Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna

Outline

1. String
2. Methods of String
3. Examples of String
4. StringBuffer and StringBuilder

String

- In Java, a string is a sequence of characters, i.e. a character array. For example, “hello” is a string containing a sequence of characters ‘h’, ‘e’, ‘l’, ‘l’ and ‘o’.
- We use double quotes to represent a string in Java. For example, String Variable_Name = “Sequence_Characters?”;
- String in Java is not primitive type.
- All strings are objects of a predefined class named **String** and all string variables are instances of the **String class**.
- String is an immutable object, i.e. it is constant and cannot be changed once it is created.

String (Cont…)

- If we need to make modifications to Strings, we should use **StringBuffer** and **StringBuilder** classes.
- The String class has 11 constructors that allow us to provide the initial value of the string using different sources.
- It has many methods to perform many tasks.
- The java.lang.String class implements Serializable, Comparable and CharSequence interfaces.
- The CharSequence interface is used to represent the sequence of characters. String, StringBuffer and StringBuilder classes implement it.

String (Cont…)

Creating a String

- There are two ways to create a String:

1. **String literal:** Whenever a String object is created as a literal, the object will be created in **String constant pool**. This allows JVM to optimize the initialization of String literal. Example:

   String str = "Welcome";

2. **Using new Operator:** In case of Strings are dynamically allocated, they are assigned a new memory location in heap. This string is not be added to String constant pool. Example:

   String str = new String("Welcome");

Methods of String

- There are many methods, some are given below:

Method	Description
char charAt(int index)	It returns char value for the particular index.
int length()	It returns string length.
String substring(int beginIndex)	It returns substring for given begin index.
String substring(int beginIndex, int endIndex)	It returns substring for given begin index and end index.
boolean isEmpty()	It checks if string is empty or not.
String concat(String str)	It concatenates the specified string.
String toLowerCase()	It returns a string in lowercase.
String toUpperCase()	It returns a string in uppercase.
int indexOf(String substring)	It returns the specified substring index.

Examples of String

```
class String1
{
   public static void main(String args[])
   {
      String str1="NIT Patna";
      char ch[]={'C','S','E', '\0'};
      String str2 = new String(ch);
      String str3 = new String("Department of CSE");
      System.out.println(str1);
      System.out.println(str2);
      System.out.println(str3);
   }
}
```

Examples of String (Cont…)

- Output

  NIT Patna

  CSE

  Department of CSE

Examples of String

```
class String2
{
   public static void main(String args[])
   {
      String str = "Hello World";
      System.out.println("String: " + str);
      int x = str.length();
      System.out.println("Length of the string: " + x);
  }
}
```

Examples of String (Cont…)

- Output

  String: Hello World

  Length of the string: 11

Examples of String (Cont…)

```
class String3
{
   public static void main(String args[])
   {
      String str1 = "Hello, ";
      String str2 = "How are you?";
      String str3 = str1.concat(str2);
      String str4 = str1 + str2;
      System.out.println("Joined String: " + str3);
      System.out.println("Joined String: " + str4);
   }
}
```

Examples of String (Cont…)

- Output

```
Joined String: Hello, How are you?
Joined String: Hello, How are you?
```

Examples of String (Cont…)

```
class String4
{
   public static void main(String args[])
   {
      String str1 = "Hello, ";
      String str2 = "Hello, ";
      String str3 = "Hey";
      boolean b1 = str1.equals(str2);
      boolean b2 = str2.equals(str3);
      System.out.println("First result: " + b1 + ", Second result: " + b2);
   }
}
```

Examples of String (Cont…)

- Output

  First result: true, Second result: false

StringBuffer and StringBuilder

- StringBuffer and StringBuilder classes are used, when there is a necessity to make modifications to Strings of characters.
- Unlike String, objects of type StringBuffer and StringBuilder can be modified over and over again.
- The main difference between the StringBuffer and StringBuilder is that StringBuilder's methods are not thread safe (not synchronised).
- StringBuilder is recommended to use because it is faster than StringBuffer.
- When thread safety is necessary, we must use StringBuffer.

StringBuffer and StringBuilder (Cont…)

StringBuffer

- Java StringBuffer class is used to create mutable string.
- Java StringBuffer class is thread-safe, i.e. multiple threads cannot access it simultaneously. So, it is safe and will result in an order.
- Important constructors of StringBuffer are:
  - **StringBuffer():** It creates an empty string buffer with the initial capacity of 16.
  - **StringBuffer(String str):** It creates a string buffer with the specified string.
  - **StringBuffer(int capacity):** It creates an empty string buffer with the specified capacity as length.

