Preparing for the assignment

Please find the attached started code for this assignment zipped as **assignment3.zip**

Import this folder as a gradle project as shown in the class / as mentioned in assignment # 2 (Question # 2)

Run the following gradle tasks (Ctrl + Alt + Shift + R)

For Mortgage Calculator:
```
clean mortgage
```

For Flickr Downloader:
```
clean flickr
```

To make sure that everything is setup, run the demo code by running the following gradle tasks:
```
clean demo
```

Note: Using chrome instead of Firefox
Demo.java (src/test/java/edu/ucsc/extension/Demo.java) uses Chrome instead of Firefox

If you want to use Chrome, you will need to download Chrome Driver from the following website:
http://chromedriver.storage.googleapis.com/index.html?path=2.16/

Download the appropriate zip file and unzip it somewhere on your system. Before instatiating ChromDriver, you will need to set the system property to tell where to find the chrome driver.

Windows: If you have downloaded this to C:\toools you want to set the property as:
```
System.setProperty("webdriver.chrome.driver",
"C:\\tools\\chromedriver");
```

Mac: If you have downloaded this into Downloads folder, you want to set the property as:
```
System.setProperty("webdriver.chrome.driver",
"/Users/jatins/Downloads/chromedriver");
```

Please use the course forums (instead of sending me a message) if you face any difficulties during the assignment

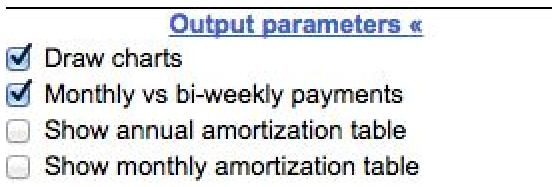
Question # 1: Mortgage Calculator

Your task is to test the following website:
http://www.mortgagecalculator.org/

After the page is loaded, please enter the following information:
Home Value: 600000
Loan Amount: 500000
Interest Rate: 5%
Start Date: Jan 2016

(Rest of the details remains the same)

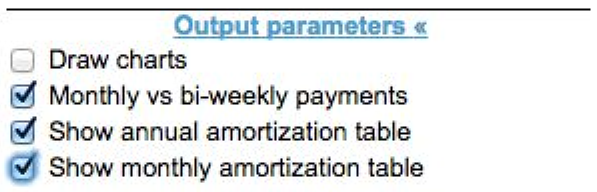
By default the following parameters are set:



Output parameters «

- ✓ Draw charts
- ✓ Monthly vs bi-weekly payments
- ☐ Show annual amortization table
- ☐ Show monthly amortization table

For this assignment you will change it such that following is selected instead:



Output parameters «

- ☐ Draw charts
- ✓ Monthly vs bi-weekly payments
- ✓ Show annual amortization table
- ✓ Show monthly amortization table

Press “Calculate” to generate the report

After the report is generated, you will need extract information and print it on System.out.

The output of your test should look like following:

```
Monthly Payment : $3,309.11
Monthly Payment : $3,309.11
Total of 360 Payments : $1,191,278.92
Total Interest Paid : $459,820.59
Total Interest Paid : $459,820.59
Pay-off Date : Dec, 2045
Total Tax Paid : $225,000.00
Total Tax Paid : $225,000.00
Total PMI Paid : $6,458.33
31 Monthly PMI Payments of $208.33 Each : Monthly PMI
31 Monthly PMI Payments of $208.33 Each : Monthly PMI
PMI Pay-off Date : Aug, 2018
Monthly Payment : $3,309.11
Monthly Payment : $3,309.11
Bi-weekly Payment : $1,654.55
Monthly Pay-off Date : Dec, 2045
Monthly Pay-off Date : Dec, 2045
Bi-weekly Pay-off Date : Mar, 2041
Total Interest Paid : $459,820.59
Total Interest Paid : $459,820.59
Total Interest Paid : $373,929.13
```

