CS4141 Introduction to Programming Assignment

Part 4 Specification

Whilst the `ToolBox` class developed in Part 3 is a useful mechanism for organising a collection of methods it has the drawback that even though all of the methods manipulate the same type of 'thing' or object (i.e. a tweet text) we have to keep passing the tweet text to each method.

In the final part of the assignment, we are going to develop a class called `Tweet` that will be used to store a single tweet and will include a set of methods for constructing and manipulating `Tweet` objects. The class will be stored in a file called `Tweet.java`. `Tweets` are immutable objects – once you create on you cannot alter the contents.

The following skeletal class provides an overview of the class.

```
public class Tweet {
   // NOTE: All the instance variables are private
   //       so they can only be manipulated by the
   //       methods in the class. So we control
   //       how a Tweet "behaves."
   private String postedBy;
   private String tweetText;
   private int hashtagCount;
   private int usernameCount;
   private long postedAt;

   private final int DEFAULT_DISPLAY_WIDTH = 30;
   private final int MINIMUM_DISPLAY_WIDTH = 1;
   private final int MAXIMUM_DISPLAY_WIDTH = 80;
   private final int MINIMUM_HEADER_WIDTH = 10;
}
```

Add code to the class to support the following operations

1. `public Tweet(String text)`

   This is the constructor method used to create instances of the Tweet class. The `text` passed as a parameter should be stored in the `tweetText` instance variable. The `postedBy` String should be set to the user name with an '@' symbol preceding it. The `hashtagCount` and the `usernameCount` should be set to the number of '#' and '@' symbols contained in the `text`. The `postedAt` value should be set to the current system time in milliseconds.

2. **Add a `'get'` method for each instance variable (i.e. `getTweetText`, `getHashtagCount`, `getUserCount` and `getPostedAt`). Each method should return the current value of the instance variable. Note: Because a Tweet is an immutable object we are not providing any 'set' operations – so you cannot change the instance variable data after a Tweet has been created.**
3. `private int count(char symbol)`

   The method returns the number of times the `symbol` passed as a parameter appears in the `tweetText`.

4. `public String postedAtTime()`

   The method returns a String that contains the `postedAt` time formatted in the style HH:MM:SS, with leading zeros for any value less than 10. For example, the method might return any of the values "12:34:56" or "07:22:06" or "00:00:09" or "23:59:21". Note: You may find the String format operation helpful when coding this method.

5. `public int timeInSeconds()`

   The method returns the value of the `postedAt` time as a number of seconds.

6. `public int timeInSeconds(int hour, int minute, int second)`

   The method returns the time specified in the parameter values as a number of seconds. The parameter values must satisfy the following constraints

   0 <= hour <=23　　0 <= minute <= 59　　0 <= second <= 59

If any of the parameter values is incorrect the method should return -1.

7. `public char activityIndicator()`

   The method returns one of the following values depending on the `postedAt` time.

Return Value	postedAt
‘N’	22:00:00 <= postedAt < 06:00:00
‘M’	06:00:00 <= postedAt < 12:00:00
‘A’	12:00:00 <= postedAt < 17:00:00
‘E’	17:00:00 <= postedAt < 22:00:00

8. `public boolean containsWord(String aWord)`

   The method returns `true` if the `tweetText` contains the word passed as a parameter. Remember, the word may be in upper, lower or mixed case. The word must appear in the `tweetText` on its own and not as part of a longer word.

   For example, a `tweetText` with the words – *hard*y, orc*hard*, or c*hard*onnay) – does NOT contain the word hard.

   Similarly, a `tweetText` with the words – *work*ing, un*work*able, wood*work* – does NOT contain the word work.

   If the `tweetText` does not contain the word the method returns `false`.

   NOTE: To code this method you will need to reason carefully about the possibilities. For example, the word might be the first word with no text before it; or the last word with no text succeeding it. Take some time to work through the possibilities BEFORE you start coding. And remember, `indexOf` is a useful way of finding text in strings.

9. `public void display(int width, boolean includeHeader)`

The method displays the tweet as a series of lines using the specified `width`. If the `includeHeader` parameter is `true` then the display should include a header. Otherwise, no header should be displayed.

If the `width` is less than 1 then the `tweetText` lines should be displayed using the MINIMUM_DISPLAY_WIDTH.

