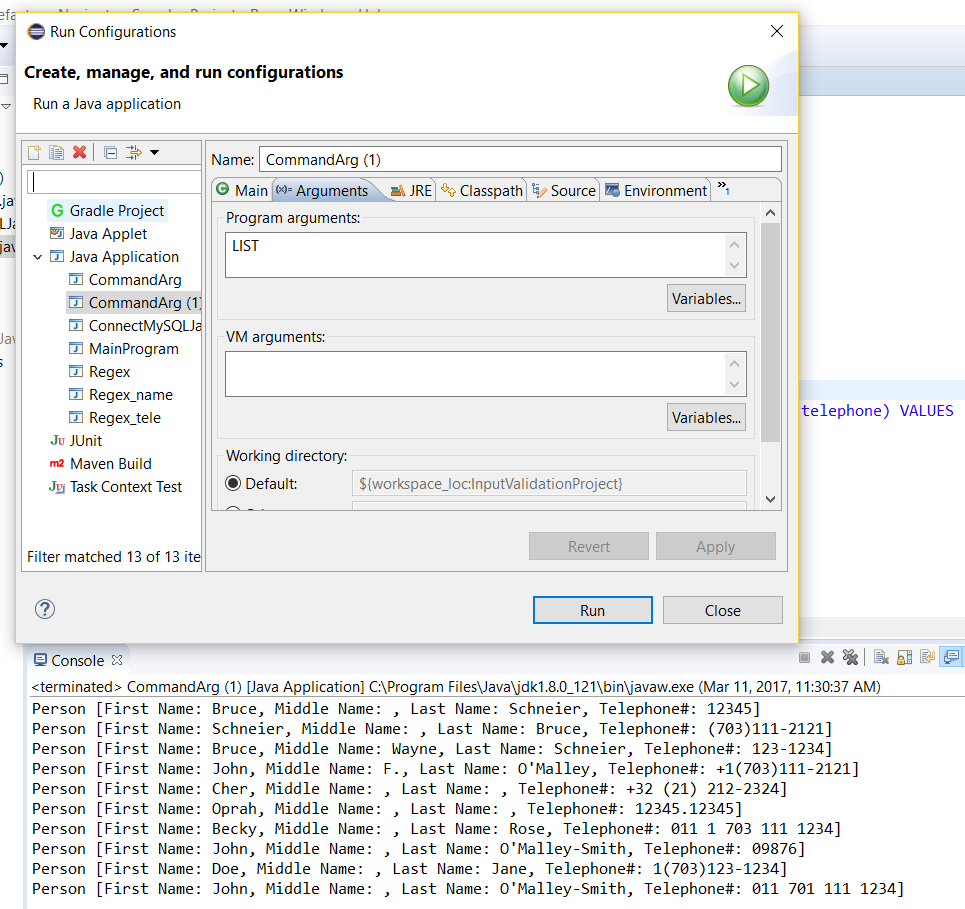




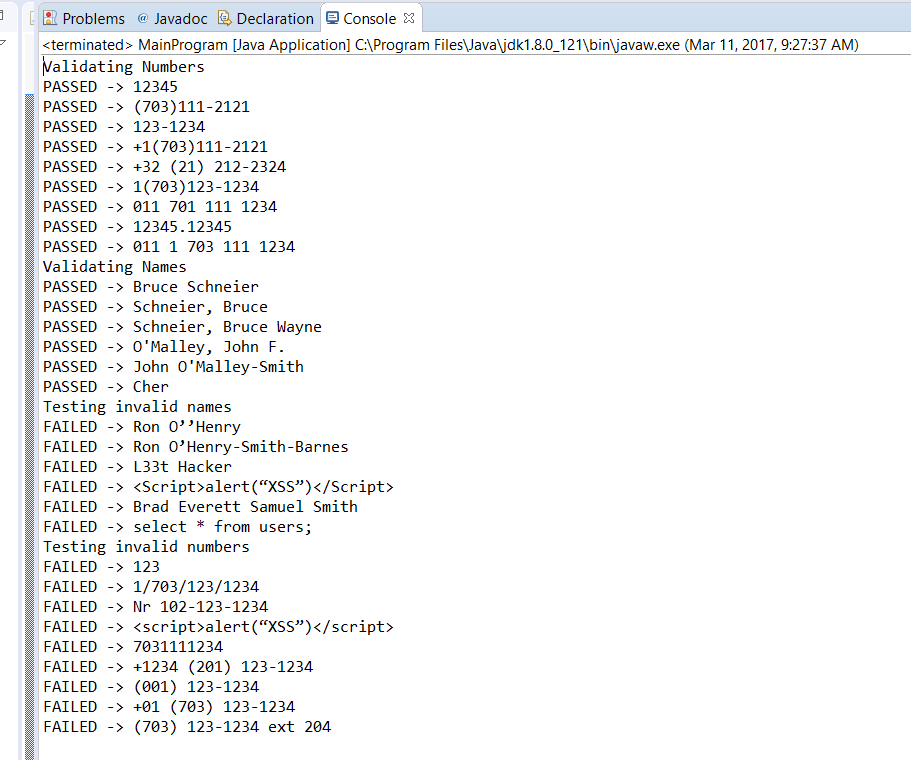


A case for invalid data

The following is a screenshot of the LIST mode.



A separate program has been written to test if the data is VALID or not without using the modes. It gives the following output.



1. References
2. <http://www.testingsecurity.com/whitelists_vs_blacklists>
3. <https://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/validation-tools-regex.html>