COMP 1030

Lab #10

String Class Methods

Introduction

During this lab you will build three java classes. The first class will contain the required state and behaviour for the object, **but NO main method**. The second class will contain simply the main method to give the JRE an entry point into the program, a line to instantiate a new object based upon the first class and a few lines to exercise the functionality of the first class.

When writing your code, keep these guidelines in mind:

- Start each class with the proper javadoc comment header. The first line of that comment should be the purpose of the class.
- Provide a comment for **each** section of code as well as a javadoc header.
- Follow the layout for your class as illustrated below:

Javadoc comment header
Import statements (if required)
Class declaration
    State (instance variables/data)
    Constructor(s) (if required)
    Behaviour(s) (method(s))
Close class declaration

- Use whitespace and indentations to make your code more readable and easy to debug.
- Be sure to clearly understand your work – do not simply copy code from someone else.

Instructions: Use any IDE to complete this assignment other than a simple notepad or BlueJ.

- You will need to create a third class to solve method two. You need to determine its required state and behaviour.
- Create a class which represents a SuperString object.
    - The class has no instance variables.
    - The class has no setters. The class has no getters.
    - The class has no constructor.
    - The class has three static methods
        - A static method which will take one String argument made up of 5 possible words (chosen from a list of 10 that you make up) that range from 2-4 characters in length. These words are concatenated with no spaces and from the argument for the method. The purpose of the method is to look through the single string argument and identify the five words based upon the list of 10 possible words and print out each word found on a separate line. For example, assume the 10 words you have chosen to be on your list were:
            - Hat
            - Cat
            - Bat
            - Fine
            - Pine

- At
- Pi
- Bee
- Art
- Dog

And the string passed into the method was: HATCATBATFINEAT – the job of the method will be to separate out the five words used from the list of ten and print each word on a separate line.

- Static method number two will take three arguments: String, int, char. The purpose of the method is to break the string up into substrings of a length equal to the int argument (which must not be greater then the length of the String argument) and return the result to the caller in the form of a new object that you create for the purpose of holding the information to be returned. Each substring will begin at the location of the occurrence of the char that is passed in as the third argument. For example:
  - if the string passed in was: ABCDEFCHJYCDERLCHHHCYDC
  - If the int passed in was 2
  - If the char passed in was 'c'

  The return value would be an object which contained the following data:

  - CD
  - CH
  - CD
  - CY

- Write a test harness class (this is the third class) to test these two methods and print out the appropriate strings for each method.

Things to consider for success:

- Follow the layout for your class as illustrated above.
- Write POC code before tackling the full problem.
- Write a section at a time and compile after each section so you do not have a volume of compiler errors to deal with.
- Comment as you go.
- Use indents and whitespace appropriately to make your code readable.
- Be sure that you actually understand the code you have written, simply copying someone else's code will not aid in your understanding of the java language.
- Stay focused and work diligently, collaborate with others if you are stuck.