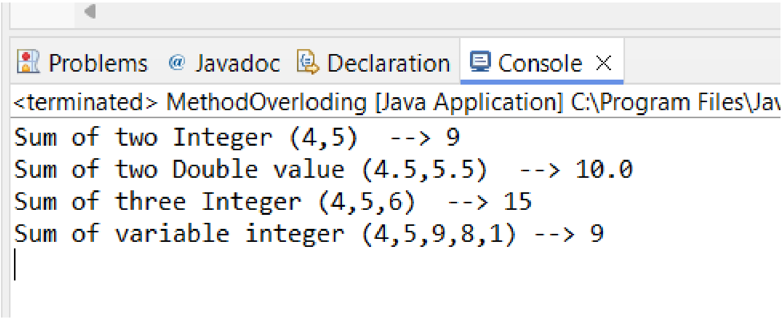
1. Method Overloading: Write a class Calculator with overloaded methods add(). Implement add() methods that take:

- Two integers
- Two double values
- Three integers
- A variable number of integers

CODE:-

```
package Lab_4;

public class MethodOverloding {

    public int add(int a,int b)    //method to add two integer value
    {
        return a+b;
    }

    public double add(double a,double b)    //method to add to double value
    {
        return a+b;
    }

    public int add(int a,int b,int c)    //method to add three integer value
    {
        return a+b+c;
    }

    public int add(int... numbers)  // method to add variable no.of Integer
    {
        int sum=0;
        for(int number : numbers)
        {
            sum += number;
        }
        return sum;
    }
    public static void main(String[] args) {
        MethodOverloding m1=new MethodOverloding(); //objection creation of class


        System.out.println("Sum of two Integer (4,5) "+" --> "+m1.add(4,5));
        System.out.println("Sum of two Double value (4.5,5.5) "+" --> "
+m1.add(4.5,5.5));
        System.out.println("Sum of three Integer (4,5,6) "+" --> " +m1.add(4,5,6));
        System.out.println("Sum of variable integer (4,5,9,8,1)"+" --> " +m1.add(4,5));
    }


}
```

OUTPUT:-



<terminated> MethodOverloding [Java Application] C:\Program Files\Ja

```
Sum of two Integer (4,5)  --> 9
Sum of two Double value (4.5,5.5)  --> 10.0
Sum of three Integer (4,5,6)  --> 15
Sum of variable integer (4,5,9,8,1) --> 9
```

2. Super Keyword: Create a class Person with a constructor that accepts and sets name and age.

- Create a subclass Student that adds a grade property and initializes name and age using the super keyword in its constructor.

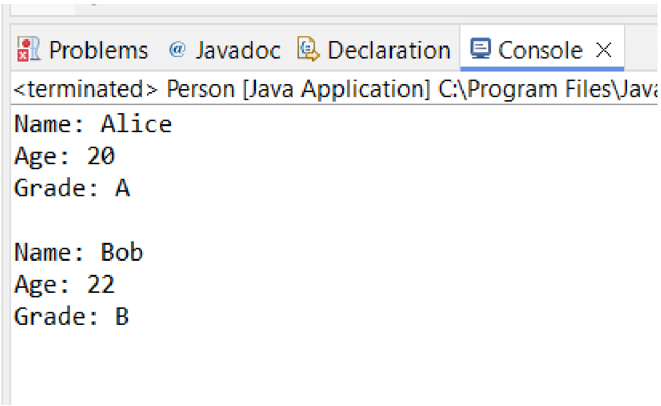
- Demonstrate the creation of Student objects and the usage of super to call the parent class constructor.

CODE:-

```
package Lab_4;


class Person1{   //Define the Person class


        private String name;    // Properties of the Person class

        private int age;


        // Constructor of the Person class

        public Person1(String name, int age) {

                this.name = name;

                this.age = age;

        }


        // Getter method for name

        public String getName() {

                return name;

        }


        // Getter method for age

        public int getAge() {

                return age;

        }
}

//Define the Student class which is a subclass of Person
class Student extends Person1 {
    // Additional property for the Student class
    private String grade;

    // Constructor of the Student class
    public Student(String name, int age, String grade) {
        // Call the constructor of the Person class using super
        super(name, age);
        // Initialize the grade property
        this.grade = grade;
    }

    // Getter method for grade
    public String getGrade() {
        return grade;
    }

    // Method to display student details
    public void displayStudentDetails() {
        System.out.println("Name: " + getName());
        System.out.println("Age: " + getAge());
        System.out.println("Grade: " + getGrade());
    }
}
```

```
//Main class to demonstrate the creation of Student objects
public class Person {
        public static void main(String[] args) {
                // Create a Student object
                Student stu1 = new Student("Alice", 20, "A");
                // Display the details of the student
                stu1.displayStudentDetails();

                System.out.println();

                // Create another Student object
                Student stu2 = new Student("Bob", 22, "B");
                // Display the details of the student
                stu2.displayStudentDetails();
        }
}
```

