Java Lab 5 Fall 2022
Due: Thursday, September 15, 10:10 AM EDT

In this lab, you will practice with loops, arrays, the String class, calling methods, and console input and output.

Problem statement: You will write the code needed to complete Palindromer, which takes phrases entered by the user and checks whether they are palindromes – strings that read the same in both directions. Note that real palindromes are supposed to be words or phrases that actually mean something; for this program, you'll simply check the characters, not the sense (or lack thereof) in English of the phrases.

Create a project named Lab5. Download the files Palindromer.java and testPalindrome.java into the src directory.

The main method is already coded for you – do not change it. You will provide the code for the other methods of class Palindromer. The program first prompts the user to enter an integer no larger than 10. It will then prompt the user for that many phrases. The program then tests if the phrases are palindromes by first removing any non-letter characters, converting the characters to upper case for conformity, then testing the cleaned-up phrases. It will display only the phrases that are palindromes and keep count of how many of the phrases are palindromes, displaying that count at the end. Figure 1 shows a typical run of the program, with the user entering two phrases – the first is a palindrome and the second one is not. Figure 2 shows a run where the user enters three phrases; although the first one reads the same both ways, it uses only non-letters so does not count as a palindrome; the second contains punctuation which does not prevent it from being a palindrome; the third is a nonsense string but still a palindrome. Figure 3 shows a longer palindrome, attributed to Peter Hilton [Wikipedia].

```
Palindromer App

Enter the number of palindromes to store (10 max): 2
Enter a phrase: Eve
Enter a phrase: Dog is a word

"Eve" is a palindrome
"Dog is a word" is not a palindrome

Number of palindromes: 1
```

Figure 1

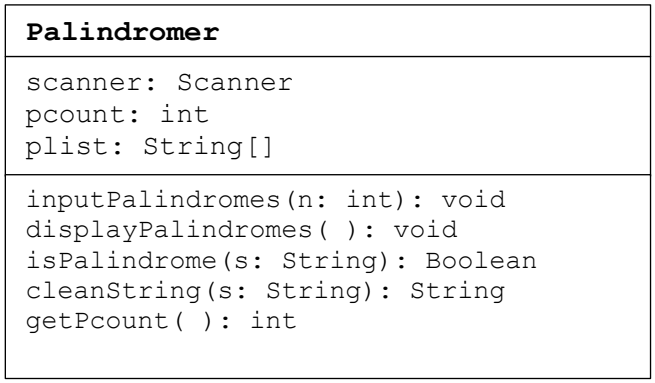
```
Palindromer App

Enter the number of palindromes to store (10 max): 3
Enter a phrase: ??,??
Enter a phrase: Able was I, ere I saw Elba
Enter a phrase: xxyxx

"??,??" is not a palindrome
"Able was I, ere I saw Elba" is a palindrome
"xxyxx" is a palindrome

Number of palindromes: 2
```

Figure 2

```
Palindromer App

Enter the number of palindromes to store (10 max): 1
Enter a phrase: Doc, note: I dissent. A fast never prevents a fatness. I diet on cod

"Doc, note: I dissent. A fast never prevents a fatness. I diet on cod" is a palindrome

Number of palindromes: 1
```

Figure 3

Solution Design: The program has one class, Palindromer, with a main method and five other methods as shown in the class diagram. The main method prompts the user for the number of palindromes to be entered. After an error check, it calls inputPalindromes( ) to enter that many phrases into Palindromer's internal String array. It then calls displayPalindromes( ), which should examine each cleaned-up phrase to see if it's a palindrome; it does this using the helper methods cleanString( ) – this method is coded for you – and isPalindrome( ); it displays the phrases, one by one, and whether or not they are palindromes, and counts the number of phrases that are palindromes. Finally, the main method displays that count. See Palindromer.java for comments about the specifications.



Hints: The String class has the method toCharArray( ) that returns an array of characters stored in the String object. It also has the nextLine( ) method to input an entire line of text (including spaces), so use that for getting user input. The Character wrapper class has a method Character.isLetter(char) that returns true if its parameter is a letter. The `StringBuilder` class has several useful methods for this assignment, including a reverse( ) method.

Instructions:

1. Download the files Palindromer.java and TestPalindrome.java from Canvas. You'll have to add junit5 before anything works (even if you just want to run Palindromer's main). Just right-click on any of the code in TestPalindrome, and IntelliJ will give you some options (don't choose junit4, though).
2. Create a project named 4and copy the .java files into the src directory.
3. Fill in your code for the missing methods. Do not change the main method. Do not change the method signatures.
4. Test your program using testPalindrome.java, which contains 10 test cases. Do not change this file.
To run the test cases, right-click on testPalidrome.java and choose "Run TestPalindrome.java". This will run the test cases only – not the main program. You'll see green checkmarks for the passed tests and red X's for the failed tests. Also, running the test cases sets the default for the green Run arrow to this file, so if you want to run the main program, use the drop-down box next to the green Run arrow to choose Palindromer (or right-click on Palindromer.java in the Project window).

5. Make sure your console output comes out as shown in the Figures.
6. Write your Andrew ID and name as a comment one the first line of Palindromer.java.
7. Submit your solution it to Canvas.