JAVA

Class 12

Agenda

String Manipulations

String

- String is one of the widely used java classes.
- String class is encapsulated under java.lang package.
- The Java platform provides the String class to create and manipulate strings.
- String are objects. i.e every string that you create is actually an object of type String

String

There are two ways to create String object:

1. By string literal

```
String str = "Hello world!";
```

2. By new keyword

```
String s1=new String("HelloWorld");
```

String Methods

.length()

Description:

This method returns the length of this string. The length is equal to the number of 16-bit Unicode characters in the string.

Syntax:

Here is the syntax of this method:

String.length()

Example

```
public static void main(String args[]){
      String Str1 = new String(" Welcome Student ");
      String Str2 = new String("Tutorials" );

       System.out.print("String Length :" );
       System.out.println(Str1.length());
       System.out.print("String Length :" );
       System.out.println(Str2.length());
   }
```

Output:

String Length :15

String Length :9

String Methods

String toLowerCase()

Description:

This method converts all of the characters in this String to lowercase

Syntax:

Here is the syntax of this method:

public String toLowerCase()

Example

```
import java.io.*;
public class Test {

    public static void main(String args[]) {

    String str = new String("Welcome to Happy
    World");
          System.out.print("Return Value :");
          System.out.println(str.toLowerCase());
      }
}
```

Output:

Return Value :welcome to happy world

String Methods

String toUpperCase()

Description:

This method converts all of the characters in this String to uppercase

Syntax:

Here is the syntax of this method:

```
public String toUpperCase()
```

Example

```
import java.io.*;
public class Test {

    public static void main(String args[]) {
    String Str = new String("Welcome on Board");
    System.out.print("Return Value :" );
    System.out.println(Str.toUpperCase() );

    }
}
```

Output:

Return Value :WELCOME ON BOARD

String Methods

boolean equals(Object anObject)

Description:

This method compares this String to another String. The result is true only if the argument is not null and if a String object that represents the same sequence of characters as this object.

Syntax:

public boolean equals(Object anObject)

Example

```
public class Test {
        public static void main(String args[]) {
        String Str1 = "This is really not immutable!!";
        String Str2 = Str1;
        String Str3 = "This is really not immutable!!";
        boolean retVal;
        retVal = Str1.equals( Str2 );
        System.out.println("Value = " + retVal );
        retVal = Str1.equals( Str3 );
        System.out.println("Value = " + retVal );
        }
}
```

Output:

Value = true

Value = true

String Methods

boolean equalsIgnoreCase(String anotherString)

Description:

This method compares this String to another String, ignoring case considerations. Two strings are considered equal ignoring case if they are of the same length, and corresponding characters in the two strings are equal ignoring case.

Syntax:

public boolean equalsIgnoreCase(String anotherString)

Example

```
public class Test {
    public static void main(String args[]) {
        String Str1 = new String("This is really not immutable!!");
        String Str2 = Str1;
        String Str3 = new String("This is really not immutable!!");
        String Str4 = new String("This IS REALLY NOT IMMUTABLE!!");
        boolean retVal;
        retVal = Str1.equals( Str2 );
        System.out.println("Returned Value = " + retVal );
        retVal = Str1.equals( Str3 );
        System.out.println("Returned Value = " + retVal );
        retVal = Str1.equalsIgnoreCase( Str4 );
        System.out.println("Returned Value = " + retVal );
    }
}
```

Output:

Returned Value = true
Returned Value = true
Returned Value = true

String Methods

boolean contains(String anotherString)

Description:

This method searches the sequence of characters in the string. If the sequences of characters are found, then it returns true otherwise returns false.

Syntax:

public boolean contains(CharSequence s)

Example

```
public class StringManipulations {

	public static void main(String[] args) {

	String str="Good morning, students!";

	boolean contains=str.contains("students");

	System.out.println(contains);
	}
}
```

Output:
true

String Methods

boolean startsWith(String prefix)

Description:

This method tests if a string starts with the specified prefix beginning

Syntax:

Here is the syntax of this method:

public boolean startsWith(String prefix)

Example

```
public class Test {

   public static void main(String args[]) {
      String str = "Welcome to the World");
      System.out.print("Return Value :" );
      System.out.println(str.startsWith("Welcome“));

      System.out.print("Return Value :" );
      System.out.println(str.startsWith(“World“));
   }
}
```

Output:

Return Value :true

Return Value :false

String Methods

boolean endsWith(String suffix)

Description:

This method tests if this string ends with the specified suffix.

