String Handling

OBJECTIVES

String Handling

LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Create String using String class
- Use String methods
- Create StringBuilder and use its methods
- Differentiate between StringBuilder and StringBuffer

Refer package **com.mgait.api_classes** in the provided code base for demo programs on the topics covered in this presentation

CONCEPT

String

String

- String represents a sequence of characters
- Java provides “String” class to create strings
  - String class is ***final***

- String class provides methods for manipulating Strings
  - Concatenating Strings
  - Extracting part of Strings
  - Changing case of character in Strings
  - Replacing characters in String
  - And many more …

Creating Strings

- Using Literal
  - String literal is a sequence of characters enclosed in double quotes
  - String object can be created directly by assigning a string literal

```
String str = "Java World";
```

- Using Constructor
  - String object can be created using a new operator

```
String str = new String("Java World");
```

Concatenation

- Using + operator

```
String s1 = "Good";
String s2 = s1 + " Day";        //  Good Day
S2 += " folks";                 //  Good Day folks
```

- Using method concat()

```
String s1 = "Good";
String s2 = s1.concat("Day");       //  Good Day
```

Concatenation

```
System.out.println(2 + 3);                  // 5

System.out.println("O" + "K");              // OK

System.out.println("O" + "K" + 2 + 3);      // OK23

System.out.println("O" + "K" + 2 * 3);      // OK6

System.out.println(2 + 3 + "O" + "K");      // 5OK
```

Immutability

- String objects are immutable
  - Once a String object is created, it cannot be changed
  - Any change to the String object results in creation of new String object

```
String s1 = "B";
String s2 = s1.concat("Y");
s2.concat("E");
System.out.println(s2);
```

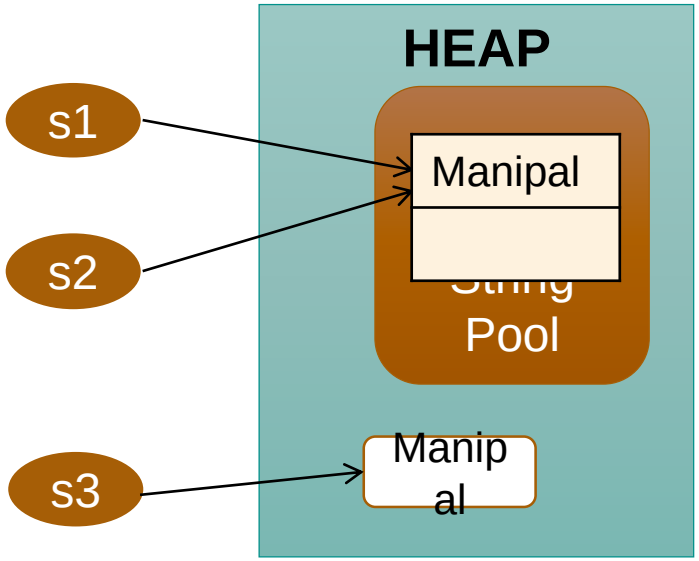
String Pool

DEMO
Class : StringDemo

- Strings are heavily used in applications
- Special memory area in Heap for storing string literals
- Allows reusing of string literals
  - i.e. existence of identical literals are checked, before creation of literals
  - If identical literal exists, only reference is assigned

```
String s1 = "Manipal";
String s2 = "Manipal";
System.out.println(s1 == s2);   //TRUE
```

```
String s3 = new String("Manipal");
System.out.println(s1 == s3);   //FALSE
```



String Methods

```
String str = "Prolearn";
```

- length()
  - Returns the number of characters in a String

```
System.out.println(str.length());  // 8
```

- valueOf(..)
  - Returns the String representation of primitives and objects

```
int price = 100;
String priceStr = String.valueOf(price);
```

String Methods – Checking occurrence of Characters/Strings

```
String str = "Prolearn";

System.out.println(str.indexOf('r'));                    // 1

System.out.println(str.lastIndexOf('r'));                // 6

System.out.println(str.indexOf("learn"));                // 3

System.out.println(str.contains("role"));                // true

System.out.println(str.startsWith("Pro"));               // true

System.out.println(str.endsWith("learn"));               // true
```

String Methods - Manipulation

```
String str = "Prolearn";

System.out.println(str.toUpperCase());                  //PROLEARN

System.out.println(str.toLowerCase());                  //prolearn

System.out.println(str.replace('r', 'o'));              //Pooleaon

System.out.println(str.replace("Pro", "Prof"));         //Proflearn

String str2 = "  Hello World  "
System.out.println(str2.trim());                        //Hello World
```

String Methods - Extraction

```
String str = "Prolearn";

char c = str.charAt(2);
System.out.println(c);                          // o

char[] charArray = str.toCharArray();
System.out.println(charArray[0]);               //P

System.out.println(str.substring(3));           //learn
System.out.println(str.substring(1,5));         //role


String str2 = "learning strings";
String[] words = str2.split(" ");
System.out.println(words[1]);                   //strings
```

String Methods - Comparison

```
String s1 = new String("ok");
String s2 = new String("ok");
String s3 = new String("OK");

System.out.println(s1.equals(s2));            //true
System.out.println(s1.equalsIgnoreCase(s3));  //true

String s4 = "A";
String s5 = "B";
System.out.println(s4.compareTo(s5));         //-1
```

String Methods – Method Chaining

```
String str = " mole ";

String new = str.trim().replace('m','r').toUpperCase();
System.out.println(new);                                    //ROLE
```

CONCEPT

StringBuilder & StringBuffer

StringBuilder

- StringBuilder class is used to create mutable (modifiable) string objects.

```
StringBuilder sb1 = new StringBuilder();
StringBuilder sb2 = new StringBuilder("prolearn");
```

StringBuilder Methods

- append(): The characters of the String argument are appended

```
StringBuilder Builder = new StringBuilder(“Java”);
Builder.append(“Programming”);
```

- insert(): The characters of the String argument are inserted, into the sequence indicated in the offset.

```
StringBuilder sb = new StringBuilder(“Java");
sb.insert(3,“Programming");
```

StringBuilder Methods

- delete(int start, int end)
- deleteCharAt(int index)
- replace(int start, int end, String s)
- setCharAt(int index, char c)
- reverse()
- toString()

StringBuffer vs StringBuilder

- StringBuffer is used for creating mutable String objects like StringBuilder
- StringBuffer methods are thread safe
- Contains same methods as StringBuilder

QUIZ QUESTION

1. Which of the following are valid declarations of a String?

- ❑ String s1 = null;
- ❑ String s2 = 'null';
- ❑ String s3 = 'abc';
- ❑ String s4 = "abc";

2. Which of these operators can be used to concatenate two or more String objects?

- ❑ +
- ❑ +=
- ❑ &
- ❑ ||

3. Which of these method of class String is used to extract a single character from a String object?

- ❑ subString
- ❑ extract
- ❑ charAt
- ❑ ChatAt

4. Which of these class is used to create an object whose character sequence is mutable?

- ❑ String
- ❑ StringBuilder
- ❑ Both
- ❑ None

SUMMARY

String Handling

SUMMARY

In this lesson, you've learned to:

- Create String using String class
- Use String methods
- Create StringBuilder and use its methods
- Differentiate between StringBuilder and StringBuffer