StringBuffer and StringBuilder (Cont…)

StringBuffer (Cont…)

- Some important methods are:

Method	Description
public StringBuffer append(String s)	It is used to append the specified string with this string.
public StringBuffer insert(int offset, String s)	It is used to insert the specified string with this string at the specified position.
public StringBuffer replace(int startIndex, int endIndex, String str)	It is used to replace the string from specified startIndex and endIndex.
public StringBuffer delete(int startIndex, int endIndex)	It is used to delete the string from specified startIndex and endIndex.
public StringBuffer reverse()	It is used to reverse the string.

StringBuffer and StringBuilder (Cont…)

StringBuilder

- StringBuilder class is also used to create mutable string.
- StringBuilder class is same as StringBuffer class except that it is non-synchronized.
- Important constructors of StringBuffer are:
  - **StringBuilder():** It creates an empty string builder with the initial capacity of 16.
  - **StringBuilder(String str):** It creates a string builder with the specified string.
  - **StringBuilder(int capacity):** It creates an empty string builder with the specified capacity as length.

StringBuffer and StringBuilder (Cont…)

StringBuilder (Cont…)

- Some important methods are:

Method	Description
public StringBuilder append(String s)	It is used to append the specified string with this string.
public StringBuilder insert(int offset, String s)	It is used to insert the specified string with this string at the specified position.
public StringBuilder replace(int startIndex, int endIndex, String str)	It is used to replace the string from specified startIndex and endIndex.
public StringBuilder delete(int startIndex, int endIndex)	It is used to delete the string from specified startIndex and endIndex.
public StringBuilder reverse()	It is used to reverse the string.
public int capacity()	It is used to return the current capacity.

StringBuffer and StringBuilder (Cont…)

- **Differences among String, StringBuffer and StringBuilder**

Parameter	String	StringBuffer	StringBuilder
Storage	String Pool/Heap	Heap	Heap
Mutability	Immutable	Mutable	Mutable
Thread Safe	Not used in a threaded environment	Used in a multi-threaded environment	Used in a single-threaded environment
Performance	Slow	Slower than StringBuilder, but faster than String	Faster than StringBuffer

StringBuffer and StringBuilder (Cont…)

```
class String5
{
   public static void main(String args[])
   {
      StringBuffer sb = new StringBuffer("Hello ");
      sb.append("World");
      System.out.println(sb);
      sb.insert(5, "Hey");
      System.out.println(sb);
      sb.replace(2, 6, "CSE");
      System.out.println(sb);
   }
}
```

StringBuffer and StringBuilder (Cont…)

- Output

```
Hello World
HelloHey World
HeCSEey World
```

StringBuffer and StringBuilder (Cont…)

```
class String6
{
   public static void main(String args[])
   {
      StringBuffer sb = new StringBuffer();
      System.out.println(sb.capacity());
      sb.append("NIT Patna");
      System.out.println(sb);
      sb.delete(1, 3);
      System.out.println(sb);
      System.out.println(sb.capacity());
      sb.reverse();
      System.out.println(sb);
      System.out.println(sb.capacity());
      sb.append("Hello, how are you?");
      System.out.println(sb);
      System.out.println(sb.capacity());
   }
}
```

StringBuffer and StringBuilder (Cont…)

- Output

```
16
NIT Patna
N Patna
16
antaP N
16
antaP NHello, how are you?
34
```

StringBuffer and StringBuilder (Cont…)

```
class String7
{
   public static void main(String args[])
   {
      StringBuilder sd = new StringBuilder("Hello ");
      sd.append("World");
      System.out.println(sd);
      sd.insert(1, "Hey");
      System.out.println(sd);
      sd.replace(1, 5, "CSE");
      System.out.println(sd);
   }
}
```

StringBuffer and StringBuilder (Cont…)

- Output

StringBuffer and StringBuilder (Cont…)

```
class String8
{
   public static void main(String args[])
   {
      StringBuilder sd = new StringBuilder();
      System.out.println(sd.capacity());
      sd.append("NIT Patna");
      System.out.println(sd);
      sd.delete(1, 3);
      System.out.println(sd);
      System.out.println(sd.capacity());
      sd.reverse();
      System.out.println(sd);
      System.out.println(sd.capacity());
      sd.append("Hello, how are you?");
      System.out.println(sd);
      System.out.println(sd.capacity());
   }
}
```

StringBuffer and StringBuilder (Cont…)

- Output

```
16
NIT Patna
N Patna
16
antaP N
16
antaP NHello, how are you?
34
```

thank you . . .

Slides are prepared from various sources, such as Book, Internet Links and many more.