Exercise 01: Stack and palindrome

In this exercise, we will be implementing the *Stack* data structure (see `CharStack` interface), and then using it to check strings for palindrome properties.

Preparation

1. Clone the lectures project locally, by using IntelliJ to import the project from Git, or by doing so "manually" in the console:
   `git clone LINK SEE LEARNING CAMPUS`
2. Create a local branch called `solution` (or any other name of your choice)
3. IntelliJ: GIT button at the bottom right -> "`+ New Branch`"
4. In the terminal (button at the bottom left: "Console"/"Konsole"): `git checkout -b solution`

Task 1: Stack

1. Complete the `CharStackImpl` class by implementing the `push`, `pop` and `size` methods
   - `push` places an element on top of the stack, `pop` removes the top element; the stack is thus FIFO -- first in - last out, or LIFO -- last in - first out.
   - Also, remember the helper class that you need to model the elements.
   - Verify that the `CharStackTest` test runs without errors.
2. Add the modified `CharStackImpl.java` file as well as the helper class to your commit, in IntelliJ with right click -> Git -> Add, or in the console with `git add CharStackImpl.java`
3. Commit the changes

   - IntelliJ: VCS -> Commit Changes -> Commit
   - Terminal:
   - `git commit -m "Your commit message"`

Task 2: Palindrome

7. Implement the static method `Palindrome.isPalindrome()` in which you now use the stack to test arbitrary strings for palindrome properties.
   - A string is a palindrome if it has the same sequence of letters when read both forwards and backwards, i.e. the text is "mirrored".
     How can the stack be used to check this?
   - The `String.replaceAll` method can be used to remove all spaces.
   - The `String.toLowerCase` (or `String.toUpperCase`) method converts all characters to lower case or upper case respectively.
   - The `String.toCharArray` method returns the string as an array of `chars`.
   - Verify that the `testPalindrome()` test runs without errors.
8. Commit and push the changes.

Please note: the upcoming tasks will always use Git in the same way. You should get used to this workflow, as we will not reformulate it again in later exercises.