OUTPUT:-



```
<terminated> Person [Java Application] C:\Program Files\Javi
Name: Alice
Age: 20
Grade: A

Name: Bob
Age: 22
Grade: B
```

3. Super Keyword: Create a base class Shape with a method draw() that prints "Drawing Shape".

- Create a subclass Circle that overrides draw() to print "Drawing Circle".

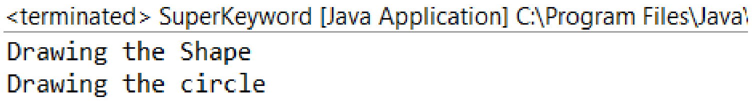
- Inside the draw() method of Circle, call the draw() method of the Shape class using super.draw().

- Write a main method to demonstrate calling draw() on a Circle object.

CODE:-

```
package SuperKeyWord;
class Shape
{
        public void draw()
        {
                System.out.println("Drawing the Shape");
        }
}
class Circle extends Shape
{
        public void draw()
        {
                super.draw();
                System.out.println("Drawing the circle");
        }
}
public class SuperKeyword
{
public static void main(String[] args)

                Shape s=new Circle();
                s.draw();
        }
}
```

OUTPUT:-



```
<terminated> SuperKeyword [Java Application] C:\Program Files\Java\
Drawing the Shape
Drawing the circle
```

4. Write a Java Program to count the number of words in a String without using the Predefined method?

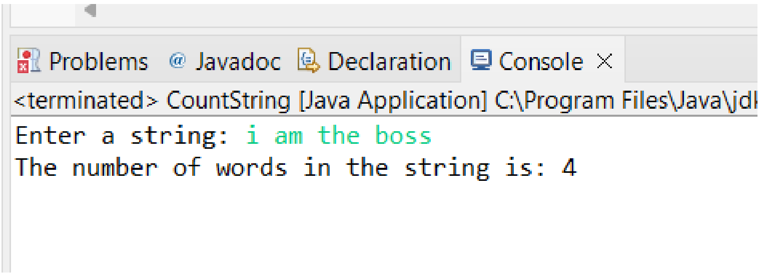
CODE:-

```
package Lab_4;

import java.util.Scanner;

public class CountString {



        public static void main(String[] args) {

                // Create a Scanner object to read input from the user

                Scanner scanner = new Scanner(System.in);


                // Prompt the user to enter the string

                System.out.print("Enter a string: ");

                // Read the entire line of input as the string

                String inputString = scanner.nextLine();


                // Call the countWords method to count the number of words in the input string

                int wordCount = countWords(inputString);
```

```java
    // Display the number of words
    System.out.println("The number of words in the string is: " + wordCount);

    // Close the scanner to free up resources
    scanner.close();
}

// Method to count the number of words in a string
public static int countWords(String str) {
    // Initialize the word count to 0
    int wordCount = 0;
    // Get the length of the string
    int length = str.length();
    // Initialize a flag to indicate if we are inside a word
    boolean isWord = false;

    for (int i = 0; i < length; i++) {
        // Check if the current character is a letter or digit (part of a word)
        if (Character.isLetterOrDigit(str.charAt(i))) {
            if (!isWord) {
                // We are entering a new word
                wordCount++;
                isWord = true;
            }
        } else {
            // We are outside a word
```

```
                isWord = false;
            }
        }

        return wordCount;
    }
}
```

OUTPUT:-



```
<terminated> CountString [Java Application] C:\Program Files\Java\jdl
Enter a string: i am the boss
The number of words in the string is: 4
```

