Assignment 1 - CT2109 Object Oriented Programming: Data Structures and Algorithms

Daniel Hannon (19484286)

Feburary 2021

1 Problem Analysis

Abstract: In this assignment, you will write a program for reading in the letters of the alphabet, one at a time, in order from the console. The user should press enter between each letter and incorrect letters in the sequence should be ignored. Once the user has typed the letters in the correct sequence, the program should tell them how long it took to complete. The user can also decide if they would like to type the letters forwards or backwards.

In order to achieve this I must come to grips with the **Scanner** Class.
I must be able to also be able to use the **Date** Class.
As this reads one letter at a time, I want to make sure that words starting with the expected letter do not qualify as valid input, This can easily be achieved by:

- Using **String.stringify(***startIndex,endIndex***)**
- Using **String.equals(***anotherString***)** in conjunction with the afforementioned method

I felt that there was no need for it to be case sensitive so I utilized **Scanner.nextLine().toLowerCase()** to guarantee it was always lowercase.
I stored the alphabet as a string either forwards or backwards depending on the initial input. I then invoke **String.stringify(***position,(position + 1)***)** to get the current letter if **String.equals(***otherString***)** evaluates to true, then *position* is incremented.
Once *position* is equal to the length of the Alphabet string, it breaks.
Time keeping is done as follows

- Initialize *date* to value **new Date()**
- Create a long called *totalTime*
- Set it's value to the value **date.getTime()**
- Once the input is complete reinitialize date to **new Date()**
- Perform the operation *totalTime* = ***date.getTime()*** - *totalTime*
- Display the result.

By using **String.equals()** I am ultimately eliminating the need for **String.length()** as it will always be compared to a string of length 1

2 Code

```java
import java.util.Scanner;
import java.util.Date;

public class Main {
  public static void main(String[] args) {
    String sequence = "";
    String inpstring = "";
    //Reduce invocations of String.substring() so it runs faster :)
    String currString = "";
    Scanner input = new Scanner(System.in);
    int position = 0;
    System.out.println("Type the alphabet in order (hit enter between letters)");
    System.out.print("Forwards or backwards (f/b)? ");
    while(true) {
      // Take string in and force it to be lowercase
      inpstring = input.nextLine().toLowerCase();
      if(inpstring.equals("f")) {
        sequence = "abcdefghijklmnopqrstuvwxyz";
        break;
      } else if(inpstring.equals("b")) {
        sequence = "zyxwvutsrqponmlkjihgfedcba";
        break;
      }
      System.out.println("Invalid. You must type 'f' or 'b' to start.");
    }
    Date date = new Date();
    long totalTime = date.getTime();
    currString = sequence.substring(0,1);
    while(position != sequence.length()) {
      inpstring = input.nextLine().toLowerCase();
      //Java substrings need (startIndex,endIndex) rather than (startIndex,length) for some reason
      //I use the equals method as it deals with both length and the value at position zero
      if(inpstring.equals(currString)) {
        if(position != sequence.length() - 1) {
          System.out.print("["+currString + ": Correct now type: ");
          currString = sequence.substring(position+1,position+2);
          System.out.print(currString + "]\n");
        } else {
          System.out.println("[Correct! Well Done!]");
        }
        position++;
      }
    }
    date = new Date();
    totalTime =  date.getTime()- totalTime;
    System.out.print("Time taken: " + ((float)totalTime/1000) + " seconds\n");
    //Close Scanner
    input.close();
  }
}
```

3 Testing

3.1 Testing inital input

```
[daniel@Void3 Assignment 1]$ java Main
Type the alphabet in order (hit enter between letters)
Forwards or backwards (f/b)? fog
Invalid. You must type 'f' or 'b' to start.
bog
Invalid. You must type 'f' or 'b' to start.
cat
Invalid. You must type 'f' or 'b' to start.
sheep
Invalid. You must type 'f' or 'b' to start.
f
a
[a: Correct now type: b]
b
[b: Correct now type: c]
x
c
[c: Correct now type: d]
d
[d: Correct now type: e]
e
[e: Correct now type: f]
```

3.2 Testing reverse and Caps

```
[daniel@Void3 Assignment 1]$ java Main
Type the alphabet in order (hit enter between letters)
Forwards or backwards (f/b)? b
z
[z: Correct now type: y]
y
[y: Correct now type: x]
dog
cat
xylophone
x
[x: Correct now type: w]
w
[w: Correct now type: v]
v
[v: Correct now type: u]
u
[u: Correct now type: t]
T
[t: Correct now type: s]
S
[s: Correct now type: r]
ROLLS ROYCE
R
[r: Correct now type: q]
Q
[q: Correct now type: p]
```

3.3 Demonstration of Time

```
[s: Correct now type: t]

t
[t: Correct now type: u]
u
[u: Correct now type: v]
v
[v: Correct now type: w]
w
[w: Correct now type: x]
x
[x: Correct now type: y]
y
[y: Correct now type: z]
z
[Correct! Well Done!]
Time taken: 7.277 seconds
[daniel@Void3 Assignment 1]$
```