# 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.

First is compilation error: convert method must return result of type string

package converters;  
  
public class RomanNumber  
{  
 private int number;

//Sonar Code Review Comment : Rename this constant name to match the regular expression '^[A-Z][A-Z0-9]\*(\_[A-Z0-9]+)\*$'.  Like MAX\_VALUE  
 private static final int maxValue = 3000;  
 public RomanNumber(int number) throws Exception

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

//Sonar Code Review Comment : Define and throw a dedicated exception instead of using a generic one.  
//Sonar Code Review Comment : Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]\*$'.  Method name starts with lower case  
 {  
 this.number = Integer.parseInt(number);  
 if(this.number > maxValue) throw new Exception("RomanNumber only supports numbers up to 3000");  
 //Sonar Code Review 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 Code Review Comment: Either log or rethrow this exception.  
 System.out.println("An error occured");

//Sonar Code Review Comment: Replace this usage of System.out or System.err by a logger.   
 }

// Return statement is missing  
 }  
  
 private String times(int number, String character)

//Sonar Code Review Comment: Remove this unused method parameter "number". Because this method using this.number not local variable  
 {  
 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 Code Review Comment: Define and throw a dedicated exception instead of using a generic one.

//Sonar Code Review Comment: The Cyclomatic Complexity of this method "times" is 17 which is greater than 10 authorized.  
 {  
 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 Code Review Comment: Define and throw a dedicated exception instead of using a generic one  
 }  
 }  
}

Below Issues and Recommendation after code review of the below code, which needs to be resolved before the Code Release.

1. Convert () method is not returning any String where it's signature says, there must be a String returned. So, return statement must be added for this method.
2. it has been observed that all the methods are throwing Generic Exception where it should be custom exception or specific exception
3. String concatenation should be mandatorily avoided and instead, the String Buffer / String Builder should be used.
4. Methods and variables names have no proper naming. All method and variable names should be descriptive enough following the standard Java Naming Convention for proper code readability.
5. There is no comment inside the code. Code should be properly commented wherever it is needed.
6. the variable name should be changed to follow Java naming convention for constant field. could be like 'MAX\_VALUE' and this static filed is to be accessed in static way using class Reference like RomanNumber.MAX\_VALUE. Also, we might think of using ENUM instead of constant of integer.

private static final int maxValue = 3000; // Original line of code

1. SetNumber() method is not a recommended setter method as per the standard Java naming convention, so it might be renamed as setNumber(String number) instead of SetNumber During Integer.parseInt(number) the NumberFormatException is to be handled properly and specifically. Exception Should be handled First and then the value is to be initialized in this method. Static filed is to be accessed in static way using class Reference like RomanNumber.MAX\_VALUE.

Below method must adhere above recommendation

public void SetNumber(String number) throws Exception // Original line of code

1. The Switch case should have break statement to execute specific case based on the need, not all the case. Case block should not return a value directly rather one single return statement could be used for the whole method. The Switch case block in the below method is to change and handled properly

private String times(int number, String o, String f, String t) throws Exception // Original line of code

9) The codes intent is to have a class that takes a number and returns the roman number as string. So, the class can be enhanced to Singleton pattern.

10) There should be proper unit test cases written covering all the Line of Codes.