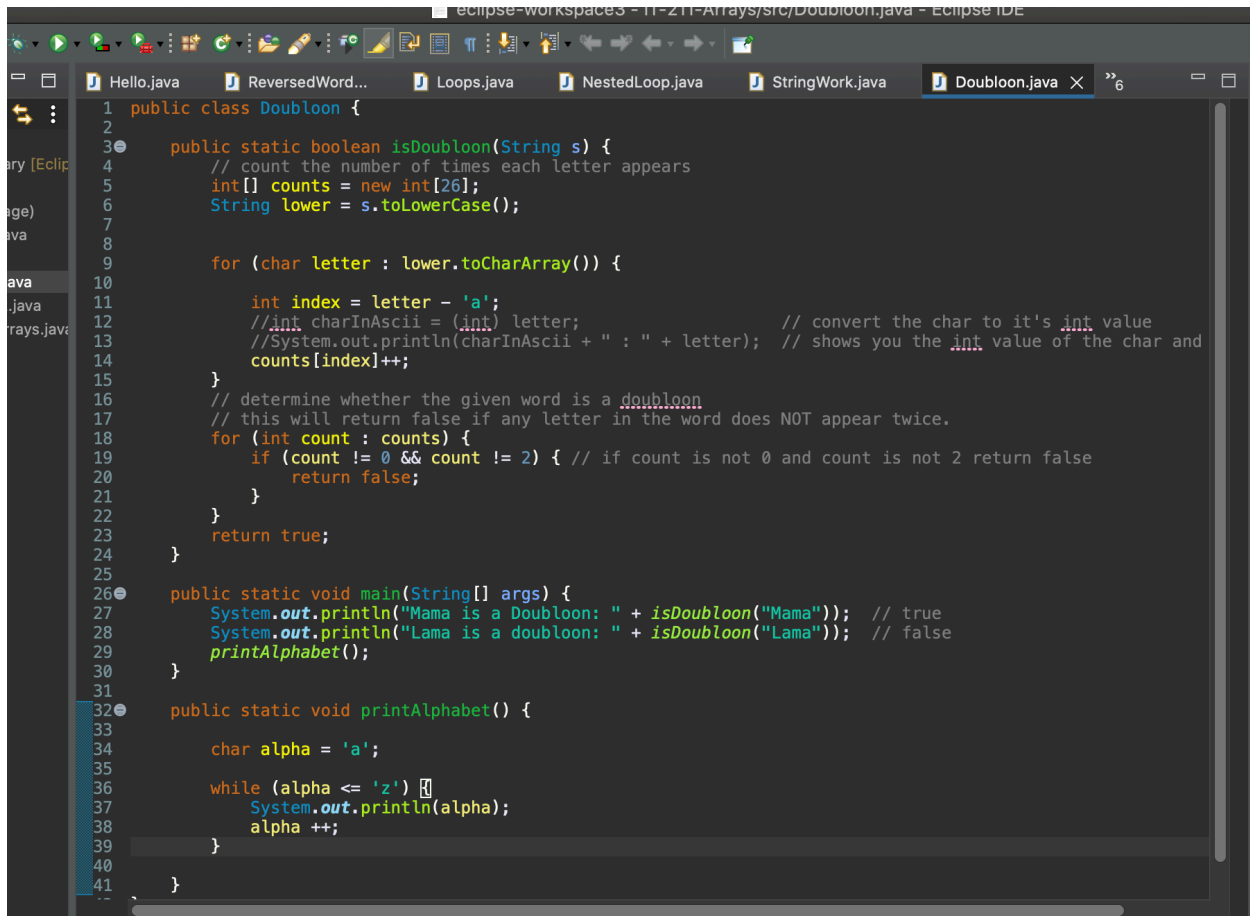


Doubloon Breakdown

Let's look at the state of the variables as we go through in each part! We'll use the example inputs of Mama & Lama that are used in the main method.



```
1 public class Doubloon {
2
3     public static boolean isDoubloon(String s) {
4         // count the number of times each letter appears
5         int[] counts = new int[26];
6         String lower = s.toLowerCase();
7
8         for (char letter : lower.toCharArray()) {
9
10             int index = letter - 'a';
11             //int charInAscii = (int) letter; // convert the char to it's int value
12             //System.out.println(charInAscii + " : " + letter); // shows you the int value of the char and
13             counts[index]++;
14         }
15         // determine whether the given word is a doubloon
16         // this will return false if any letter in the word does NOT appear twice.
17         for (int count : counts) {
18             if (count != 0 && count != 2) { // if count is not 0 and count is not 2 return false
19                 return false;
20             }
21         }
22         return true;
23     }
24
25     public static void main(String[] args) {
26         System.out.println("Mama is a Doubloon: " + isDoubloon("Mama")); // true
27         System.out.println("Lama is a doubloon: " + isDoubloon("Lama")); // false
28         printAlphabet();
29     }
30
31     public static void printAlphabet() {
32         char alpha = 'a';
33
34         while (alpha <= 'z') {
35             System.out.println(alpha);
36             alpha ++;
37         }
38     }
39
40 }
41 }
```

Breakdown:

Entering the method, setting the counts integer array, populating the lower variable with the contents of the string the user passed in, set to lower case (so it won't have weird char values)

Mama:

String s: **Mama**

String lower: **mama** (because it is equal to s.toLowerCase())

Integer array **counts**:

Index 0 : 0

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0

Index 8 : 0

Index 9 : 0

Index 10 : 0

Index 11 : 0

Index 12 : 0

Index 13 : 0

Index 14 : 0

Index 15 : 0

Index 16 : 0

Index 17 : 0

Index 18 : 0

Index 19 : 0

Index 20 : 0

Index 21 : 0

Index 22 : 0

Index 23 : 0

Index 24 : 0