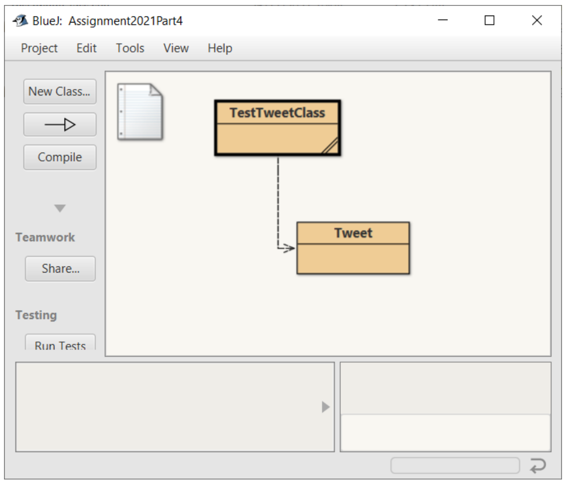
If the `width` is greater than the MAXIMUM_DISPLAY_WIDTH then the `tweetText` lines should be displayed using the MAXIMUM_DISPLAY_WIDTH.

If a header is included then it should be at least the `MINIMUM_HEADER_WIDTH`. If the display `width` for the tweet text is greater than the `MINIMUM_HEADER_WIDTH` the header should be displayed using the same width as the tweet text display. The header and tweet display should have the same layout as those included in the Part 3 specification.

Testing Your Tweet Class

To test the `Tweet` class you should use a "driver" program that creates a number of instances (i.e. more than one) of the `Tweet` class and uses the various methods provided by the class. A sample driver program, that may assist you in developing your own driver program, has been included below. Please note this is just a simple example.

You should create your "driver" program class and your `Tweet` class in the same project. The driver program class should be called TestTweetClass.java and the code for the `Tweet` class should be stored in a file called Tweet.java. Here is how BlueJ shows the relationship between the code in the `Tweet` class and the `TestTweetClass` driver program.



Submission Requirements

Your solution to the assignment should be submitted on Sulis on or before 16h00 on Friday 10th December 2021. We will accept submissions up to 23h55 Friday 10th December 2021 but please note that Sulis identifies submissions that are after the deadline. For this part of the assignment there is no penalty for submitting on or before 23h55.

You should submit TWO files (1) Tweet.java and (2) TestTweetClass.java. In each file you should include prominent comments that contain your ID number and your Name. Sulis will accept any filenames you use but it helps us if you adhere to these conventions.

This part of the assignment is worth 30% of the total of 100%.

Marking Scheme for CS4141/CS6371 Programming Assignment Part 4

Requirement	Marks
`Constructor` method	3
All of the `get` methods	2
Integrate methods from Part 3	4
`timeInSeconds` method	2
`activityIndicator method`	4
`containsWord` method	10
Driver Program	5
Total	30

Sample Driver Program

```
public class TestTweetClass {

        public static void main(String[] args) {
```

```
String aMessage = “Have a nice Christmas everyone. #relax #santa @ULStudents”;
String anotherMessage = “This semester I hardly went out. #staySafe”;
String yetAnotherMessage = “Glad to have a break from UL work. @CS4141 #CSIS”;
kerstynsTweet = new Tweet(aMessage);
kerstynsTweet.display(12,true);
if(kerstynsTweet.contains(“christmas”)) {
System.out.println(“Season Greetings”);
}
String theText = kerstynsTweet.getTweetText();
int numberOfHashtags = kerstynsTweet.getHashtagCount();
char twitterTime = kerstynsTweet.activityIndicator();
if(twitterTime == ‘N’) {
System.out.println(“Don’t stay up late on twitter”);
} else if(twitterTime == ‘M’ || twitterTime == ‘A’) {
System.out.println(“Angela Merkel probably thinks you’re working now!”);
} else {
System.out.println(“Is twitting better than watching the telly?”);
}


System.out.println(“Posted at ” + kerstynsTweet.timeOfDay());
Tweet covidAwareStudent = new Tweet(anotherMessage);
covidAwareStudent.display(4,false);
int n = covidAwareStudent.timeInSeconds(11,45,00);
System.out.print(“Interesting fact... The number of seconds from midnight”);
System.out.println(“ to 11:45:00 is %,11d\n”, n);
Tweet itsChristmas = new Tweet(yetAnotherMessage).
itsChristmas.display(15,true);
if(itsChristmas.contains(“break”)) {
System.out.println(“yea”);
}
```

```
    }
}
```