Assignment #2

Due Date October 30

CIT 2266 Object Oriented Programming

Objective:

- To develop skills manipulating strings and becoming familiar with the String class
- Converting primitive type to Objects
- Converting Object to primitive types
- Converting one data type to others

Directions:

Design a class named **MyString** that implements the following methods

1. Write a **static** method **isSubString** in the class **MyString** that searches for a specific string (could be a character or a word) within another string; the method must return true if the former exists in the latter string otherwise the method must return false.

   For example the following calls will return true

   ```
   MyString.isSubString("cat","the cat in the hat.")
   MyString.isSubString("he cat","the cat in the hat.")
   MyString.isSubString("at.","the cat in the hat.")
   ```

   while the following will return false:

   ```
   MyString.isSubString("bat","the cat in the hat.")
   MyString.isSubString("mouse","the cat in the hat.")
   MyString.isSubString("at!","the cat in the hat.")
   ```

2. Write a **non-static** method **isSubString** in the class **MyString** that searches for a specific string (could be a character or a word) within the object's data; the method must return true if the argument is in the object's data otherwise the method must return false.

   For example the following calls will return true

   ```
   MyString sentence = new MyString("the cat in the hat.");
   sentence.isSubString("cat");
   sentence.isSubString("he cat") ;
   bool found = new MyString("the cat in the hat.").isSubString("at.");
   ```

   while the following will return false:

   ```
   sentence.isSubString("bat")
   sentence.isSubString("mouse")
   sentence.isSubString("at!")
   ```

Assignment #2

Due Date October 30

CIT 2266 Object Oriented Programming

---

3. A palindrome is a number or a text phrase that reads the same backward as forward. Write a family of overloaded **static** methods (a minimum or four methods) in the class **MyString** that will check if the argument is a palindrome. Note: The method that handles strings should ignore anything but alphabetic characters and numeric digits.

   For example the following calls will return true

   ```
   isPalindrome("12321")
   isPalindrome(12321)
   isPalindrome(123.21)
   isPalindrome("Niagara. O roar again!")
   isPalindrome("Dot saw I was Tod")
   isPalindrome("A man, a plan, a Canal; Panama");
   ```

   Be sure to sufficiently test your methods and demonstrate the methods functionality in the driver class (the one that defines main).

4. Create a method named **upConversion(byte)** in the class **MyString** that takes a byte as an argument and converts the value to the following primitive data types: short, int, long, float, double and to the object types: Short, Integer, Long, Float and Double. Make sure to print to the console the values of both the primitive types and the objects.

5. Create a method named **downConversion(Double)** in the class **MyString** that takes a Double object as an argument and converts the value to the following primitive data types: short, int, long, float, double and to the object types: Short, Integer, Long, Float and Double. Make sure to print to the console the values of both the primitive types and the objects.

   Note: For questions #4 and #5 determine the best style of method to write.
   (i.e. class method or an object method)

6. Create a Java application to thoroughly test the MyString class that you have made.

Upload your completed project to Assignment/CIT/CIT2266/Assign2