Syntax:

Here is the syntax of this method:

public boolean endsWith(String suffix)

Example

```
public class Test {
public static void main(String args[]){
        String Str = "This is really not immutable!!");
        boolean retVal;
        retVal = Str.endsWith( "immutable!!" );
        System.out.println("Value = " + retVal );

        retVal = Str.endsWith( "immu" );
        System.out.println("Value = " + retVal );
}
}
```

Output:

Value = true

Value = false

String Methods

boolean isEmpty()

Description:

This method checks whether a String is empty or not. This method returns true if the given string is empty, else it returns false.

Syntax:

Here is the syntax of this method:

```
public boolean isEmpty()
```

Example

```
public class Test {
public static void main(String args[]){
        String str="";

        boolean b=str.isEmpty();

        System.out.println(b);



}
}
```

Output:
true

String Methods

String concat(String str)

Description:

This method appends one String to the end of another. The method returns a String with the value of the String passed in to the method appended to the end of the String used to invoke this method.

Syntax:

Here is the syntax of this method:
public String concat(String s)

Example

```
public class Test  {

    public static void main(String args[]) {

    String s = "Strings are immutable";
    s = s.concat("all the time");
    System.out.println(s);
   }
}
```

Output:
Strings are immutable all the time

String Methods

String trim()

Description:

This method returns a copy of the string, with leading and trailing whitespace omitted.

Syntax:

Here is the syntax of this method:

public String trim()

Example

```
public class Test {

    public static void main(String args[]) {

      String str = " Welcome Student ";
       System.out.println(“Return Value :”);
       System.out.println(Str.trim() );

     }
}
```

Output:

Return Value :Welcome Student

Task

Accept username, password and confirm password and check following requirements:

1. Username and Password cannot be empty, if so→ message=”Username and Password cannot be empty”.
2. Password should be minimum 8 characters, if less → message=”Password is too short”.
3. Password cannot contain username if so, → message=”Password cannot contain username”.
4. Password should match confirmed password, if not → message=“Passwords do not match”.

Only after all requirements met → message “Your username and password has been created”

String Methods

char charAt(int index):

Description:

This method returns the character located at the String's specified index. The string indexes start from zero.

Syntax:

public char charAt(int index)

Example

```
public class Test {

 public static void main(String args[]) {

     String s = "Strings are immutable";
       char result = s.charAt(8);
       System.out.println(result);
  }
}
```

Output:

a

String Methods

int indexOf(int ch)

Description:

This method returns the index within this string of the first occurrence of the specified character or -1 if the character does not occur.

Syntax:

public int indexOf(int ch)

Example

```
public static void main(String args[]) {
        String Str = new String("Welcome on Board");
        String SubStr1 = new String("Tutorials");
        String SubStr2 = new String("Sutorials");
         System.out.print("Found Index :" );
        System.out.println(Str.indexOf( 'o' ));

        System.out.print("Found Index :" );
        System.out.println( Str.indexOf( SubStr1 ));

        System.out.print("Found Index :" );
        System.out.println(Str.indexOf( SubStr2 ));
 }
```

Output:

```
Found Index :4
Found Index :-1
Found Index :-1
```

String Methods

String substring(int beginIndex)

Description:

This method returns a new string that is a substring of this string.

Syntax:

Here is the syntax of this method:

```
public String substring(int beginIndex)
```

Example

```
public class Test {
 public static void main(String args[]) {

      String str = "Welcome to Happy World";
      System.out.print("Return Value :" );
      System.out.println(str.substring(10));

      System.out.print("Return Value :" );
      System.out.println(str.substring(10, 15));
  }
}
```

Output:

Return Value : Happy World

Return Value : Happ

String Methods

String substring(int beginIndex, int endIndex)

Description:

This method returns a new string that is a substring that begins with the character at the specified index and extends to the end of this string

Syntax:

Here is the syntax of this method:

public String substring(int beginIndex, int endIndex)

Example

```
public class Test {
 public static void main(String args[]) {

        String Str ="Welcome to Happy World");
         System.out.print("Return Value :" );
         System.out.println(Str.substring(10));

         System.out.print("Return Value :”);
         System.out.println(Str.substring(10, 15));
  }
}
```

Output:

Return Value : Happy World

Return Value : Happ