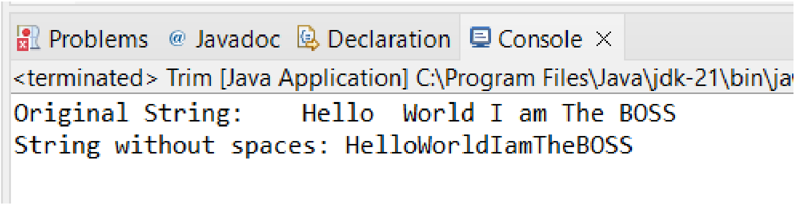
---

5. Write a Java Program to remove all white spaces from a String?

CODE:-

```
package Lab_4;

public class Trim {

    public static void main(String[] args) {
        String str = "   Hello  World I am The BOSS   ";

        // Remove leading and trailing spaces using trim()
        String trimmedString = str.trim();

        // Remove all whitespaces from the trimmed string
        String stringWithoutSpaces = trimmedString.replaceAll("\\s", "");

        System.out.println("Original String: " + str);
        System.out.println("String without spaces: " + stringWithoutSpaces);
    }
}
```

OUTPUT:-



```
<terminated> Trim [Java Application] C:\Program Files\Java\jdk-21\bin\ja
Original String:    Hello  World I am The BOSS
String without spaces: HelloWorldIamTheBOSS
```

---

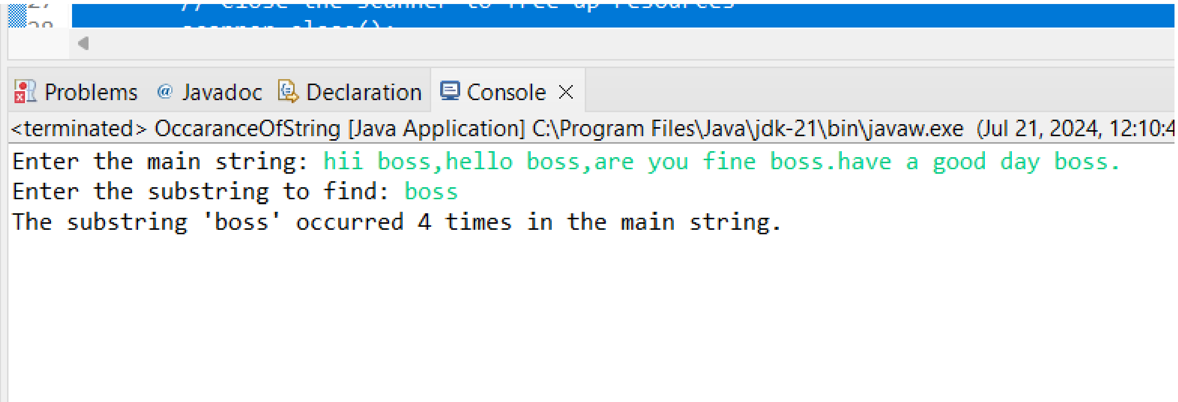
6. WAP to find occurrence of given in the given string.

CODE:-

```
package Lab_4;
import java.util.Scanner;

public class OccaranceOfString {


    public static void main(String[] args) {

        // Create a Scanner object to read input from the user
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the main string
        System.out.print("Enter the main string: ");
        // Read the entire line of input as the main string
        String mainString = scanner.nextLine();

        // Prompt the user to enter the substring to find
        System.out.print("Enter the substring to find: ");
        // Read the entire line of input as the substring
        String subString = scanner.nextLine();

        // Call the findOccurrences method to count the occurrences of the substring
        int occurrences = findOccurrences(mainString, subString);
        // Display the number of occurrences
        System.out.println("The substring '" + subString + "' occurred " + occurrences + "
times in the main string.");

        // Close the scanner to free up resources
        scanner.close();
    }

    // Method to find the number of occurrences of subString in mainString
    public static int findOccurrences(String mainString, String subString) {
        // Initialize count of occurrences to 0
        int count = 0;
        // Start searching from the beginning of the main string
        int fromIndex = 0;

        // Loop to find all occurrences of the substring
        while ((fromIndex = mainString.indexOf(subString, fromIndex)) != -1) {
```

```
                    // Increment the count for each occurrence found
                    count++;
                    // Move the start index past the current occurrence to search for further occurrences
                    fromIndex += subString.length();
                }

                // Return the total count of occurrences
                return count;


        }

}
```

