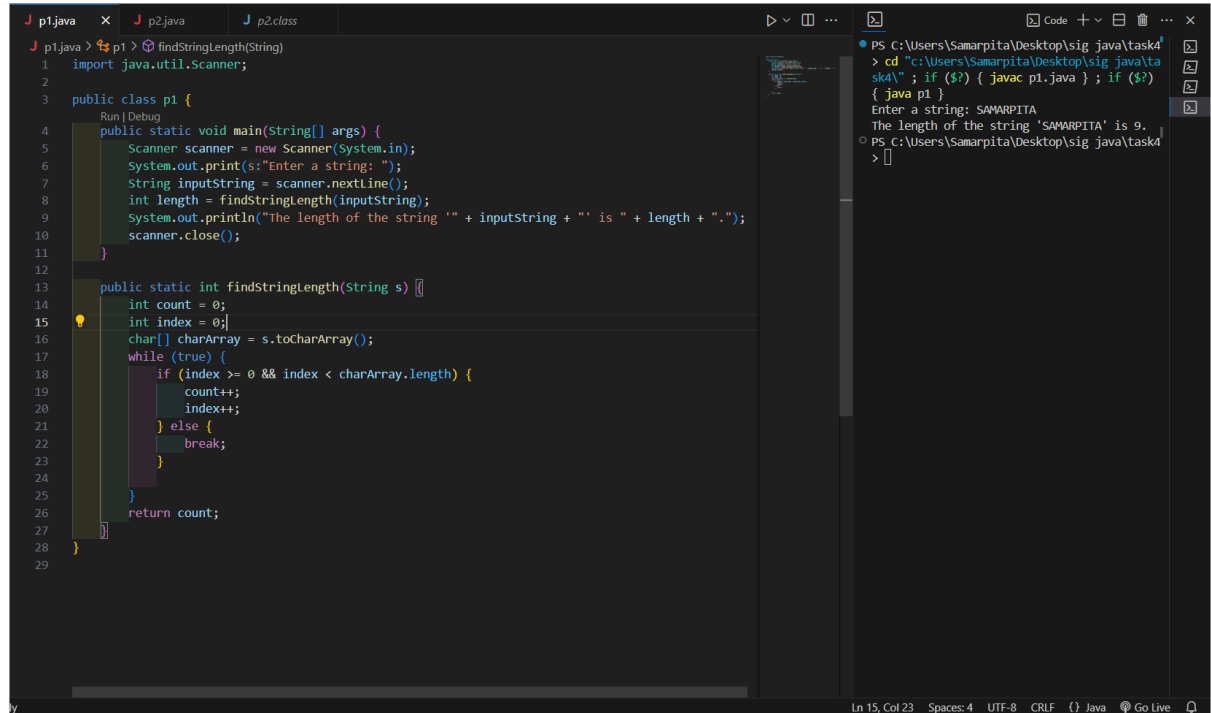


1. WAP to find the length of a string without using string functions.

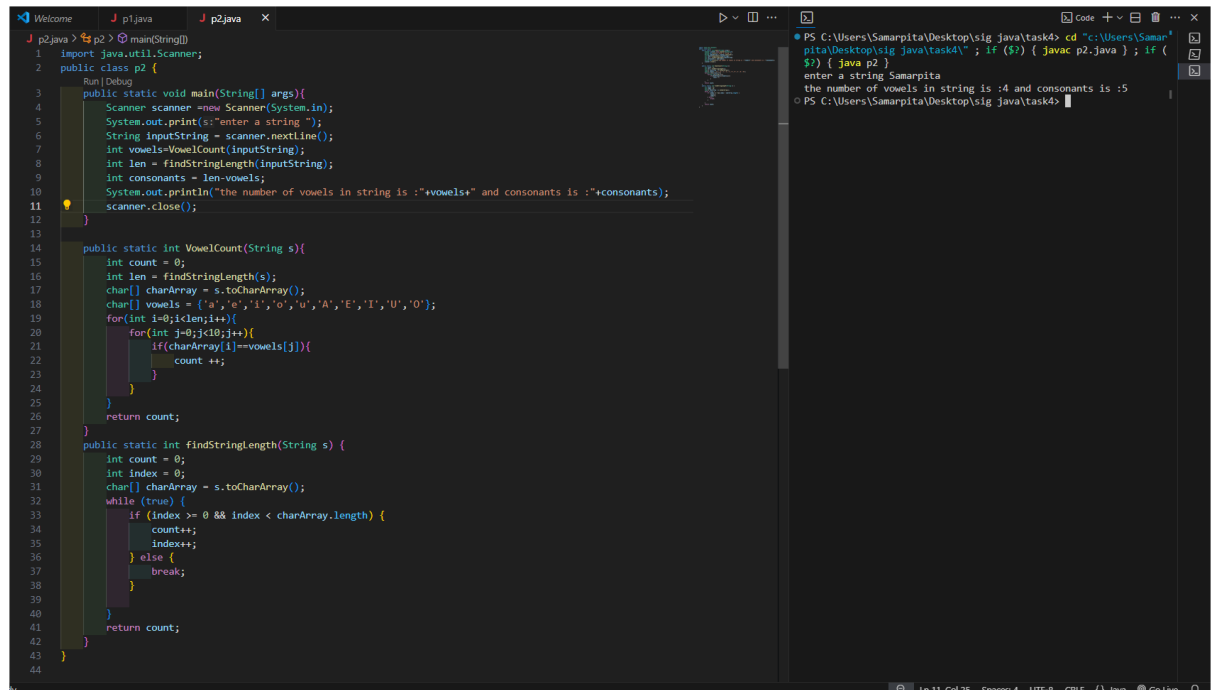


The screenshot shows an IDE with a Java file named `p1.java`. The code defines a class `p1` with a `main` method and a `findStringLength` method. The `main` method uses a `Scanner` to take input from the user and prints the length of the string. The `findStringLength` method uses a `while` loop to iterate through the characters of the string and count them.

```
1 import java.util.Scanner;
2
3 public class p1 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter a string: ");
7         String inputString = scanner.nextLine();
8         int length = findStringLength(inputString);
9         System.out.println("The length of the string '" + inputString + "' is " + length + ".");
10        scanner.close();
11    }
12
13    public static int findStringLength(String s) {
14        int count = 0;
15        int index = 0;
16        char[] charArray = s.toCharArray();
17        while (true) {
18            if (index >= 0 && index < charArray.length) {
19                count++;
20                index++;
21            } else {
22                break;
23            }
24        }
25        return count;
26    }
27 }
28
29
```

The terminal output shows the program running and the user entering "SAMARPITA". The output is: "The length of the string 'SAMARPITA' is 9."

2.WAP to find the number of vowels or consonants in a string.