Index 25 : 0 (note this is position 26 because we start counting at 0)

Lama:

String **s**: **Lama**

String **lower**: **lama** (because it is equal to s.toLowerCase())

Integer array **counts**:

Index 0 : 0

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : 0
Index 12 : 0
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

```
public static boolean isDoubloon(String s) {  
    // count the number of times each letter appears  
    int[] counts = new int[26];  
    String lower = s.toLowerCase();  
  
    for (char letter : lower.toCharArray()) {  
        int index = letter - 'a';  
        //int charToAscii = (int) letter;  
        //counts[charToAscii]++;  
    }  
}
```

We enter the for loop and we get two new variables:

Char **letter** which will take on the value of every Char in the Char array created by converting **lower** into a Char array.

Int **index** which will take on the numeric value of **letter - 'a'** remember, you can do math to characters because they are equal to their ascii value. SO a when converted to an integer is equal to 97. All the letters of the alphabet after a are higher, which means that a - a will equal 0, while b - a will equal 1, z - a will equal 25, this is important for the next part!!

Using the value of index to increment the counts integer array for the value of whatever letter was just matched

```

//int charInAscii = (int) letter; // co
//System.out.println(charInAscii + " : " + letter); // sh
counts[index]++;
}

```

K, so, here's where things are going to get complicated.

Mama:

String **s: Mama**

String **lower: mama** (because it is equal to s.toLowerCase())

(in the loop – pass 1 **mama**)

Char **letter: m**

Int **index: 12 (m - a = 12)**

Integer array **counts:**

Index 0 : 0

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0

Index 8 : 0

Index 9 : 0

Index 10 : 0

Index 11 : 0

Index 12 : **1** <- position 12 gets incremented to 1.

Index 13 : 0

Index 14 : 0

Index 15 : 0

Index 16 : 0

Index 17 : 0

Index 18 : 0

Index 19 : 0

Index 20 : 0

Index 21 : 0

Index 22 : 0

Index 23 : 0

Index 24 : 0

Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 2 mama)

Char **letter: a**

Int **index: 0 (a - a = 0)**

Integer array **counts:**

Index 0 : **1** <- position 0 gets incremented to 1.

