Homework 11:- Input Validation Report

Name:- Aparna Krishna Bhat

ID:- 1001255079

Description of how your code works

When a user types something on the command line, this program performs all of the input validation needed. It uses the corresponding regex pattern to verify the person's name and phone number and returns true or false. Matching is done with the Pattern and Matcher classes. every pattern of regex Pattern returns the Pattern instance after compiling the specified regex.

It constructs a matcher that compares the input to the pattern. Validate and match the name with the regex pattern with the NamePhoneValidator() process. Validating phone numbers is done by the NamePhoneValidator() process. We also verify that the country code in the phone number is right, in addition to the pattern match. In order to do so, a text file containing 248 country codes is created. It is opened and read one at a time, and the results are compared to the user's input. True is returned if it is correct. It returns false otherwise.

The method validateAndAddToDB() gets and validates the person's name and phone number. It performs an INSERT query in the table after validation and adds the record to the database table.

listAllRecords () runs a SELECT query and returns a list of all the records in the database table.

The method deleteRecord () gets the person's name and deletes the corresponding record in the database table using a DELETE query in the table. It also gets the person's phone number and deletes the corresponding record in the database table using a DELETE question in the table.

The method getUID() gets the UID of the program.

Installation, setup

This program uses sqlite database with prepared statement for queries.

1) The submitted folder should be unzipped.
2) Using cmd, open the command line and navigate to the directory containing the project.
3) To see the code, open the project in Eclipse IDE.
4) Download sqlite-jdbc-3.20.1.jar or any other sqlite jdbc driver jar

Compilation/build instructions and execution instructions

Make sure the java source file and the sqlite jar are in the same directory/ folder

Steps for compilation

1) javac NamePhoneValidator.java

Execution step

1) java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidator (This displays Usage)
2) java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidator ADD “Aparna Bhat” “(805)908-8355”
3) java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidator DEL “Aparna Bhat”
4) java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidator DEL “(805)908-8355”
5) java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidator LIST

Assumptions you have made

1) I'm assuming you have a device with the most recent version of Java installed.
2) I'm also assuming you have Eclipse installed on your computer.
3) I'm assuming a consumer types in a phone number with a country code.
4) I anticipate the user using cmd from the command line to execute the commands. The command line is used to include the arguments.

Pros/Cons of your approach

Pros

1) Prepared statements have been used as an API to support parameterized queries in SQL queries. This prevents any attacker from launching a SQL injection attack.
2) I used SQLite as a database engine to save the phonebook to disk.
3) To prevent invalid user inputs, a regex expression is used.

Cons

1) Regex expressions may not be able to handle all types of invalid data. Names like O'Malley and John F., for example, are not recognized as legitimate inputs. Names like Brad Everett Samuel Smith, on the other hand, are recognized as true inputs. Numbers such as 011 1 703 111 1234 are not recognized as legitimate inputs. Phone numbers like +01 (703) 123-1234, on the other hand, are recognized as legitimate inputs.

Screenshots

```
[05/07/21]seed@VM:~$ javac NamePhoneValidator.java
[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or
Opened database successfully
Usage:
ADD "<Name>" "<Phone number>"
DEL "<Name>"
DEL  "<Phone number>"
LIST
[05/07/21]seed@VM:~$
```

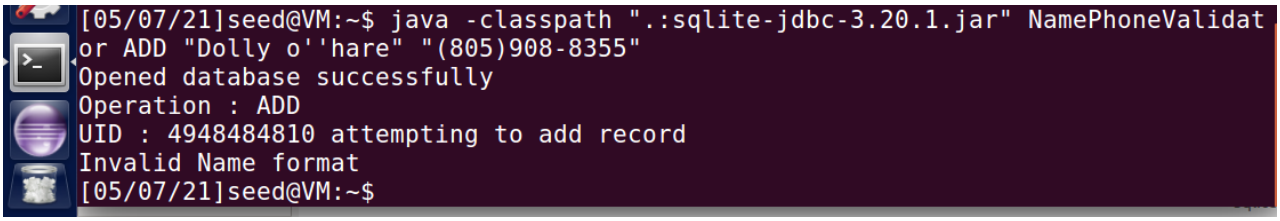
```
[05/07/21]seed@VM:~$ javac NamePhoneValidator.java
System Settings
eed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or ADD "Aparna Bhat" "(805)908-8355"
Opened database successfully
Operation : ADD
ph2 match
Opened database successfully
added  Aparna Bhat into DB at Fri May 07 20:12:23 EDT 2021
[05/07/21]seed@VM:~$
```

```
[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or LIST
Opened database successfully
Operation : LIST
Opened database successfully
Aparna Bhat      (805)908-8355
[05/07/21]seed@VM:~$
```