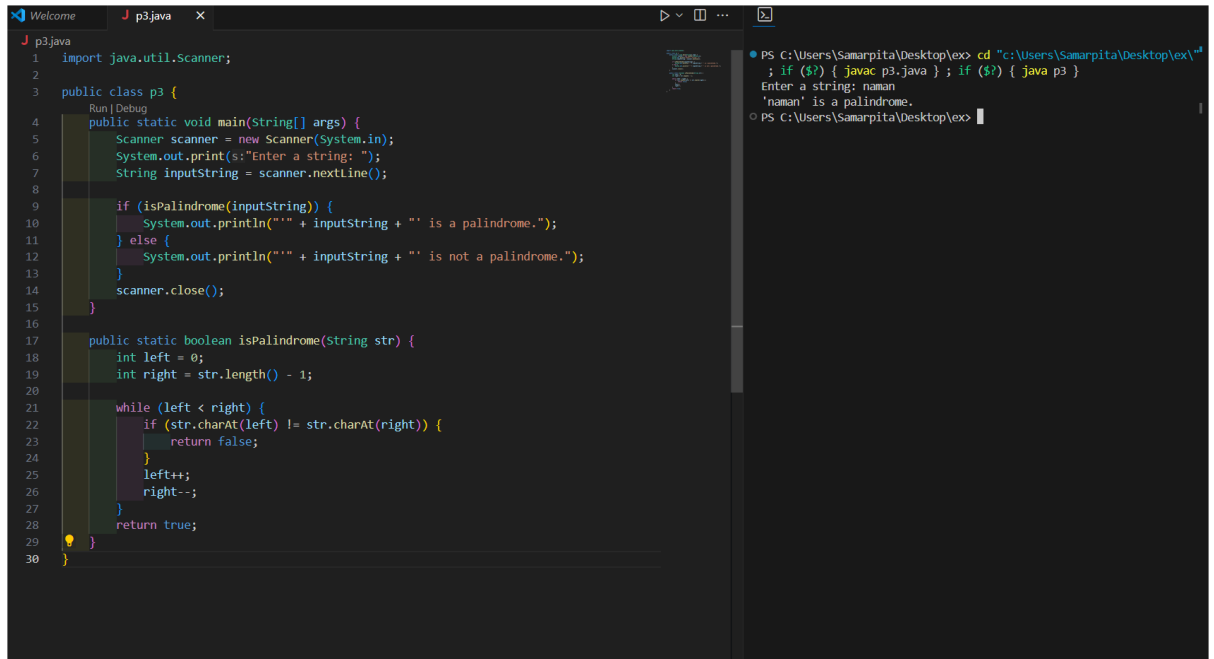


The screenshot shows an IDE with a Java file named `p2.java`. The code defines a class `p2` with a `main` method and two methods: `VowelCount` and `findStringLength`. The `main` method uses a `Scanner` to take input from the user and prints the number of vowels and consonants. The `VowelCount` method uses a `for` loop to iterate through the characters of the string and count the vowels. The `findStringLength` method uses a `while` loop to iterate through the characters of the string and count them.

```
1 import java.util.Scanner;
2 public class p2 {
3     public static void main(String[] args){
4         Scanner scanner =new Scanner(System.in);
5         System.out.print("enter a string ");
6         String inputString = scanner.nextLine();
7         int vowels=VowelCount(inputString);
8         int len = findStringLength(inputString);
9         int consonants = len-vowels;
10        System.out.println("the number of vowels in string is :"+vowels+" and consonants is :"+consonants);
11        scanner.close();
12    }
13
14    public static int VowelCount(String s){
15        int count = 0;
16        int len = findStringLength(s);
17        char[] charArray = s.toCharArray();
18        char[] vowels = {'a','e','i','o','u','A','E','I','U','O'};
19        for(int i=0;i<len;i++){
20            for(int j=0;j<10;j++){
21                if(charArray[i]==vowels[j]){
22                    count ++;
23                }
24            }
25        }
26        return count;
27    }
28    public static int findStringLength(String s) {
29        int count = 0;
30        int index = 0;
31        char[] charArray = s.toCharArray();
32        while (true) {
33            if (index >= 0 && index < charArray.length) {
34                count++;
35                index++;
36            } else {
37                break;
38            }
39        }
40        return count;
41    }
42 }
43
44
```

The terminal output shows the program running and the user entering "Samarpita". The output is: "the number of vowels in string is :4 and consonants is :5"

