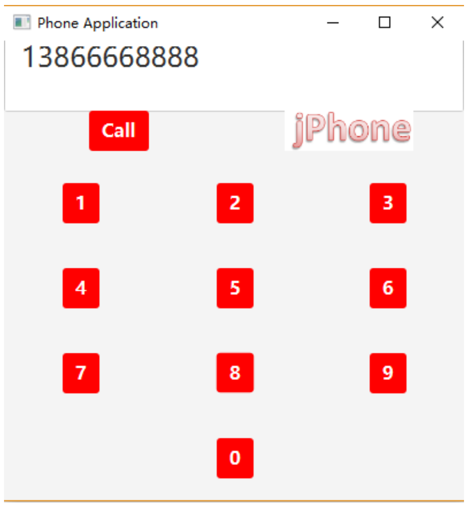
Lab 09: Maintaining A Telephone App

Aims:
1. Maintain an old-versioned project to make it work in a new environment.
2. Hopefully it would help you to better implement your coursework.

This lab is designed to provide some hands-on experience to help you start your coursework, if you have no idea where to begin. To be more specific, you will explore how to maintain a PhoneApp that was originally built using Java 8 in IntelliJ version 2020.2.1. Notice that in this semester, our IntelliJ version is 2024.1.x. Your task is to shift the PhoneApp project built in the old versions to our currently working environment. In technology term, this behavior is called "porting". This is where the term "portable software" comes from.

1. Description of the PhoneApp

The PhoneApp is a JavaFX-based application with the following features:
a. It has buttons (numbers of 0 to 9) to enter the phone number, similar to the telephone keypad of the mobile device. You can of course add other buttons, such as asterisk (*) or hash (#).
b. The buttons are decorated using a .css file.
c. The phone numbers are a purely digits, with length being at least 8.
d. It has a display field to show the phone number as entered.
e. It has a button to make a call. This button is not to make an actual call but should pass the number to a `makeCall` method of a class which implements an `ITelephone` interface. This class should represent a dummy telephone device, and the method should wait for a random duration in seconds and then return the duration of the "call".
f. It has a logo on the phone GUI. You may look into the code on how to display an image in the JavaFX interface.
g. In the current version, it is already implemented in MVC pattern.
h. If can run, the PhoneApp UI should look as below.



2. Hints when you port the code

a. The current version can be downloaded from the Moodle, Week 11, named "PhoneAppOld.zip".
b. Open it with your IntelliJ, and check the structure of the code, you may find various JavaFX project files (src, fxml, css, png, etc.) are clustered in the same package. Now go back and open your previous lab, Lab 07: MVC pattern, observe how those JavaFX project files are structured there.
c. Now come return the current version. To follow a better JavaFX project file, you are suggested to open a new empty JavaFX project in your IntelliJ, and drag the files into it.
d. A working version named "PhoneAppNew.zip"is provided in Moodle for your reference, if you are not confident about how to complete step 2.d.
e. After you have made properly structured files, you may start to debug now. Since the old version was originally runnable, the bugs probably come from mismatched configuration, deprecated libraries used (such as junit, see Lab 08 to find our working Junit library), etc.
f. The image can be lost when you re-organize the structure. Open the "dialpad.fxml" using SceneBuilder. Do you see the icon "jPhone"? It cannot, how to bring it back? Please explore it by yourself.
g. If you are not confident on steps 2.f and 2.e, check "PhoneAppNew.zip". Be reminded that:
    a. When porting a code, you have to thoroughly check if there are any code that should change to adapt to the new environment, not only in source code, but also in resource code, such as fxml, css, and pom (if Maven is used). In this lab, you can do the checking by comparing the code between "PhoneAppOld" and "PhoneAppNew".
    b. Be **very patient** when you do the checking.

Once PhoneApp works on your own environment, congratulations. Then try to adopt your experience into your coursework, if any. Enjoy!