

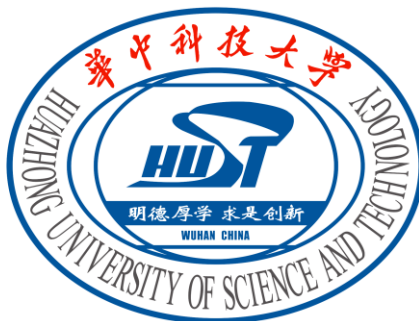
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

第19讲：Java字符串



1. 初始化
2. 常用方法
3. 类型转换
4. StringBuilder

□ 创建 Creation

String类有13个不同的构造函数，可以借助char数组和byte数组初始化。

```
String a = new String("Hello World");  
  
char[] helloArray = {'h', 'e', 'l', 'l', 'o', '.'};  
String helloString = new String(helloArray);  
  
String a = "Hello World";
```

```
String s1;  
String s2 = "";
```

```
String s1 = null;
```

2. 常用方法



□ 方法 Methods

0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		W	o	r	l	d

字符串比较

字符串查找

字符串操作

□ 字符串比较

```
String a = new String("Hello World");

String str1 = new String("ABC");
String str2 = new String("ABC");
System.out.println(str1 == str2); // #1

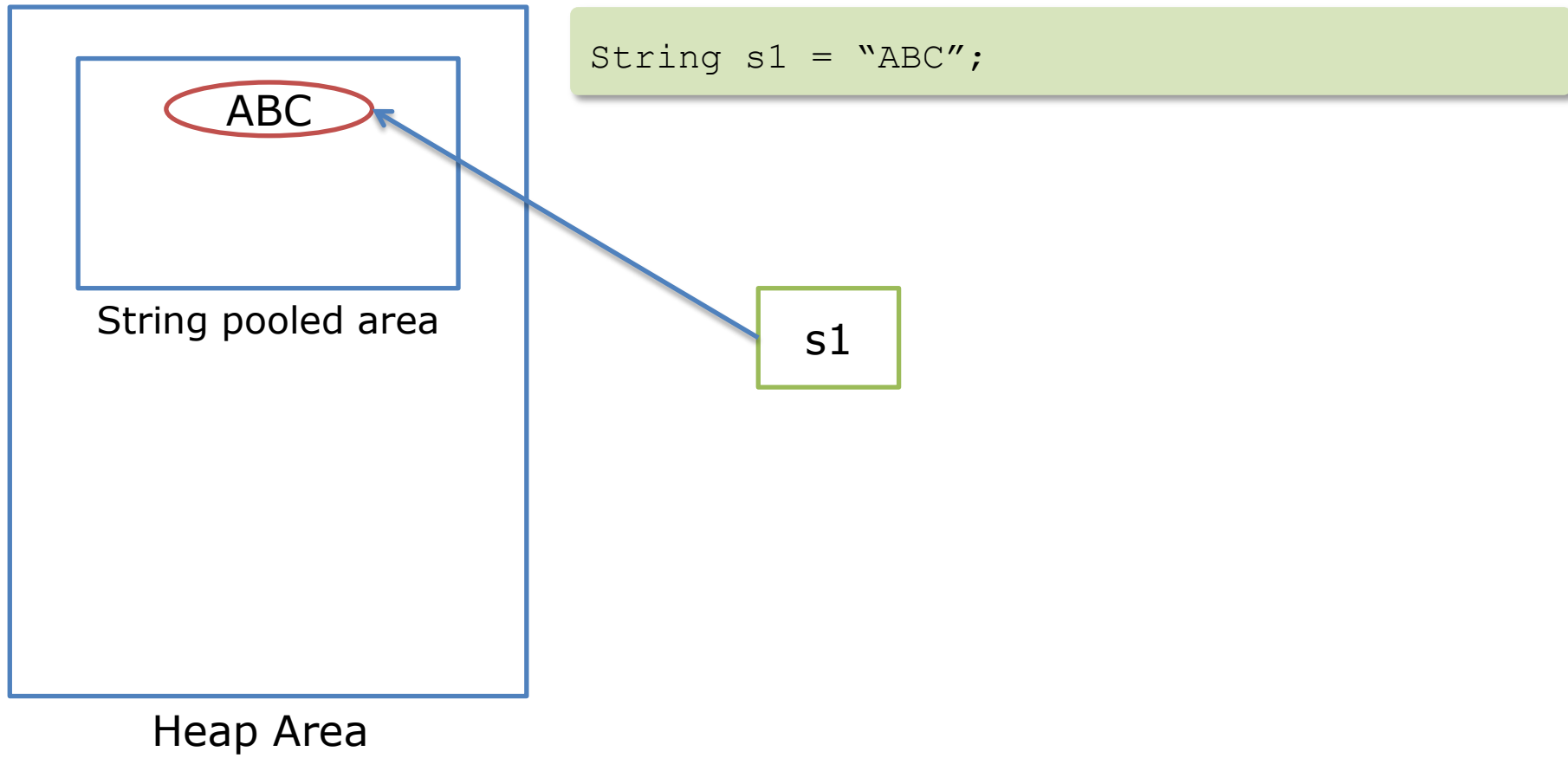
String str3 = "ABC";
String str4 = "ABC";
String str5 = "AB" + "C";
System.out.println( str3 == str4 ); // #2
System.out.println(str3 == str5 ); // #3

String a  = "ABC";
String b = "AB";
String c = b + "C";
System.out.println( a == c ); // #4
```

