Homework #4

Due: **10/29/18 at 10:00 PM**
Late Submissions: **No Late Submissions**
Program Files (2): **BinaryFormatException.java, NumberConvertor.java**

Since computers store everything in terms of sequences of 0’s and 1’s, computer scientists often use a base-2 number system. We represent numbers in this system using only two digits, 0 and 1 and call them “binary numbers.” This is different from the base-10 system that modern humans usually use, which has ten distinct digits. Instead of units, ten, hundreds, and thousands places, binary numbers have units, two, fours, and eights places. The binary number 10001 is the same as the decimal 17 ($1\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0$). For this assignment, you will write a simple program that can convert decimal numbers into binary numbers and vice versa. Understanding how to convert numbers is an important component of this assignment, so you are **not allowed** to use the parseInt method from the Integer class. This assignment will also test your ability to declare, create, throw, and handle exceptions.

The details of the two classes are given in UML form below. Please pay close attention to the details of the specification when writing your program.

BinaryFormatException	Subclass of NumberFormatException
-badChar:char	A character that is not a ‘0’ or ‘1’ that caused the exception
-charPos:int	The index of the invalid character in the string.
+BinaryFormatException(charPos:int, badChar:char)	Create a new exception object with given index and invalid character. Sets the message to “Binary numbers consist only of 0's and 1's”
+getBadChar():char	Get method for **badChar**
+getCharPos():int	Get method for **charPos**

NumberConvertor	
+binaryToDecimal(binText:String):int	Given a binary number parameter in string form, returns the integer equivalent. If any of the characters are not ‘0’ or ‘1’, throws a **BinaryFormatException** for the first such character.
+decimalToBinary(decInt:int):String	Given a nonnegative integer parameter, returns a string that is the equivalent binary number.
+main(args:String[]):void	See below.

Your main method should begin by asking the user to enter a choice of **B**,**D**, or **Q**. B stands for “Translate Binary to Decimal”, D stands for “Translate Decimal to Binary” and Q stands for “Quit.” The lower-case version of each option should work just like the upper-case version.

If **B** (or **b**), is chosen, the system asks the user to “Enter a binary number:”. If the number is not valid, it will print “Character *char* at index *index* is not a valid binary digit” for the first *char* in the user’s string that is not a 0 or 1. If the number is valid, it prints “The decimal equivalent is *d*.” This should be implemented by invoking the **binaryToDecimal** method and exception handling.

If **D** (or **d**) is chosen, the system asks the user to "Enter a nonnegative decimal integer:" If they enter a string that is not an integer, the system prints "You did not enter a valid integer." If they enter a negative integer, it prints "That's a negative integer!" Otherwise, it should print "The binary equivalent is *b*." This should be implemented by invoking the **decimalToBinary** method and exception handling.

After **B** or **D** is chosen, regardless if the requested input was valid, the system asks for the user to enter another choice. It will continue to do this until the user enters **Q**. When Q is entered, the system prints "Goodbye!" and exits. If at any time the user enters a choice other than B,D or Q, the system prints "*entry* is not a valid option" and asks for another choice.

Here is sample of what your output should look like (user input is in **bold**):

```
> run NumberConvertor
Enter a choice (B,D, or Q): B
Enter a binary number: 1011
The decimal equivalent is 11

Enter a choice (B,D, or Q): b
Enter a binary number: 2
Character 2 at index 0 is not a valid binary digit

Enter a choice (B,D, or Q): d
Enter a nonnegative decimal integer: 44
The binary equivalent is 101100

Enter a choice (B,D, or Q): D
Enter a nonnegative decimal integer: -67
That's a negative integer!

Enter a choice (B,D, or Q): D
Enter a nonnegative decimal integer: five
You did not enter a valid integer.

Enter a choice (B,D, or Q): X
X is not a valid option.

Enter a choice (B,D, or Q): Q
Goodbye!
```

At a minimum provide a Javadoc comment explaining what each method does, and another one that summarizes the class. Remember, Javadoc comments start with the characters "/**" and end with "*/". Include additional comments for lines that are especially complicated. At the top of the program include the standard comment with the class, your name, your user id, "Homework #4", the deadline, and a short description of the program.

Once the program compiles, runs, and has been tested to your satisfaction, upload both .java files to Course Site. To do so, click on the name of the assignment in the Course Site page, and then press the "Add submission" button at the bottom of the next page. Drag and drop each file into the area under "File submissions". If necessary, you can update your submission at any time before the deadline passes. Be very careful to ensure that you named the class and files correctly, including using the correct case for all letters in the names.