OUTPUT:-



<terminated> OccaranceOfString [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Jul 21, 2024, 12:10:4

```
Enter the main string: hii boss,hello boss,are you fine boss.have a good day boss.
Enter the substring to find: boss
The substring 'boss' occurred 4 times in the main string.
```

---

7. Write a java class to implement any 10 string methods:

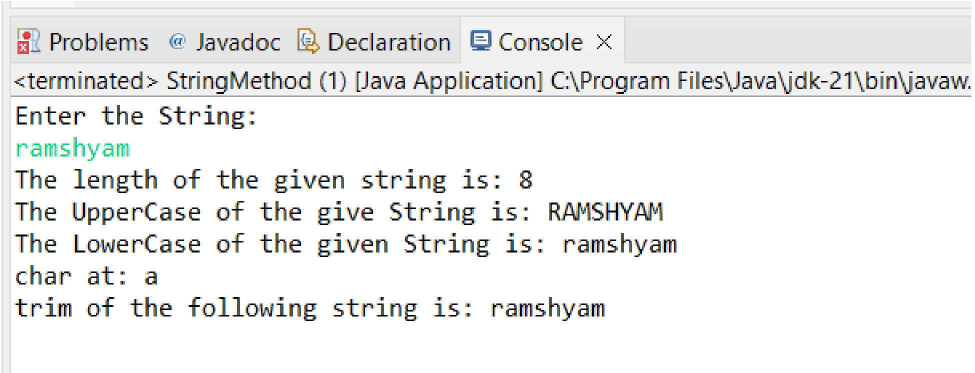
● replace ● contains ● replaceAll ● indexOf ● substring ● Equals ● lastIndexOf ● startsWith

● endsWith ● EqualsIgnoreCase ● toLowerCase ● toUpperCase ● isEmpty ● Length ● split

CODE:-

```
package Lab_4;
import java.util.Scanner;
public class StringMethod {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		String name;
		//initiating the scanner
		Scanner sc= new Scanner(System.in);

		//entering the string
		System.out.println("Enter the String: ");

		name=sc.next();

		//closing the scanner
		sc.close();

		//for finding the length of the string
		System.out.println("The length of the given string is: "+name.length());

		//getting the Uppercase of the string
		System.out.println("The UpperCase of the give String is: "+name.toUpperCase());

		//getting the lowercase of the string
		System.out.println("The LowerCase of the given String is: "+name.toLowerCase());

		//for finding char at as in which number
		System.out.println("char at: "+name.charAt(1));

		//removing whitespace of the string
		System.out.println("trim of the following string is: "+name.trim());

	}

}
```

OUTPUT:-



```
<terminated> StringMethod (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.
Enter the String:
ramshyam
The length of the given string is: 8
The UpperCase of the give String is: RAMSHYAM
The LowerCase of the given String is: ramshyam
char at: a
trim of the following string is: ramshyam
```

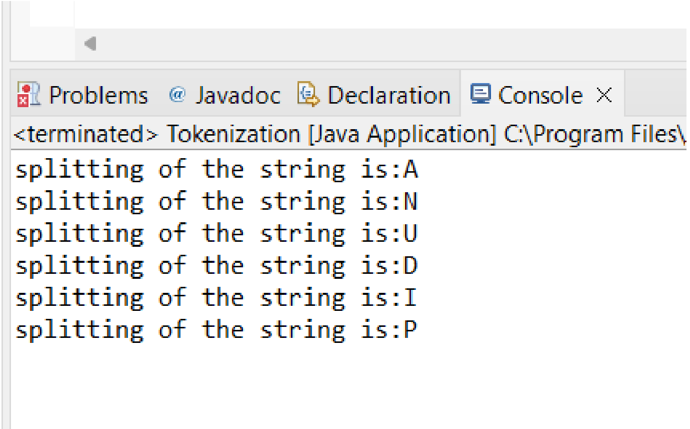
8. Write a java program to implement string tokenizer.

CODE:-

```
package Lab_4;

public class Tokenization {
	//Split()
```

```
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            String name="ANUDIP";

            // for splitting the function

            String[]nameArray=name.split("");
            for(String eachString:nameArray) {
                System.out.println("splitting of the string is:"+eachString);
            }

        }

    }
```

OUTPUT:-



<terminated> Tokenization [Java Application] C:\Program Files\

```
splitting of the string is:A
splitting of the string is:N
splitting of the string is:U
splitting of the string is:D
splitting of the string is:I
splitting of the string is:P
```