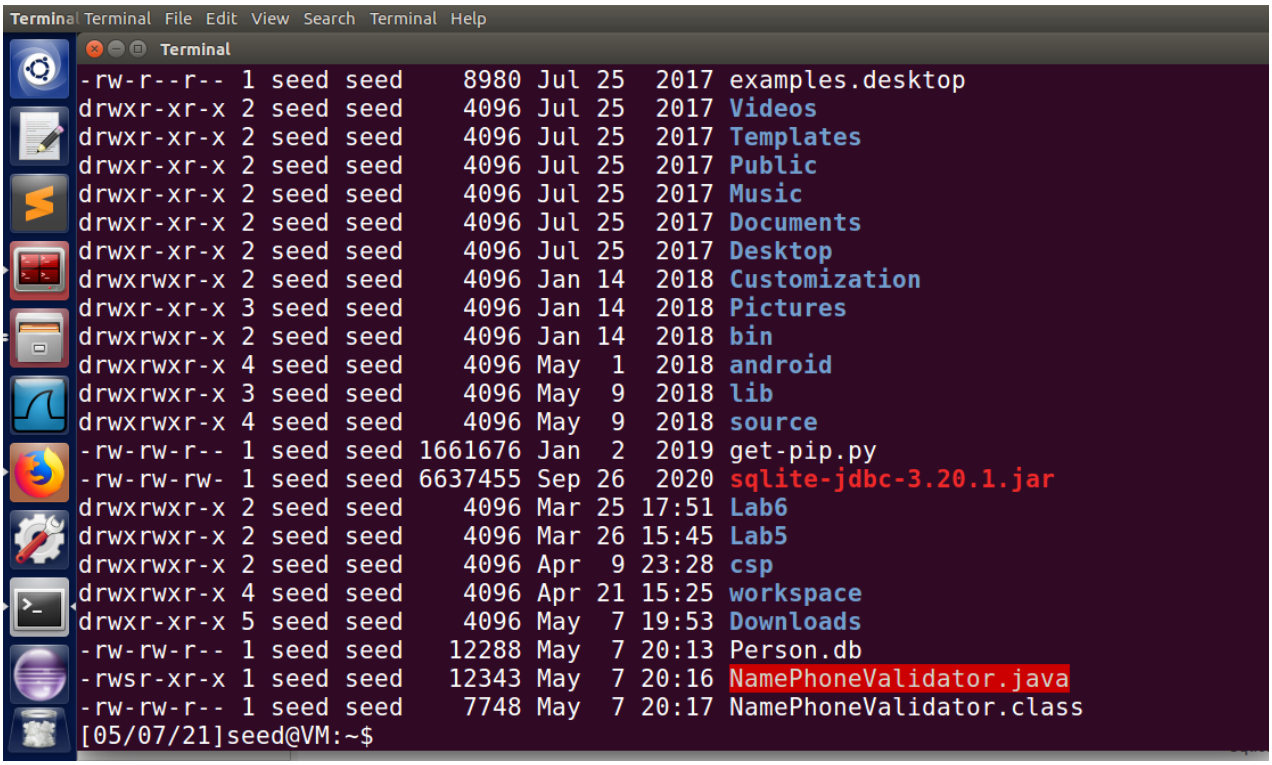
```
[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or DEL "Aparna Bhat"
Opened database successfully
Operation : DEL
Opened database successfully
Deleted Aparna Bhat record from DB at Fri May 07 20:13:25 EDT 2021
[05/07/21]seed@VM:~$
```

```
Opened database successfully
Operation : DEL
Opened database successfully
Deleted Aparna Bhat record from DB at Fri May 07 20:13:25 EDT 2021
[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or LIST
Opened database successfully
Operation : LIST
Opened database successfully
[05/07/21]seed@VM:~$
```

```
[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or LIST
Opened database successfully
Operation : LIST
UID : 4948484810 attempting to list records
Opened database successfully
[05/07/21]seed@VM:~$
```


[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or ADD "Dolly o''hare" "(805)908-8355"
Opened database successfully
Operation : ADD
UID : 4948484810 attempting to add record
Invalid Name format
[05/07/21]seed@VM:~$


[05/07/21]seed@VM:~$ java -classpath ".:sqlite-jdbc-3.20.1.jar" NamePhoneValidat
or ADD "Dolly" "+1234 (805)908-8355"
Opened database successfully
Operation : ADD
UID : 4948484810 attempting to add record
Invalid Phone number format
[05/07/21]seed@VM:~$


Terminal Terminal File Edit View Search Terminal Help
Terminal
-rw-r--r-- 1 seed seed 8980 Jul 25 2017 examples.desktop
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Videos
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Templates
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Public
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Music
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Documents
drwxr-xr-x 2 seed seed 4096 Jul 25 2017 Desktop
drwxrwxr-x 2 seed seed 4096 Jan 14 2018 Customization
drwxr-xr-x 3 seed seed 4096 Jan 14 2018 Pictures
drwxrwxr-x 2 seed seed 4096 Jan 14 2018 bin
drwxrwxr-x 4 seed seed 4096 May 1 2018 android
drwxrwxr-x 3 seed seed 4096 May 9 2018 lib
drwxrwxr-x 4 seed seed 4096 May 9 2018 source
-rw-rw-r-- 1 seed seed 1661676 Jan 2 2019 get-pip.py
-rw-rw-rw- 1 seed seed 6637455 Sep 26 2020 sqlite-jdbc-3.20.1.jar
drwxrwxr-x 2 seed seed 4096 Mar 25 17:51 Lab6
drwxrwxr-x 2 seed seed 4096 Mar 26 15:45 Lab5
drwxrwxr-x 2 seed seed 4096 Apr 9 23:28 csp
drwxrwxr-x 4 seed seed 4096 Apr 21 15:25 workspace
drwxr-xr-x 5 seed seed 4096 May 7 19:53 Downloads
-rw-rw-r-- 1 seed seed 12288 May 7 20:13 Person.db
-rwsr-xr-x 1 seed seed 12343 May 7 20:16 NamePhoneValidator.java
-rw-rw-r-- 1 seed seed 7748 May 7 20:17 NamePhoneValidator.class
[05/07/21]seed@VM:~$