

## CSCI 2073 – Fall 2017 – Programming Assignment 3

Internet browsers provide users with a lot of flexibility in navigating sites and documents on the Internet. As we all know, users can load individual web pages, return to previously viewed pages, move forward to more recently viewed pages, and display a history of visited pages. Your task is to write a Java class to simulate the navigational tools of an Internet browser, using Stack object(s) to keep track of the sites visited.

Your assignment is to implement the *Browser* class, with the methods described below. A *Browser* object contains a web document represented by its *Uniform Resource Locator* or *URL*:

- A no-argument constructor that simply initializes a *Browser* object with url “http://www.ulm.edu”
- A constructor that takes an argument a String object representing a page to be loaded initially. If successful (see *load* method below), the argument string should become the URL for the current page. Otherwise, the *Browser* object should be initialized with the page “ERROR: CANNOT FIND <url>”
- `boolean load(String url)`: attempts to load the web page with the URL passed as argument. To determine whether the URL is valid, use the java URL class from the `java.net` package (see example below). If successful, the URL argument should become the current page and the method should return true. Otherwise, the method should return false and the current page should be “ERROR: CANNOT FIND <url>” (Note that loading a page erases any “forward” pages)
- `String currentPage()`: returns a String object containing the URL for the current page.
- `boolean canGoBack()`: returns true if there is a previous page in the browsing history, false otherwise.
- `boolean canGoForward()`: returns true if there is a next page in the browsing history, false otherwise.
- `String goBack()`: loads and returns the previous page in the browsing history. If there is no “previous” page to load, the method should have no effect and should simply return the current page.
- `String goForward()`: which should “load” and return the next page in the browsing history. If there is no “next” page to load, the method should have no effect and should simply return the current page.
- `String history()`: returns a list of the pages loaded in the current browsing session, with most recent page listed first. No “ERROR: CANNOT FIND url” pages should be included in the history.

### REQUIREMENTS and RESOURCES:

---

- Download and use the `StackInt` interface and the `LinkedList` class studied in class. Do not modify either one. Be sure to declare any `Stack` variables using the `StackInt` type.
- The Java URL class can be used to establish a connection to a resource on the internet. In the simplest case, the constructor accepts a String containing a URL. The resulting object can be used to open a stream, read its contents, etc. The `getContent()` method can be used to test whether the URL corresponds to an actual website. The constructor throws an exception if the URL is not properly formatted. Sample code segment:

```
try {
    URL webpage = new URL(url);
    if (webpage.getContent() != null)
        ...                // url properly formatted and found
}
catch (Exception e) {      // url was not properly formatted
    ...
}
```

- The `BrowserTest` class is provided for basic testing. Once you are satisfied your class works as expected, submit `Browser.java` to Mimir for testing.