

Assignment 3
Due October 2, 11:59 pm

- 1a. (5 points) Add a method to the Array class to return the average of all the odd numbers in the array. Assume all elements in the array are numbers. So #(1 5.3 4 5 3) would result in the average 3.
- 1b. (2 points) Write Sunit test(s) for your method in part a.
- 2a. (5 points) Add a method valuesBetween: a and: b to the Array class. The method returns an Array that contains all the elements of the receiver that are between the values "a" and "b". Use select: in the method.
- 2b. (2 points) Write Sunit test(s) for your method in part a.
- 3b. (5 points) Add a method squares to the Collection class. This method returns a collection that contains the squares of the values in the receiver collection. Use collect: to compute the square of each number of a collection .
- 3b. (2 points) Write Sunit test(s) for your method in part a.
- 4a. (20 points) Implement a Stack class that implements the methods listed below. The stack should be able to contain any type of object. The stack should use a linked list to store the elements. Do not use an array. While Smalltalk does have Link class it will not be useful for this assignment so do not use it.

push: - Adds the argument on top of the stack

pop - Removes and returns the top of the stack

clear - Removes all items from the stack

size - Returns the number of elements in the stack

do: aBlock - Evaluates "aBlock" for each element in the stack

withAll: aCollection - a class method that creates and returns a stack. The stack contains the elements in the argument (aCollection). The first element of aCollection ends up in the bottom of the stack, the last element of aCollection ends up on the top of the stack.

printOn: aStream - prints a presentation of the stack. For example:

```
| sample |  
sample := Stack new.  
sample
```

```
push: 'a';
push: 'b';
push: 'c'.
sample printString "returns (c, b, a)"
```

The elements in the stack are inside parentheses, separated by commas and the elements are in the same order as they are in the stack with the top of the stack being the left most element in the printString result.

4b. (5 points) Write Sunit tests for your Stack class.

5. (16 points + 4 points for SUnit tests) Create an HtmlTable class and SUnit tests for it. An HtmlTable object holds an N*K matrix of strings. The class needs to have the following operations:

```
HtmlTable class>>rows: numberOfRows columns: numberOfColumns
Returns an HtmlTable object that has the given number of rows and columns.
```

```
HtmlTable>>row: rowIndex column: columnIndex
Returns the string in the given location
```

```
HtmlTable>>row: rowIndex column: columnIndex put: aString
Puts aString in the given location.
```

```
HtmlTable>>asHtml
Returns a string containing the html representation of the table.
```

For example:

```
I table I
table := HtmlTable rows: 2 columns: 2.
table
  row: 1 column:1 put: 'hi';
  row: 1 column: 2 put: 'mom';
  row: 2 column: 1 put: 'how';
  row: 2 column: 2 put: 'are you'.
table asHtml
```

the last statement will return the string

```
<table>
  <tr>
    <td>hi</td> <td>mom</td>
  </tr>
  <tr>
    <td>how</td> <td>are you</td>
  </tr>
</table>
```

You might find the method CharacterArray>>match: useful in your unit test(s).

How to turn in the assignment

In your image create Package called Assignment2. Make sure that all the code for this assignment is in your Assignment2 package. You will be given a store account (source code repository) for the course. You will upload your Assignment2 package to your store account. We will discuss this operation in class.