Detailed Explanation of Java Code for Roman Numeral Conversion

Diana Akolzina

December 29, 2023

1 Introduction

This document provides an in-depth explanation of the Java code for converting Roman numerals to integers and vice versa, focusing on the specific features of Java, data structures, and algorithms used.

2 The RomanNumerals Class

The code defines a class named `RomanNumerals` in Java, encapsulating the functionality for Roman numeral conversion.

2.1 Class Members

The class contains a static final member variable `intToRomanMap` used for integer to Roman numeral conversion.

```
private static final LinkedHashMap<Integer, String>
                        intToRomanMap = new LinkedHashMap<>();
static {
    intToRomanMap.put(1000, "M");
    intToRomanMap.put(900, "CM");
    intToRomanMap.put(500, "D");
    ... // More entries
}
```

Detailed Explanation:

- `intToRomanMap`: A LinkedHashMap that maintains the insertion order. It maps integers to their corresponding Roman numeral strings.
- The static block initializes this map with Roman numeral values, ensuring they are stored in descending order.

2.2 Method: romanToInt

Converts a Roman numeral string to an integer.

```
public static int romanToInt(String s) {
    Map<Character, Integer> romanToIntMap =
                Map.of('I', 1, 'V', 5, 'X', 10, 'L', 50,
                'C', 100, 'D', 500, 'M', 1000);
    int total = 0, prevValue = 0;
    for (int i = s.length() - 1; i >= 0; i--) {
        int value = romanToIntMap.get(s.charAt(i));
        total = value < prevValue ? total - value : total + value;
        prevValue = value;
    }
    return total;
}
```

Detailed Explanation:

1. The method defines a local map `romanToIntMap` for the conversion from Roman numerals to integers.
2. It iterates over the input string `s` in reverse order.
3. For each character, the method gets the corresponding integer value and updates the `total` based on the Roman numeral rules.
4. The ternary operator is used to decide whether to add or subtract the value from the total.
5. The method returns the accumulated `total`.

2.3 Method: intToRoman

Converts an integer to a Roman numeral string.

```
public static String intToRoman(int num) {
    StringBuilder roman = new StringBuilder();
    for (Map.Entry<Integer, String> entry : intToRomanMap.entrySet()) {
        while (num >= entry.getKey()) {
            roman.append(entry.getValue());
            num -= entry.getKey();
        }
    }
    return roman.toString();
}
```

Detailed Explanation:

1. Initialize a `StringBuilder` named `roman`.

2. Iterate over the entries in `intToRomanMap`.

3. For each entry, append the Roman numeral to `roman` as long as the `num` is greater than or equal to the entry's key.

4. Decrease `num` by the entry's key value each time the corresponding Roman numeral is appended.

5. Convert `roman` to a string and return it.