Object-Oriented Programming (CS F213)

Module II: Arrays and Strings in Java

CS F213 RL 8.4: StringBuffer class in Java

BITS Pilani

Dr. Pankaj Vyas
Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 8.4 : Topics

- StringBuffer class in Java

StringBuffer class

- Supports mutable, modifiable, writable and growable strings in Java
- You can insert characters or sub-strings in the middle or in the end
- 'length' of a StringBuffer instance means how many characters it currently holds. The current 'length' of a StringBuffer instance can be checked via length() method
- 'capacity' of a StringBuffer instance means how many more characters it can accommodate. The current 'capacity' of a StringBuffer instance can be checked via capacity() method
- 'capacity' increases automatically whenever the need arises. The rule to increase capacity is ' if current-capacity < 16 then new-capacity = 16 else new-capacity = 2 * current-capacity + 2.
- Important Constructors
  1. StringBuffer() → length =0, capacity = length +16 = 16
  2. StringBuffer(int size) → length =0, capacity = size
  3. StringBuffer(String str) → length =str.length(), capacity = length + 16

StringBuffer class : Example 1

```
StringBuffer strb1 = new StringBuffer();
System.out.println(strb1.length());        → 0
System.out.println(strb1.capacity());      → 16


StringBuffer strb2 = new StringBuffer(30);
System.out.println(strb2.length());        → 0
System.out.println(strb2.capacity());      → 30


StringBuffer strb3 = new StringBuffer("Java");
System.out.println(strb2.length());        → 4
System.out.println(strb2.capacity());      → 20
```

StringBuffer class : Important Methods

- void setCharAt(int where, char ch) → sets character 'ch' at where index. (0 <=where<=L, where L is length)
- StringBuffer append(String str) → appends 'str' at the end of invoking string buffer instance
- StringBuffer append(int num) → appends a character represented by 'num' at the end of invoking string buffer instance
- StringBuffer append(Object obj) → appends the string form of 'obj' at the end of invoking string buffer instance
  - ➢ String form of 'obj' is determined by 'toString()' method
- String insert(int index, String str) → Inserts String 'str' at 'index' in the invoking string buffer instance [0 <= index <= L, L is length]
- String insert(int index, char ch) → Inserts character 'ch' at 'index' in the invoking string buffer instance [0 <= index <= L, L is length]
- String insert(int index, Object obj) → Inserts Object 'obj' at 'index' in the invoking string buffer instance [0 <= index <= L, L is length]
  - ➢ 'obj' is first converted to String form by calling 'toString()' method and then the resultant string form of 'obj' is inserted at 'index' in the invoking string buffer instance

StringBuffer class : Important Methods ….

- StringBuffer reverse() → reverses the characters of the invoking string buffer instance and returns the reversed string
- StringBuffer deleteChatAt(int loc) → deletes a single character at 'loc' and returns the updated string. [0 <=loc <=L, L is the length of string buffer]
- StringBuffer delete(int start, int end) → deletes sequence of characters from index 'start' (inclusive) to index 'end' (exclusive) and returns the updated string. Characters will be deleted from indexes 'start' to 'end-1'. [0 <=start,end <=L and the value of end should not be less than start]
- StringBuffer replace(int startIndex, int endIndex, String str) → Replaced the character sequence from startIndex to endIndex by str. Characters will be replaced from indexes 'startIndex' to 'endIndex-1'.

