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. Create a new blank project under your user name in Git
2. Clone the exercise project locally, by using IntelliJ to import the project from Git, or by doing so "manually" in the console:
   `git clone https://inf-git.th-rosenheim.de/oop-wif-aai/ss25/exercises/exercise01.git`
3. Navigate to the cloned repository and change the remote repository to your newly created repository:
   `git remote set-url origin <URL to your repository>`
4. Push the changes into your repository
   `git push origin main (or master)`

Task 1: Stack Datastructure (using char-Array)

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 from the stack;
     The stack data structure is thus called LIFO -- *last in - first out*.
   - Also, remember to implement a helper class `CharElement` to model an Element of the stack.
   - Verify that the `CharStackTest` test runs without errors.
2. Add the modified classes to your commit, in IntelliJ with right click -> Git -> Add, or in the console with `git add <file name>`
3. Commit and push the changes

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

Task 2: Palindrome

1. Implement the static method `Palindrome.isPalindrome()` in which you now use your 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". The capitalization of characters should be ignored.
     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.

2. Add, Commit and Push your changes.