# Please review the following code

The codes intent is to have a class that takes a number and returns the roman numberal string.

You task is to code review the code and identify issues that you would like to change keeping in mind coding best practices.

package converters;  
  
public class RomanNumber  
{  
 private int number;  
  
 private static final int maxValue = 3000;  
 public RomanNumber(int number) throws Exception

Sonar comment: Define and throw a dedicated exception instead of using a generic one.    
 {  
 if(number > maxValue) throw new Exception("RomanNumber only supports numbers up to 3000");

Sonar comment: Define and throw a dedicated exception instead of using a generic one.  
 this.number = number;  
 }  
  
 public void SetNumber(String number) throws Exception

Sonar comment: Define and throw a dedicated exception instead of using a generic one.   
 {  
 this.number = Integer.parseInt(number);  
 if(this.number > maxValue) throw new Exception("RomanNumber only supports numbers up to 3000");

Sonar comment: Define and throw a dedicated exception instead of using a generic one.  
  
 }  
  
 public String convert()  
 {  
 String result = "";  
  
 try {  
 int thousands = this.number / 1000;  
 result += times(thousands, "M");  
 int hundreds = this.number / 100 % 10;  
 result += times(hundreds, "C", "D", "M");  
 int tens = this.number / 10 % 10;  
 result += times(tens, "X", "L", "C");  
 int ones = this.number % 10;  
 result += times(ones, "I", "V", "X");  
 } catch (Exception ex){

Sonar comment: Either log or rethrow this exception  
 System.out.println("An error occured");

Sonar comment: Replace this usage of System.out or System.err by a logger.   
 }  
 }  
  
 private String times(int number, String character)

Sonar comment: Remove this unused method parameter "number"  
 {  
 String result = "";  
  
 for(int i = 0; i < this.number; i++)  
 {  
 result += character;  
 }  
 return result;  
 }  
 private String times(int number, String o, String f, String t) throws Exception

Sonar comment: The Cyclomatic Complexity of this method "times" is 17 which is greater than 10 authorized.

Sonar comment: Define and throw a dedicated exception instead of using a generic one  
 {  
 switch(number)  
 {  
 case 0:  
 return "";  
 case 1:  
 case 2:  
 case 3:  
 return times(number, o);  
 case 4:  
 return o + f;  
 case 5:  
 case 6:  
 case 7:  
 case 8:  
 return f + times(number - 5, o);  
 case 9:  
 return o + t;  
 default:  
 throw new Exception("Only single digits allowed - not " + number);

Sonar comment: Define and throw a dedicated exception instead of using a generic one.   
 }  
 }  
}

**Some code venerability**

1. **public void SetNumber(String number) throws Exception**

SetNumber(String number) – There is no requirement of this method. Actually we are passing the input as a Number in class constructor which set the value to instance variable number.

1. public RomanNumber(int number) throws Exception

The class constructor is throwing the exception, Actually it is not a good as java coding standard compliance. We should throw our own custom exception and handle it.

1. Change required in times(int number, String character)

Remove this unused method parameter "number". Because this method using this.number in for loop which is global variable.

It will not give the correct roman no result if we use the global number variable with this keyword.

**private** String times(**int** number, String character)

{

String result = "";

**for** (**int** i = 0; i < **this**.number; i++) {

result += character;

}

**return** result;

}