Introductory Java Programming

Queen Mary University of London
School of Electronic Engineering and Computer Science

Course Code: EBU4201

Lab Sheet 7: Strings and Maths

---

1. Parity is a simple method of error detection used for binary numbers. A parity bit can be added to a bit pattern to give odd or even parity. For *odd parity*, both the number of ones and the number of zeros must be odd (i.e., **1**, **3**, **5**, **7**, etc). For *even parity*, both the number of ones and the number of zeros should be even. **Figure 1** shows two examples of 7-bit inputs and their corresponding 8-bit output, according to the parity type.

	Example 1	**Example 2**
input	1010011	1000101
with even parity bit added	01010011	11000101
with odd parity bit added	11010011	01000101

Figure 1

Write a Java program that takes in as argument a **`String`** representing a 7-bit pattern and an integer indicating the required type of parity (**`0 = even, 1 = odd`**) and outputs the new 8-bit binary number (i.e., consisting of 1 parity bit and a 7-bit number) as a **`String`**. **Figure 2** shows two examples of typical use and output for the program described above:

```
C:\> java ParityBitAdder 1010011 0
Adding even parity to '1010011' results in the binary pattern '01010011'.

C:\> java ParityBitAdder 1010011 1
Adding odd parity to '1010011' results in the binary pattern '11010011'.
```

Figure 2

i) General requirement of your Java program: It should check (without using Exception Hnadling) whether the program inputs given on the command line are correct before continuing; in other words, it should exit with a message if the number of inputs and/or their type is incorrect.

ii) Specific requirement of your Java program: It must contain at least one method[1] called **`calculateParity()`**.

Name this program **`ParityBitAdder.java`**

[1] **Note**: It will be useful to write other methods (for other parts of the application functionality), to achieve an "easier to test" implementation. You should not use Exception Handling in your coding solutions.

2. Write a Java application that takes in a **`String`** parameter and outputs the **`String`** that results from changing the capitalisation of the individual characters[2] in the **`String`** in the following way:

   i) All *vowels* must be in lower case; for this purpose, vowels are the characters **`a`**, **`e`**, **`i`**, **`o`**, **`u`** and **`y`**.

   ii) All *consonants* must be in upper case; a consonant is any letter that is not a vowel.

   iii) Any characters that are not letters should be converted to the character **`*`**.

   Your application should store the new content after modifications, by effectively overriding any characters in the original **`String`** that are changed because of the conditions *i)-iii)* above.[3]

   Hint: It will be useful to make use of the **`String`** and **`Character`** classes, and their methods.[4]

   Figure 3 shows an example of typical use and output of the program described above:

```
C:\> java StringConverter "My car Goes verY FAST!"

Output: My*CaR*GoeS*VeRy*FaST*
```

Figure 3

   Name this program **`StringConverter.java`**

3. Write a Java application which keeps rolling three standard dice at the same time (i.e., it generates three random integers between **`1`** and **`6`**, *inclusive*), and stops only when all three dice display the number **`6`**. Your application should also display the number of times that the three dice were rolled.

   Name this program **`ThreeRollingDice.java`**

Ensure all your programs contain both internal comments and *Javadoc* comments.

[2] **Note**: Remember that, unlike **`String`**, characters are not objects (unless you use their object representation with the **`Character`** class).

[3] It will be useful to divide the functionality of your application into several methods, to achieve an "easier to test" implementation.

[4] Look up the Java API for these classes (and their methods) at:
- https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/String.html
- https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/Character.html