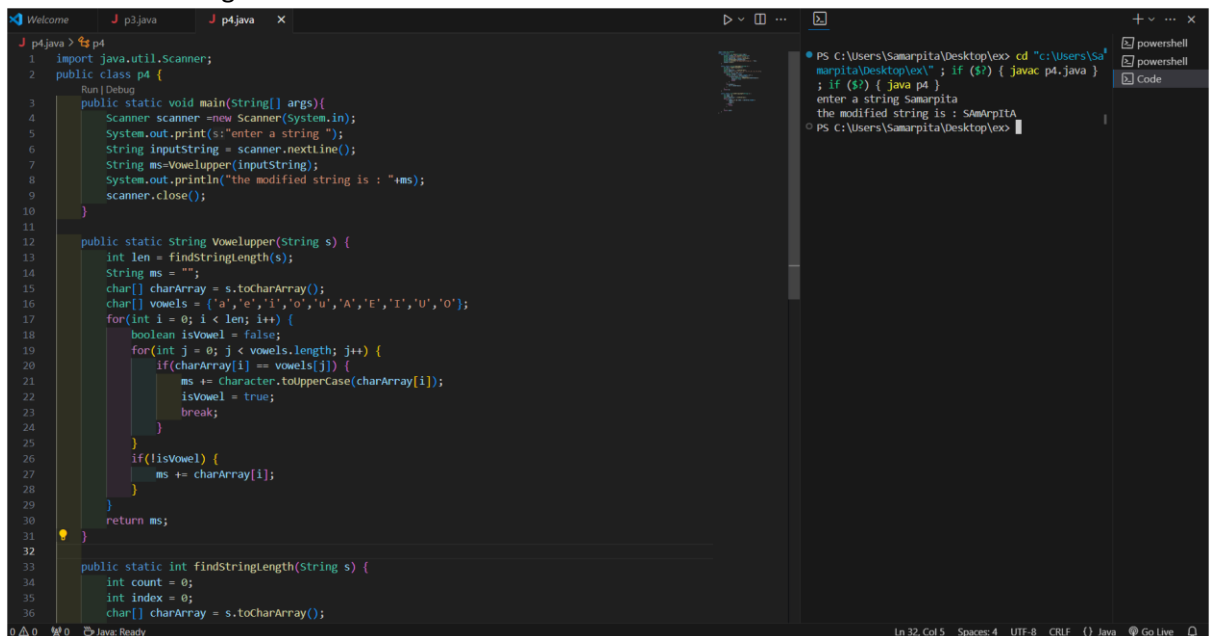
3.WAP to check whether a string is Palindrome string or not.



```
1 import java.util.Scanner;
2
3 public class p3 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print(s:"Enter a string: ");
7         String inputString = scanner.nextLine();
8
9         if (isPalindrome(inputString)) {
10             System.out.println("'" + inputString + "' is a palindrome.");
11         } else {
12             System.out.println("'" + inputString + "' is not a palindrome.");
13         }
14         scanner.close();
15     }
16
17     public static boolean isPalindrome(String str) {
18         int left = 0;
19         int right = str.length() - 1;
20
21         while (left < right) {
22             if (str.charAt(left) != str.charAt(right)) {
23                 return false;
24             }
25             left++;
26             right--;
27         }
28         return true;
29     }
30 }
```

PS C:\Users\Samarpita\Desktop\ex> cd "c:\Users\Samarpita\Desktop\ex\"
; if (\$?) { javac p3.java }; if (\$?) { java p3 }
Enter a string: naman
'naman' is a palindrome.
PS C:\Users\Samarpita\Desktop\ex>

4. WAP to convert all vowels present in a string into uppercase characters and print the modified string.



```
1 import java.util.Scanner;
2 public class p4 {
3     public static void main(String[] args){
4         Scanner scanner = new Scanner(System.in);
5         System.out.print(s:"enter a string ");
6         String inputString = scanner.nextLine();
7         String ms=Vowelupper(inputString);
8         System.out.println("the modified string is : "+ms);
9         scanner.close();
10    }
11
12    public static String Vowelupper(String s) {
13        int len = findStringLength(s);
14        String ms = "";
15        char[] charArray = s.toCharArray();
16        char[] vowels = {'a','e','i','o','u','A','E','I','O','U'};
17        for(int i = 0; i < len; i++) {
18            boolean isVowel = false;
19            for(int j = 0; j < vowels.length; j++) {
20                if(charArray[i] == vowels[j]) {
21                    ms += Character.toUpperCase(charArray[i]);
22                    isVowel = true;
23                    break;
24                }
25            }
26            if(!isVowel) {
27                ms += charArray[i];
28            }
29        }
30        return ms;
31    }
32
33    public static int findStringLength(String s) {
34        int count = 0;
35        int index = 0;
36        char[] charArray = s.toCharArray();
37    }
```

PS C:\Users\Samarpita\Desktop\ex> cd "c:\Users\Samarpita\Desktop\ex\"
; if (\$?) { javac p4.java }; if (\$?) { java p4 }
enter a string samarpita
the modified string is : Samarpita
PS C:\Users\Samarpita\Desktop\ex>

5. What is StringBuilder class, and how is it different from String?

- StringBuilder is mutable, meaning its content can be modified after creation, while String is immutable, meaning its content cannot be changed.

- `StringBuilder` provides better performance and memory efficiency for string manipulation operations like concatenation, insertion, or deletion due to its mutability.
- `StringBuilder` is not thread-safe, whereas `String` is inherently thread-safe because of its immutability.
- Use `StringBuilder` when you need to perform frequent modifications on strings, and use `String` when immutability is desired or when the string value remains constant.