Laura Florez

Software Development 1

Pablo Rivas

12 December 2016

Random Password Generator

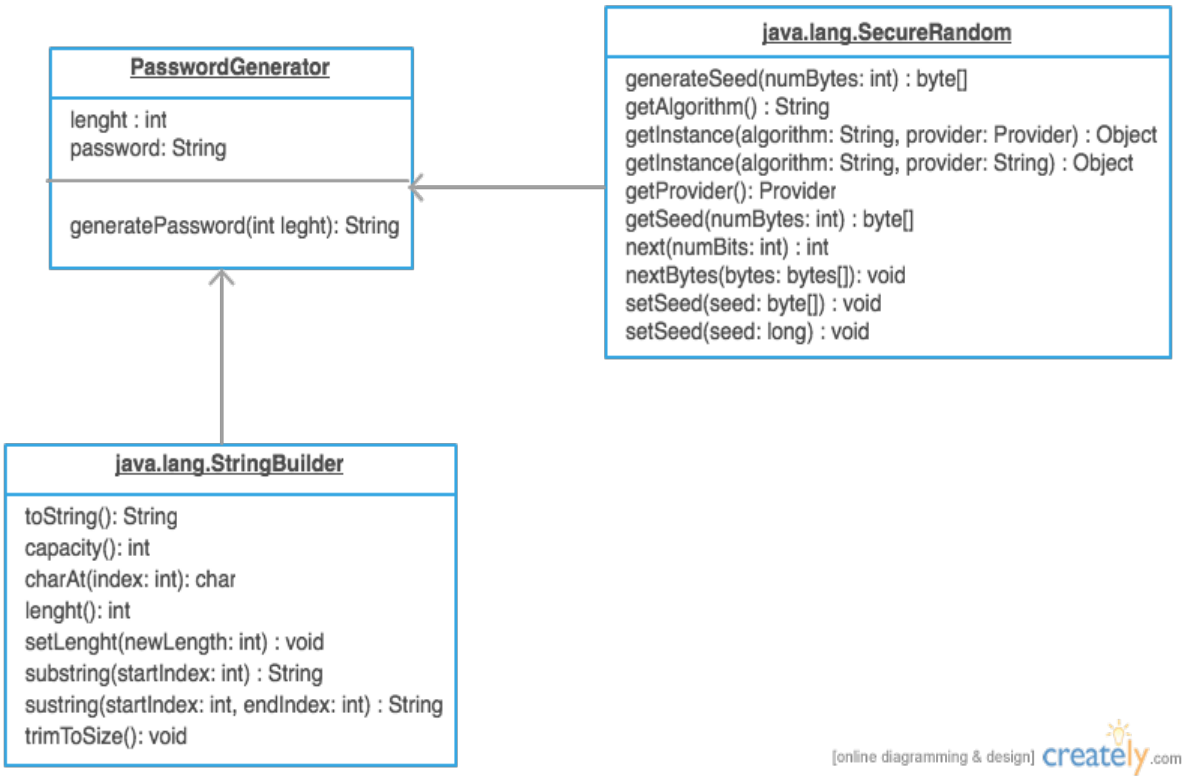
Abstract:

This project will be focus on creating strong randomly generated passwords. The length of the password will be based on the input of the user. The strength of the output will vary depending of the length. The password will be stronger as it becomes longer.

Introduction:

In technology, one of the topics most talked about is security. There is the saying that there are only two types of companies, those who know they have been hacked and those who do not know they have been hacked. There is nothing that cannot be hacked, and one of the most common security methods that are implemented are passwords. Passwords are inexpensive and easy to use. However; users often create passwords that are easily hack through dictionary attacks or other type of attacks. This program will generate passwords that are secure enough through a random selection of number, letters, and special characters. The rest of the paper will go in more details as to how the program works and interacts with the users.

Detailed System Description:



The program is very simple. The user is asked how long he/she would like the password to be. Depending on what number the user inputted, the program will select a combination of letters, special characters, and numbers and will print it. The program uses inbuilt java classes: StringBuilder and SecureRandom. StringBuilder: “Internally, these objects are treated like variable-length arrays that contain a sequence of characters. At any point, the length and content of the sequence can be changed through method invocations” (String). SecureRandom: “This class provides a cryptographically strong random number generator (RNG). Additionally, SecureRandom must produce non-

deterministic output. Therefore any seed material passed to a SecureRandom object must be unpredictable, and all SecureRandom output sequences must be cryptographically strong" (Secure). An empty string builder is constructed with the class StringBuilder. As long as the variable X is less than the length of the password, the method will pull random letters, characters, and numbers according to their corresponding ASCI table values. The combination of characters will then be append/added to the empty builder. Then the method will return the string with the combination.

Requirements:

A computer with a platform/program that is able to compile and run java programs.

Literature survey:

There are several online programs that can be used to generate a password. But since it is online, it is not necessarily safe. The password can be saved by a hacker and used to break into the user's accounts.

User manual:

The program runs based on the input that the user writes. To prevent errors the user must have the knowledge that only numbers can be input. If they input letters or an unreasonable big number, the program will not run properly. Entering a negative number will also break the program.

Conclusion:

This program is a good start to understanding the importance of security in the technological era. It is important to be able to recognize what good security looks like and be aware of the weak points of it.

Bibliography

SecureRandom (Java Platform SE 7). Oracle. N.d,
http://docs.oracle.com/javase/7/docs/api/java/security/SecureRandom.html Accessed 04 Dec. 2016.

StringBuilder (Java Platform SE 7). Oracle. N.d
https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html. Accessed 04 Dec. 2016.