2. 常用方法



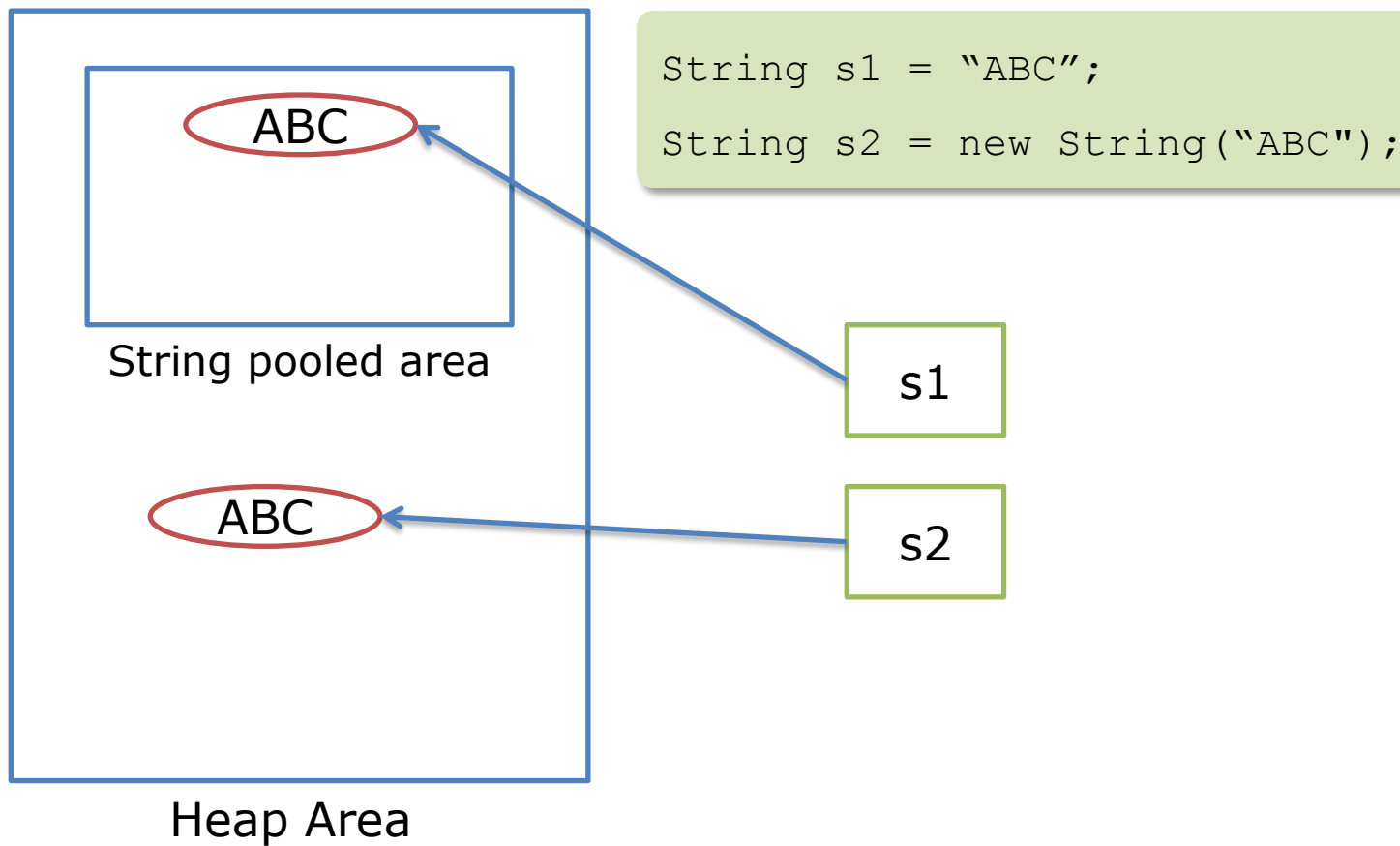
□ 字符串比较



2. 常用方法



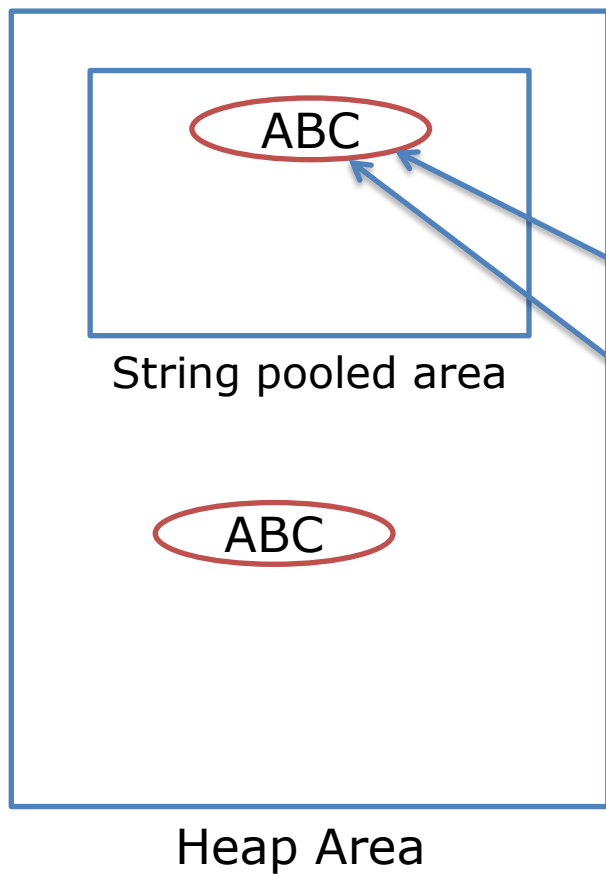
□ 字符串比较



2. 常用方法



□ 字符串比较



```
String s1 = "ABC";  
String s2 = new String("ABC");  
s2 = "ABC";
```

s1

s2

2. 常用方法



□ 字符串比较

比较方法	描述
<code>== operator</code>	比较引用而不是值
<code>equals (Object o)</code>	比较值
<code>compareTo ()</code>	从左至右比较值的大小

```
String s1 = "Ram";  
String s2 = "Ram";  
String s3 = new String(" Ram");  
String s4 = new String(" Ram");  
String s5 = "Shyam";  
String nulls1 = null;  
String nulls2 = null;
```

```
System.out.println(" Comparing strings with equals:");  
System.out.println(s1.equals(s2));  
System.out.println(s1.equals(s3));  
System.out.println(s1.equals(s5));
```

```
System.out.println(" Comparing strings with ==:");  
System.out.println(s1==s2);  
System.out.println(s1==s3);  
System.out.println(s3==s4);
```

```
System.out.println(" Comparing strings with compareTo:");  
System.out.println(s1.compareTo(s3));  
System.out.println(s1.compareTo(s5));
```

❑ 字符串查找

```
int indexOf(int ch)
int lastIndexOf(int ch)
```

```
int indexOf(int ch, int fromIndex)
int lastIndexOf(int ch, int fromIndex)
```

```
int indexOf(String str)
int lastIndexOf(String str)
```