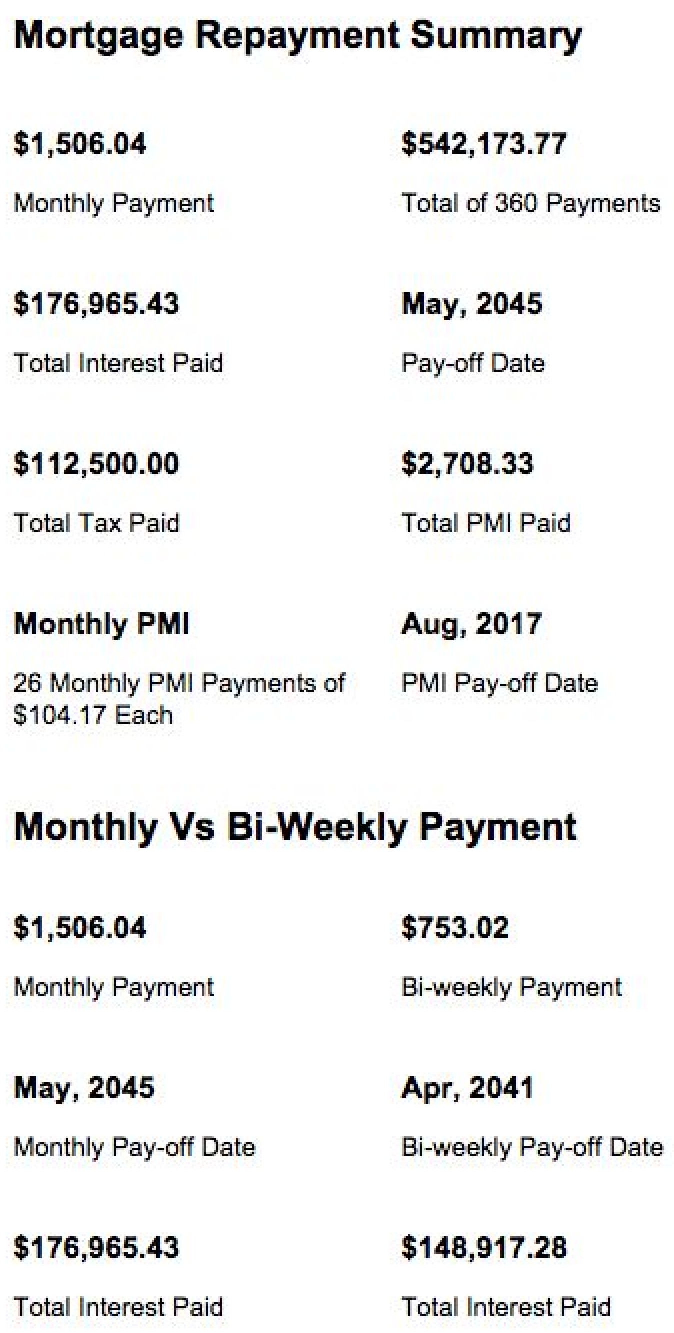
Writing code

Put all your code in MortgageCalculator.java
(src/test/java/edu/ucsc/extension/MortgageCalculator.java)

Run your test in Eclipse by:
right click project >> Gradle >> Task Quick Launcher (or use the keyboard shortcut)
and type: `clean mortgage`

To run the test from command line:

```
cd <project-root>
gradle clean mortgage
```



Mortgage Repayment Summary

$1,506.04
Monthly Payment

$542,173.77
Total of 360 Payments

$176,965.43
Total Interest Paid

May, 2045
Pay-off Date

$112,500.00
Total Tax Paid

$2,708.33
Total PMI Paid

Monthly PMI
26 Monthly PMI Payments of $104.17 Each

Aug, 2017
PMI Pay-off Date

Monthly Vs Bi-Weekly Payment

$1,506.04
Monthly Payment

$753.02
Bi-weekly Payment

May, 2045
Monthly Pay-off Date

Apr, 2041
Bi-weekly Pay-off Date

$176,965.43
Total Interest Paid

$148,917.28
Total Interest Paid

Screenshots:

Please take a screenshot before and after the test (Use FireFox since it takes screenshot of the entire page).

Helper Function: There is a helper function which is provided in the Util class with the following signature:

```
public static void takeScreenshot(WebDriver driver, String
destFileName)
```

The function requires an instance of the web driver as well as the name of the output file name.

Output folder (Where to put the screenshot)
The test class has a static variable called output folder which gets populated with the gradle temp location when the test is started. This directory is:

<project-root>/build/tmp

You want to put your screenshot (before.png and after.png) in this directory.

Calling the function as follows should do the trick:
```
takeScreenshot(driver, outputFolder + File.separator + “after.png”);
```

Question # 1: Flickr Downloader

Purpose of this test is to go to the following website:
https://www.flickr.com/explore

And download all the hi-res images into the output folder.

Note: Please do not download the photos that are shown on the explore page (they are thumbnails). We are looking at downloading the hi resolution images which are displayed when the image thumbnail on the explore page is clicked.

e.g. If I click the first image on the https://www.flickr.com/explore … I go to a page like:
https://www.flickr.com/photos/agrusoft/19227374292/in/explore-2015-06-28/

We are interested in downloading the hi resolution photo from the latter page. You may need to navigate to each of the photo pages to reach a page where you can download the hi resolution photos.

Writing code

Put all your code in FlickrDownloader.java
(src/test/java/edu/ucsc/extension/FlickrDownloader.java)

Run your test in Eclipse by:
right click project >> Gradle >> Task Quick Launcher (or use the keyboard shortcut)
and type: `clean flickr`

To run the test from command line:
```
cd <project-root>
gradle clean flickr
```

Where / How to download the photos:

All the photos should be downloaded into the output folder which is defined by the static variable “outputFolder” in your test. This folder is actually the build folder than gradle creates when building running your tests. It can be found at:

<project-root>/build/tmp

To download the photos you can use FileUtil class (provided via commons-io dependency in your gradle project). FileUtil class has variaty of functions which allow you to create a file, copy a file and download a file. Figuring out which function to use (or you can also write your own code) is left as an exercise.

Turning in the assignment

Once you have finished writing your code and made sure that it works, zip up the entire folder (delete the build directory under the project root) and submit it as a single zip file.