Assignment 1

Instructor: Musard Balliu

September 3, 2017

Introduction

String manipulation is a very important operation with applications in different areas of Computer Science, for instance web application security, parsing and many others. In the third lab session, we have used Java String operations such as `length, equals, charAt, substring` to solve different programming tasks. In this assignment, we will implement some of these operations ourselves by only using a subset of existing Java String operations. Concretely, we will implement *concatenation, equality, substring, trim, lastIndexOf* and *contains*. Given two Strings s1, s2 and a character c, the **only** String operations we are allowed to use to solve the tasks are:

- *length*: for example, you may use s1.length() to calculate the length (number of characters) of String s1.
- *charAt*: for example, you may use s.chatAt(0) to calculate the first character of String s1.
- *restricted concatenation* $+$: for example, you may write s1 + c or c + s1 to concatenate String s1 to character c (and viceversa). However, you are not allowed to use unrestricted String concatenation such as s1 + s2. In fact, unrestricted String concatenations is one of the operations that we will implement ourselves by using restricted concatenation.
- Otherwise, you are allowed to use any other Java statements and operations that we have seen so far, including conditionals and loops.

The following examples provide an overview of the String operations that we will use and implement in this assignment.

String operations you are allowed to use:

```
String s = "Hello"; // declare a String variable with initial value "Hello"
char c = 'l';       // declare a char variable with initial value 'l'
s.length();         // returns the number of characters in s, namely 5
s.charAt(1);         // returns the char at position 1 in s, namely 'e'
s+c;                 // returns the concatenation of s and c, namely "Hellol"
```

String operations you should implement (see the user interface below for the correct format):

```
1. concatenation s, s1: computes the result of appending the String s1 to s
2. equality s, s1:      checks if s and s1 are the same string
3. substring s, i, j:   computes (if it exists) the substring between position
                        i and j in s (excluding char at position j)
4. trim s:              removes all the empty spaces from the String s
5. lastIndexOf s c      computes the index within String s of last occurrence
                        of the character c. Otherwise it returns -1.
6. contains s s1        checks if the string s1 is a substring of s.
```

Main Task

You will implement a Java program that prompts the user to choose one of the operations above, and, depending on the chosen operation, it prompts the user to input the data. Subsequently, the program makes use of the input data to compute the desired operation. The program should check whether or not the chosen operation exists, namely it is one of the implemented operations, and you can assume that the input data is always of the correct type. If the chosen operation does not exist, the program should inform user to choose a correct operation, without terminating the program. Finally, if the user inputs the character 'q', the program should terminate.

The program should present the following interface to the user.

```
******************************
**                          **
**  Welcome to the DIT042   **
**    String Library        **
**                          **
******************************
```

```
Please choose a String operation or press ’q’ to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
c
Please enter the first string: He
Please enter the second string: llo!
The result of concatenating He and llo! is: Hello!

Please choose a String operation or press ’q’ to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
e
Please enter the first string: Hey
Please enter the second string: hey
The result of comparing Hey and hey is: false

Please choose a String operation or press ’q’ to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
s
Please enter the string: Programming
Please enter the first index: 0
Please enter the second index: 7
The resulting substring is: Program

Please choose a String operation or press ’q’ to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
t
Please enter a sentence to trim: Tr im
The resulting trimmed sentence is: Trim

Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
l
Please enter a string: Hello
Please enter a character: l
The index of character l in string Hello is: 3

Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
h
Please enter the first string: aaabcd
Please enter the second string: abc
The string aaabcd contains the string abc

Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
r
This is not a valid choice. Please retry!
```

```
Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
l
Please enter a string: Hej
Please enter a character: k
The index of character k in string Hej is: -1

Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
h
Please enter the first string: Hello
Please enter the second string: HelloThere
The string Hello does not contain the string HelloThere

Please choose a String operation or press 'q' to quit:
Press c for concatenation
Press e for equals
Press s for substring
Press t for trim
Press l for lastIndexOf
Press h for contains
q
Thanks! Goodbye.
```

Remarks and Notation

1. You **are only allowed** to use the restricted set of String operations as described above. Alternative solutions will not be accepted.

2. You are **not** required to implement the last operation, namely *contains*, in order to pass the assignment.
3. Recall that String indexing starts from 0. Moreover, the empty space character is denoted as ' ' (single quote, followed by one space, followed by another single quote), and the empty String is denoted as "" (two double quotes with no space in between). If you choose to use the *dit042* library for I/O operations, recall that *readString()* returns the first input String entered by the user (up to the first empty space), while *readLine()* returns the entire input line, including all the empty spaces. Note that may not necessarily need to use this information - just in case you find it useful.

Grading

- To get a G it is necessary that the program works and implements correctly all the String operations (with the exception of *contains* which is not mandatory).
- Easy? You are very much encouraged to implement the *contains* operation. This extra task will not count to the final grade, however, it will be very much appreciated by the instructor.

Note that these requirements are necessary, but not sufficient, to get the respective grades (see **Administrative** matters).

Administrative matters

Strive for readable code with appropriate comments! While the ultimate test of a program is that it does what it is supposed to do, **we should be able to read the program and understand it**.

The assignment is to be completed in groups of two students. (Groups of different size should first be agreed with the course instructor.)

Solutions must be uploaded to the GUL system **by 18:00 on September 10**.

Please note: **The submission must be made via GUL. It's no good sending it to me via email, either before or after the deadline!**

Only Java source files should be uploaded, no class files. Your Java files should be put in a zip-archive with the following name:

```
assign1_author1_author2.zip
```

where author1, author2 are the surnames (family names) of the group members. You must also supply a README-file (README.txt) containing the names of the authors and their social security numbers, together with a brief statement about each author's contribution. For example, "author1 has been responsible for user input and author2 for the program logic". A statement such as "All authors have contributed equally to all aspects of the program" is not acceptable.

Note that **each author must submit individually via the GUL** (even though it is the same program!). Only students who submit via the GUL can be graded. All comments etc. will be posted on the course web page.

Once the deadline has passed, each group **must explain the solutions to the TAs during the first Monday supervision session**. Exceptions are to be agreed with the instructor and the TAs. Group members are expected to **know and explain all parts of the code** despite their statement of contribution.