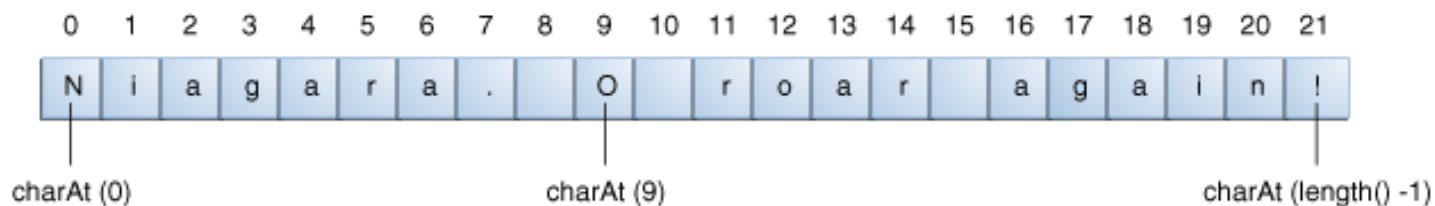
```
int indexOf(String str, int fromIndex)
int lastIndexOf(String str, int fromIndex)
```

```
boolean contains(CharSequence s)
```

□ 字符串操作

A. 获取字符

```
String anotherPalindrome = "Niagara. O roar again!";  
char aChar = anotherPalindrome.charAt(9);
```



B. 获取子串

```
String roar = anotherPalindrome.substring(11,15);  
String again = anotherPalindrome.substring(16);
```

□ 字符串操作

C. 字符串分割

```
String[] split(String regex);  
String[] split(String regex, int limit);
```

grass@over@grass

regex	limit	result
@	2	{"grass", "over@grass"}
@	5	{"grass", "over", "grass"}
@	-2	{"grass", "over", "grass"}
s	5	{"gra", "", "@over@gra", "", ""}
s	-2	{"gra", "", "@over@gra", "", ""}
s	0	{"gra", "", "@over@gra"}

□ 字符串操作

D. 字符串拼接

```
String concat(String str)
```

E. 字符串变换

```
String trim()
```

```
String toLowerCase()
```

```
String toUpperCase()
```

F. 局部替换

```
String replace(char oldChar, char newChar)
```

```
String replace(CharSequence target, CharSequence replacement)
```

```
String replaceAll(String regex, String replacement)
```

```
String replaceFirst(String regex, String replacement)
```

1. Consider the following string:

```
String hannah = "Did Hannah see bees? Hannah did.";
```

- What is the value displayed by the expression `hannah.length()`?
- What is the value returned by the method call `hannah.charAt(12)`?
- Write an expression that refers to the letter `b` in the string referred to by `hannah`.

2. How long is the string returned by the following expression? What is the string?

```
"Was it a car or a cat I saw?".substring(9, 12);
```

3. 类型转换



❑ String → Numbers

```
String strInteger = new String("10");  
int num1 = Integer.parseInt(strInteger);  
  
String strFloat = new String("3.14");  
float num2 = Float.parseFloat(strFloat);
```

❑ Numbers → String

```
int num1 = 10;  
float num2 = 3.14f;  
  
String str1 = String.valueOf(num1);  
String str2 = String.valueOf(num2);
```


4. StringBuilder



```
public final class StringBuilder  
extends Object  
implements Serializable, CharSequence
```

```
public final class StringBuffer  
extends Object  
implements Serializable, CharSequence
```

StringBuilder和String类似，但不同的是，StringBuffer 和 StringBuilder 类的对象能够被多次的修改，并且不产生新的未使用对象。

```
void setLength(int newLength)  
void ensureCapacity(int minCapacity)  
StringBuilder append(xxx)  
StringBuilder insert(xxx)
```

In the following program, called *ComputeResult*, what is the value of *result* after each numbered line executes?

```
public class ComputeResult {  
    public static void main(String[] args) {  
        String original = "software";  
        StringBuilder result = new StringBuilder("hi");  
        int index = original.indexOf('a');  
  
        /*1*/ result.setCharAt(0, original.charAt(0));  
        /*2*/ result.setCharAt(1, original.charAt(original.length()-1));  
        /*3*/ result.insert(1, original.charAt(4));  
        /*4*/ result.append(original.substring(1,4));  
        /*5*/ result.insert(3, (original.substring(index, index+2) + " "));  
  
        System.out.println(result);  
    }  
}
```

si
se
swe
sweoft
swear oft

Java 集合