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0

Index 8 : 0

Index 9 : 0

Index 10 : 0

Index 11 : 0

Index 12 : **1**

Index 13 : 0

Index 14 : 0

Index 15 : 0

Index 16 : 0

Index 17 : 0

Index 18 : 0

Index 19 : 0

Index 20 : 0

Index 21 : 0

Index 22 : 0

Index 23 : 0

Index 24 : 0

Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 3 mama)

Char **letter: m**

Int **index: 12 (m - a = 12)**

Integer array **counts:**

Index 0 : **1**

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : 0
Index 12 : **2** <- position 12 gets incremented to 2
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 4 mama)

Char **letter: a**

Int **index: 0 (a - a = 0)**

Integer array **counts:**

Index 0 : **2** <- position 0 gets incremented to 2

Index 1 : 0
Index 2 : 0
Index 3 : 0
Index 4 : 0
Index 5 : 0
Index 6 : 0
Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : 0
Index 12 : **2**
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0

Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

Lama:
String **s: Lama**
String **lower: lama** (because it is equal to s.toLowerCase())

(in the loop – pass 1 lama)

Char **letter: l**
Int **index: 11 (l - a = 11)**

Integer array **counts:**
Index 0 : 0
Index 1 : 0
Index 2 : 0
Index 3 : 0
Index 4 : 0
Index 5 : 0
Index 6 : 0
Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : **1** <- **position 11 gets incremented to 1**
Index 12 : 0
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0

Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 2)

Char **letter: a**
Int **index: 0 (a - a = 0)**

Integer array **counts:**

Index 0 : **1** <- **position 0 gets incremented to 1**
Index 1 : 0
Index 2 : 0
Index 3 : 0
Index 4 : 0
Index 5 : 0
Index 6 : 0
Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : **1**
Index 12 : 0
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 3 lama)

Char **letter: m**
Int **index: 12 (m - a = 12)**

Integer array **counts**:

Index 0 : **1**

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0

Index 8 : 0

Index 9 : 0

Index 10 : 0

Index 11 : **1**

Index 12 : **1** <- position 12 gets incremented to 1

Index 13 : 0

Index 14 : 0

Index 15 : 0

Index 16 : 0

Index 17 : 0

Index 18 : 0

Index 19 : 0

Index 20 : 0

Index 21 : 0

Index 22 : 0

Index 23 : 0

Index 24 : 0

Index 25 : 0 (note this is position 26 because we start counting at 0)

(in the loop – pass 3 lama)

Char **letter: a**

Int **index: 0 (a - a = 0)**

Integer array **counts**:

Index 0 : **2** <- position 0 gets incremented to 2

Index 1 : 0

Index 2 : 0

Index 3 : 0

Index 4 : 0

Index 5 : 0

Index 6 : 0

Index 7 : 0

Index 8 : 0

Index 9 : 0
Index 10 : 0
Index 11 : 1
Index 12 : 1
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

Iterating over the count integer array and searching for any values that are not 0 or 2. If it finds any, it returns false, because a doubloon must use all letters twice or not use them at all, if every one of the 26 letter counts passes this test, then it returns true.

```
// determine whether the given word is a doubloon
// this will return false if any letter in the word does NOT appear twice.
for (int count : counts) {
    if (count != 0 && count != 2) { // if count is not 0 and count is not 2 return false
        return false;
    }
}
return true;
```

Which means that Mama returns true, because there are only 0's and 2's:

Integer array **counts**:

Index 0 : 2
Index 1 : 0
Index 2 : 0
Index 3 : 0
Index 4 : 0
Index 5 : 0
Index 6 : 0
Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : 0

Index 12 : **2**
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0
Index 24 : 0
Index 25 : 0 (note this is position 26 because we start counting at 0)

And Lama returns false, because there are 0's, 2's, and 1's.

Integer array **counts**:

Index 0 : **2**
Index 1 : 0
Index 2 : 0
Index 3 : 0
Index 4 : 0
Index 5 : 0
Index 6 : 0
Index 7 : 0
Index 8 : 0
Index 9 : 0
Index 10 : 0
Index 11 : **1**
Index 12 : **1**
Index 13 : 0
Index 14 : 0
Index 15 : 0
Index 16 : 0
Index 17 : 0
Index 18 : 0
Index 19 : 0
Index 20 : 0
Index 21 : 0
Index 22 : 0
Index 23 : 0

Index 24 : 0

Index 25 : 0 (note this is position 26 because we start counting at 0)