String Buffer Example 2

```
StringBuffer strb1 = new StringBuffer("Object Oriented Programming");
System.out.println(strb1.length());            → 27
System.out.println(strb1.capacity());          → 43

System.out.println(strb1.append("Object Oriented Programming")); → Object Orineted ProgrammingObject Orineted Programming
System.out.println(strb1.length());            → 54
System.out.println(strb1.capacity());          → 88

StringBuffer strb2 = new StringBuffer(5);
System.out.println(strb2.length());            → 0
System.out.println(strb2.capacity());          → 5

System.out.println(strb2.append("Incredible India")); → Incredible India
System.out.println(strb2.length());            → 16
System.out.println(strb2.capacity());          → 16

System.out.println(strb2.append("Incredible India"));     // LENGTH = 32 , CAPACITY = 2 * 16 +2 = 34
System.out.println(strb2.length());            → 32
System.out.println(strb2.capacity());          → 34
System.out.println(strb2.append("Incredible India"));     // LENGTH = 32 + 16 = 48, CAPACITY = 2 * 34 + 2 = 70
System.out.println(strb2.append("Incredible India"));     // LENGTH = 48 + 16 = 64, CAPACITY = 70
System.out.println(strb2.length());            → 64
System.out.println(strb2.capacity());          → 70
```

ensureCapacity()

- StringBuffer ensureCapacity(int minCapacity) → sets the current capacity to minCapacity as per the following rule

```
if      minCapacity <= current-capacity                           Then
                no change in capacity [current-capacity remains same]
else    if      minCapacity > current-capacity && minCapacity <= 2* current-capacity +2    Then
                new-capacity = 2* current-capacity +2
else            new-capacity = minCapacity
```

```
StringBuffer strB3 = new StringBuffer();
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =16
strB3.ensureCapacity(10);               // 10 < 16
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =16
```

```
StringBuffer strB3 = new StringBuffer(5);
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =5
strB3.ensureCapacity(10);               // 10 > 5 && 10 < 12
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =12
```

```
StringBuffer strB3 = new StringBuffer();
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =16
strB3.ensureCapacity(20);               // 20 > 16 && 20 < 34
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =34
```

```
StringBuffer strB3 = new StringBuffer();
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =16
strB3.ensureCapacity(50);               // 50 > 16 && 50 > 34
System.out.println(strB3.length());     // Length    =0
System.out.println(strB3.capacity());   // Capacity  =50
```

setLength()

- void setlength(int len) → Sets the current lentgth of the invoking string buffer object to 'len'. Changing length may affect capacity also
- Suppose 'L' is the current length of a string buffer instance and 'C' is the current capacity.
- If 'len' < L (current length) then 'L-len' characters will be removed from the end of the string buffer instance and 'len' becomes the new length of string buffer but no change in 'C'
- If 'len' > L (current length) && 'len' <= 'C' (current capacity) then extra white space characters will be appended in the end of the string buffer instance and 'len' becomes the new length of string buffer but no change in 'C'
- If 'len' > 'C' (current capacity) && 'len' <= 2*C + 2 then extra white space characters will be appended in the end of the string buffer instance and 'len' becomes the new length of string buffer and new capacity = 2 * 'C' + 2.
- If 'len' > 2*C + 2 then extra white space characters will be appended in the end of the string buffer instance and 'len' becomes the new length of string buffer and new capacity = 'len'.

setLength() : Example

```
StringBuffer strB = new StringBuffer("Object");
System.out.println(strB.length());
System.out.println(strB.capacity());
strB.setLength(4);
System.out.println(strB.length());
System.out.println(strB.capacity());
```

```
6
22
4
22
```

```
StringBuffer strB = new StringBuffer("Object");
System.out.println(strB.length());
System.out.println(strB.capacity());
strB.setLength(16);
System.out.println(strB.length());
System.out.println(strB.capacity());
```

```
6
22
16
22
```

```
StringBuffer strB = new StringBuffer("Object");
System.out.println(strB.length());
System.out.println(strB.capacity());
strB.setLength(30);
System.out.println(strB.length());
System.out.println(strB.capacity());
```

```
6
22
30
46
```

```
StringBuffer strB = new StringBuffer("Object");
System.out.println(strB.length());
System.out.println(strB.capacity());
strB.setLength(80);
System.out.println(strB.length());
System.out.println(strB.capacity());
```

```
6
22
80
80
```

innovate achieve lead

Thank You